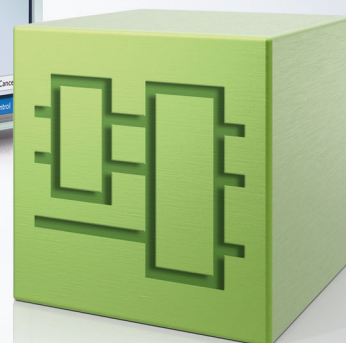
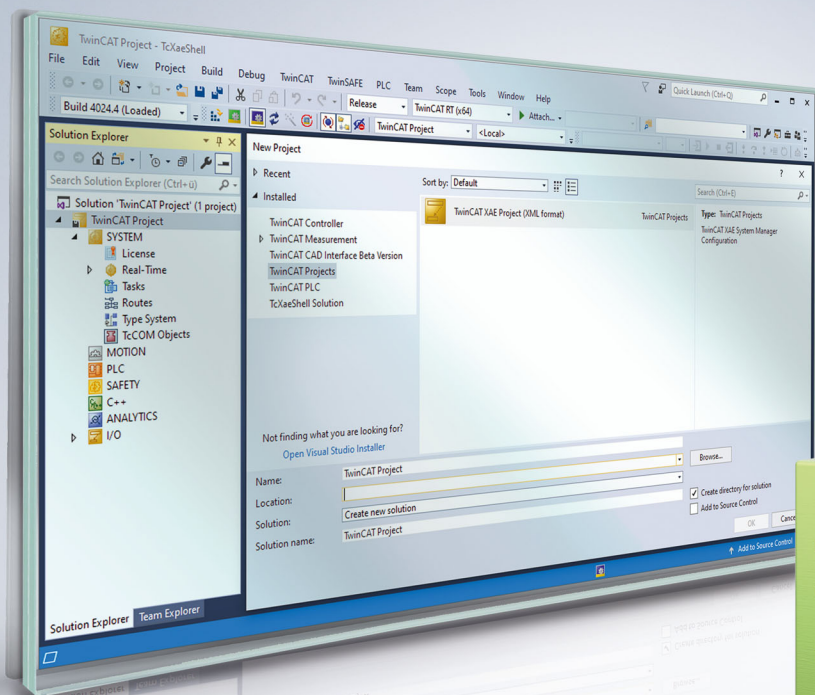


BECKHOFF New Automation Technology

Handbuch | DE

TE1000

TwinCAT 3 | PLC-Bibliothek: Tc2_Utilities



Inhaltsverzeichnis

1	Vorwort.....	13
1.1	Hinweise zur Dokumentation	13
1.2	Zu Ihrer Sicherheit.....	14
1.3	Hinweise zur Informationssicherheit	15
2	Übersicht.....	16
3	Funktionsbausteine	26
3.1	[veraltet]	26
3.1.1	FB_GetDeviceIdentification.....	26
3.1.2	FB_FileTimeToTzSpecificLocalTime	27
3.1.3	FB_TzSpecificLocalTimeToFileTime	29
3.2	BCD_TO_DEC	32
3.3	DCF77_TIME	32
3.4	DCF77_TIME_EX	36
3.5	DEC_TO_BCD	39
3.6	FB_AdsReadEvents.....	40
3.7	FB_AddRouteEntry	41
3.8	FB_AddRouteEntryEx	42
3.9	FB_AmsLogger	43
3.10	FB_BasicPID	45
3.11	FB_CalcHashValue.....	47
3.12	FB_CheckLicense	48
3.13	FB_CSVMemBufferReader	49
3.14	FB_CSVMemBufferWriter	50
3.15	FB_EnumFindFileEntry	51
3.16	FB_EnumFindFileList.....	53
3.17	FB_EnumRouteEntry	54
3.18	FB_EnumStringNumbers	56
3.19	FB_FileProperties	58
3.20	FB_FileRingBuffer	59
3.21	FB_FileTime64ToTzSpecificLocalTime	62
3.22	FB_FormatString.....	64
3.23	FB_FormatString2.....	65
3.24	FB_GetAdaptersInfo	67
3.25	FB_GetAdaptersInfoEx	68
3.26	FB_GetDeviceIdentificationEx	69
3.27	FB_GetDongleSystemID.....	70
3.28	FB_GetHostAddrByName	71
3.29	FB_GetHostName	72
3.30	FB_GetLicenseDongles	73
3.31	FB_GetLicenses.....	73
3.32	FB_GetLicensesEx	75
3.33	FB_GetLocalAmsNetId	76
3.34	FB_GetRouterStatusInfo	77
3.35	FB_GetSystemId.....	78

3.36	FB_GetTimeZoneInformation.....	78
3.37	FB_GetVolumeld.....	80
3.38	FB_HashTableCtrl.....	80
3.39	FB_LicFileCopyFromDongle.....	82
3.40	FB_LicFileCopyToDongle.....	83
3.41	FB_LicFileCreate.....	84
3.42	FB_LicFileDelete.....	85
3.43	FB_LicFileGetStorageInfo.....	86
3.44	FB_LicFileRead.....	87
3.45	FB_LinkedListCtrl.....	88
3.46	FB_LocalSystemTime.....	90
3.47	FB_MemBufferMerge.....	93
3.48	FB_MemBufferSplit.....	95
3.49	FB_MemRingBuffer.....	96
3.50	FB_MemRingBufferEx.....	97
3.51	FB_MemStackBuffer.....	99
3.52	FB_RegQueryValue.....	101
3.53	FB_RegSetValue.....	104
3.54	FB_RemoveRouteEntry.....	107
3.55	FB_ScopeServerControl.....	108
3.56	FB_SetTimeZoneInformation.....	111
3.57	FB_StringRingBuffer.....	112
3.58	FB_SystemTimeToTzSpecificLocalTime.....	113
3.59	FB_TzSpecificLocalTimeToFileTime64.....	115
3.60	FB_TzSpecificLocalTimeToSystemTime.....	117
3.61	FB_WritePersistentData.....	119
3.62	GetRemotePCInfo.....	120
3.63	NT_AbortShutdown.....	121
3.64	NT_GetTime.....	122
3.65	NT_Reboot.....	123
3.66	NT_SetLocalTime.....	124
3.67	NT_SetTimeToRTCTime.....	125
3.68	NT_Shutdown.....	127
3.69	NT_StartProcess.....	128
3.70	PLC_ReadSymInfo.....	130
3.71	PLC_ReadSymInfoByName.....	131
3.72	PLC_ReadSymInfoByNameEx.....	132
3.73	PLC_Reset.....	134
3.74	PLC_Start.....	134
3.75	PLC_Stop.....	135
3.76	Profiler.....	136
3.77	RTC.....	138
3.78	RTC_EX.....	139
3.79	RTC_EX2.....	140
3.80	TC_Config.....	142
3.81	TC_CpuUsage.....	142

3.82	TC_Restart.....	144
3.83	TC_Stop.....	144
3.84	TC_SysLatency.....	145
3.85	WritePersistentData.....	146
4	Funktionen.....	149
4.1	Zeitfunktionen.....	149
4.1.1	DT_TO_FILETIME64.....	149
4.1.2	DT_TO_SYSTEMTIME.....	149
4.1.3	F_GetDayOfMonthEx.....	150
4.1.4	F_GetDayOfWeek.....	151
4.1.5	F_GetDOYOfYearMonthDay.....	151
4.1.6	F_GetMaxMonthDays.....	152
4.1.7	F_GetMonthOfDOY.....	153
4.1.8	F_GetWeekOfTheYear.....	153
4.1.9	F_TranslateFileTime64Bias.....	154
4.1.10	F_YearIsLeapYear.....	155
4.1.11	FILETIME64_TO_DT.....	156
4.1.12	FILETIME64_TO_ISO8601.....	156
4.1.13	FILETIME64_TO_SYSTEMTIME.....	157
4.1.14	OTSTRUCT_TO_TIME.....	158
4.1.15	STRING_TO_SYSTEMTIME.....	158
4.1.16	SYSTEMTIME_TO_DT.....	159
4.1.17	SYSTEMTIME_TO_FILETIME64.....	159
4.1.18	SYSTEMTIME_TO_ISO8601.....	160
4.1.19	SYSTEMTIME_TO_STRING.....	160
4.1.20	TIME_TO_OTSTRUCT.....	161
4.2	Erweiterte STRING-Funktionen.....	162
4.2.1	CHAR_TO_WCHAR.....	162
4.2.2	CONCAT2.....	162
4.2.3	DATA_TO_HEXSTR2.....	163
4.2.4	DELETE2.....	164
4.2.5	F_StringIsASCII.....	164
4.2.6	FIND2.....	165
4.2.7	FindAndDelete.....	165
4.2.8	FindAndDeleteChar.....	166
4.2.9	FindAndReplace.....	167
4.2.10	FindAndReplaceChar.....	168
4.2.11	FindAndSplit.....	168
4.2.12	FindAndSplitChar.....	170
4.2.13	HEXSTR_TO_DATA2.....	172
4.2.14	INSERT2.....	172
4.2.15	LEN2.....	173
4.2.16	REPLACE2.....	174
4.2.17	sLiteral_TO_UTF8.....	174
4.2.18	STRING_TO_UTF8.....	175
4.2.19	STRING_TO_WSTRING2.....	176

4.2.20	STRNCPY	177
4.2.21	UTF8_TO_STRING.....	177
4.2.22	UTF8_TO_WSTRING	178
4.2.23	UTF8Len	179
4.2.24	WCHAR_TO_CHAR	180
4.2.25	WCONCAT2.....	180
4.2.26	WLEN2.....	181
4.2.27	wsLiteral_TO_UTF8.....	182
4.2.28	WSTRING_TO_STRING2.....	182
4.2.29	WSTRING_TO_UTF8	183
4.2.30	WSTRNCPY.....	184
4.3	Byte-Order-Konvertierungsfunktionen.....	184
4.3.1	Host Byte Order / Network Byte Order.....	184
4.3.2	HOST_TO_BE16	185
4.3.3	HOST_TO_BE32	185
4.3.4	HOST_TO_BE64	186
4.3.5	HOST_TO_BE64EX.....	186
4.3.6	HOST_TO_BE128	187
4.3.7	BE16_TO_HOST	187
4.3.8	BE32_TO_HOST	188
4.3.9	BE64_TO_HOST	188
4.3.10	BE64_TO_HOSTEX.....	188
4.3.11	BE128_TO_HOST	189
4.4	FLOAT-Funktionen.....	189
4.4.1	BOOL_TO_FLOAT.....	189
4.4.2	DINT_TO_FLOAT	190
4.4.3	FLOAT_TO_BOOL.....	190
4.4.4	FLOAT_TO_DINT	190
4.4.5	FLOAT_TO_INT.....	190
4.4.6	FLOAT_TO_SINT	191
4.4.7	FLOAT_TO_STRING	191
4.4.8	FLOAT_TO_TIME.....	191
4.4.9	FLOAT_TO_UDINT.....	192
4.4.10	FLOAT_TO_UINT	192
4.4.11	INT_TO_FLOAT.....	192
4.4.12	SINT_TO_FLOAT	192
4.4.13	TIME_TO_FLOAT	193
4.4.14	UDINT_TO_FLOAT.....	193
4.4.15	UINT_TO_FLOAT	193
4.4.16	LrealsFinite.....	194
4.4.17	LrealsNaN	194
4.4.18	RealsFinite	195
4.4.19	RealsNaN.....	195
4.5	LCOMPLEX-Funktionen.....	196
4.5.1	LcomplexIsNaN.....	196
4.5.2	LcomplexAbs.....	196

4.6	P[TYPE]_TO_[TYPE]-Konvertierungsfunktionen	196
4.6.1	PBOOL_TO_BOOL	196
4.6.2	PBYTE_TO_BYTE	197
4.6.3	PDATE_TO_DATE	197
4.6.4	PDINT_TO_DINT	197
4.6.5	PDT_TO_DT	198
4.6.6	PDWORD_TO_DWORD	198
4.6.7	PHUGE_TO_HUGE	198
4.6.8	PINT_TO_INT	199
4.6.9	PLARGE_TO_LARGE	199
4.6.10	PLINT_TO_LINT	199
4.6.11	PLREAL_TO_LREAL	200
4.6.12	PLWORD_TO_LWORD	200
4.6.13	PMAXSTRING_TO_MAXSTRING	201
4.6.14	PREAL_TO_REAL	201
4.6.15	PSINT_TO_SINT	201
4.6.16	PSTRING_TO_STRING	202
4.6.17	PTIME_TO_TIME	202
4.6.18	PTOD_TO_TOD	202
4.6.19	PUDINT_TO_UDINT	203
4.6.20	PUHUGE_TO_UHUGE	203
4.6.21	PUINT_TO_UINT	203
4.6.22	PULARGE_TO_ULARGE	204
4.6.23	PULINT_TO_ULINT	204
4.6.24	PUSINT_TO_USINT	204
4.6.25	PWORD_TO_WORD	205
4.6.26	PUINT64_TO_UINT64	205
4.7	16-Bit-Festkommazahl-Funktionen (signed)	205
4.7.1	FIX16_TO_LREAL	205
4.7.2	FIX16_TO_WORD	206
4.7.3	FIX16Add	206
4.7.4	FIX16Align	207
4.7.5	FIX16Div	208
4.7.6	FIX16Mul	208
4.7.7	FIX16Sub	209
4.7.8	LREAL_TO_FIX16	210
4.7.9	WORD_TO_FIX16	211
4.8	64-Bit-Funktionen (signed)	211
4.8.1	INT64_TO_LREAL	211
4.8.2	Int64Add64	212
4.8.3	Int64Add64Ex	212
4.8.4	Int64Cmp64	212
4.8.5	Int64Div64Ex	213
4.8.6	Int64isZero	213
4.8.7	Int64Negate	214
4.8.8	Int64Not	214

4.8.9	Int64Sub64.....	215
4.8.10	LARGE_INTEGER.....	215
4.8.11	LARGE_TO_LINT.....	215
4.8.12	LARGE_TO_ULONG.....	216
4.8.13	LINT_TO_LARGE.....	216
4.8.14	LREAL_TO_INT64.....	217
4.8.15	ULONG_TO_LARGE.....	217
4.9	64-Bit-Funktionen (unsigned).....	217
4.9.1	LREAL_TO_UINT64.....	217
4.9.2	LWORD_TO_ULONG.....	218
4.9.3	STRING_TO_UINT64.....	218
4.9.4	UInt32x32To64.....	218
4.9.5	UINT64_TO_LREAL.....	219
4.9.6	UINT64_TO_STRING.....	219
4.9.7	UInt64Add64.....	220
4.9.8	UInt64Add64Ex.....	220
4.9.9	UInt64And.....	220
4.9.10	UInt64Cmp64.....	221
4.9.11	UInt64Div16Ex.....	221
4.9.12	UInt64Div64.....	222
4.9.13	UInt64Div64Ex.....	222
4.9.14	UInt64isZero.....	223
4.9.15	UInt64Limit.....	223
4.9.16	UInt64Max.....	224
4.9.17	UInt64Min.....	224
4.9.18	UInt64Mod64.....	224
4.9.19	UInt64Mul64.....	225
4.9.20	UInt64Mul64Ex.....	225
4.9.21	UInt64Not.....	226
4.9.22	UInt64Or.....	226
4.9.23	UInt64Rol.....	226
4.9.24	UInt64Ror.....	227
4.9.25	UInt64Shl.....	227
4.9.26	UInt64Shr.....	228
4.9.27	UInt64Sub64.....	228
4.9.28	UInt64Xor.....	228
4.9.29	ULONG_INTEGER.....	229
4.9.30	ULONG_TO_ULINT.....	229
4.9.31	ULONG_TO_LWORD.....	230
4.10	T_Arg-Hilfsfunktionen.....	230
4.10.1	F_ARGCMP.....	230
4.10.2	F_ARGCPY.....	231
4.10.3	F_ARGISZERO.....	232
4.10.4	F_BIGTYPE.....	232
4.10.5	F_BOOL.....	233
4.10.6	F_BYTE.....	233

4.10.7	F_DINT.....	233
4.10.8	F_DWORD	234
4.10.9	F_HUGE.....	234
4.10.10	F_INT	235
4.10.11	F_LARGE.....	235
4.10.12	F_LINT	235
4.10.13	F_LREAL.....	236
4.10.14	F_LWORD.....	236
4.10.15	F_REAL.....	237
4.10.16	F_SINT	237
4.10.17	F_STRING	237
4.10.18	F_STRINGEx	238
4.10.19	F_UDINT	238
4.10.20	F_UHUGE	239
4.10.21	F_UINT.....	239
4.10.22	F_ULARGE	239
4.10.23	F_ULINT.....	240
4.10.24	F_USINT	240
4.10.25	F_WORD.....	240
4.10.26	F_PVOID.....	241
4.10.27	IsFinite.....	241
4.11	[veraltete Funktionen].....	243
4.11.1	Zeitfunktionen.....	243
4.11.2	F_GetVersionTcUtilities	246
4.11.3	FLOATIsFinite	247
4.11.4	FLOATIsNaN.....	247
4.12	ARG_TO_CSVFIELD	247
4.13	BIC_TO_BTN	249
4.14	BYTE_TO_BINSTR.....	250
4.15	BYTE_TO_DECSTR	250
4.16	BYTE_TO_HEXSTR	251
4.17	BYTE_TO_LREALEX.....	252
4.18	BYTE_TO_OCTSTR	253
4.19	BYTEARR_TO_MAXSTRING	253
4.20	CSVFIELD_TO_ARG	253
4.21	CSVFIELD_TO_STRING	254
4.22	DATA_TO_HEXSTR	256
4.23	DEG_TO_RAD.....	257
4.24	DINT_TO_DECSTR	258
4.25	DWORD_TO_BINSTR	259
4.26	DWORD_TO_DECSTR.....	259
4.27	DWORD_TO_HEXSTR.....	260
4.28	DWORD_TO_LREALEX.....	261
4.29	DWORD_TO_OCTSTR.....	262
4.30	F_BYTE_TO_CRC16_CCITT	263
4.31	F_CheckSum16	264

4.32	F_CreateHashTableHnd	264
4.33	F_CreateLinkedListHnd	265
4.34	F_DATA_TO_CRC16_CCITT	266
4.35	F_FormatArgToStr	266
4.36	F_GenerateHashValue	269
4.37	F_GetClassIdVersioned	269
4.38	F_LTrim	270
4.39	F_RTrim	270
4.40	F_SplitBIC	271
4.41	F_SwapRealEx	271
4.42	F_ToLCASE	272
4.43	F_ToUCASE	273
4.44	GUID_TO_REGSTRING	273
4.45	GUID_TO_STRING	274
4.46	GuidsEqualByVal	274
4.47	HEXASCNIBBLE_TO_BYTE	275
4.48	HEXCHRNIBBLE_TO_BYTE	275
4.49	HEXSTR_TO_DATA	276
4.50	LINT_TO_DECSTR	277
4.51	LREAL_TO_FMTSTR	278
4.52	LWORD_TO_BASE36STR	279
4.53	LWORD_TO_BINSTR	280
4.54	LWORD_TO_DECSTR	281
4.55	LWORD_TO_HEXSTR	282
4.56	LWORD_TO_OCTSTR	283
4.57	MAXSTRING_TO_BYTEARR	283
4.58	PVOID_TO_BINSTR	284
4.59	PVOID_TO_DECSTR	285
4.60	PVOID_TO_HEXSTR	286
4.61	PVOID_TO_OCTSTR	287
4.62	PVOID_TO_STRING	288
4.63	RAD_TO_DEG	289
4.64	REGSTRING_TO_GUID	289
4.65	ROUTETRANSPORT_TO_STRING	290
4.66	STRING_TO_CSVFIELD	290
4.67	STRING_TO_GUID	292
4.68	STRING_TO_PVOID	292
4.69	UDINT_TO_LREALEX	293
4.70	UINT_TO_LREALEX	294
4.71	ULINT_TO_ULONG	295
4.72	USINT_TO_LREALEX	295
4.73	WORD_TO_BINSTR	296
4.74	WORD_TO_DECSTR	297
4.75	WORD_TO_HEXSTR	297
4.76	WORD_TO_LREALEX	298
4.77	WORD_TO_OCTSTR	299

5	Datentypen	300
5.1	ADSDATATYPEID	300
5.2	E_AmsLoggerMode	300
5.3	E_ArgType	301
5.4	E_DbgContext	302
5.5	E_DbgDirection	302
5.6	E_EnumCmdType	303
5.7	E_HashMode	303
5.8	E_LDevType	304
5.9	E_LDongleStatus	304
5.10	E_LicenseHResult	304
5.11	E_MIB_IF_Type	305
5.12	E_NumGroupTypes	305
5.13	E_PersistentMode	306
5.14	E_RegValueType	306
5.15	E_RouteTransportType	307
5.16	E_SBCSType	307
5.17	E_ScopeServerState	308
5.18	E_TimeZoneID	308
5.19	E_TypeFieldParam	308
5.20	FLOAT	309
5.21	GUID	309
5.22	OTSTRUCT	310
5.23	PROFILERSTRUCT	310
5.24	REMOTEPC	310
5.25	REMOTEPCINFOSTRUCT	311
5.26	ST_AmsRouteEntry	311
5.27	ST_AmsRouteEntryEx	312
5.28	ST_CheckLicense	312
5.29	ST_DeviceIdentification	313
5.30	ST_DeviceIdentificationEx	313
5.31	ST_FileAttributes	314
5.32	ST_FileRBufferHead	315
5.33	ST_FindFileEntry	316
5.34	ST_IPAdapterHwAddr	317
5.35	ST_IPAdapterInfo	317
5.36	ST_LicenseDongle	318
5.37	ST_ReadEvent	318
5.38	ST_SplittedBIC	319
5.39	ST_TcOnlineLicenseInfoDataEx	320
5.40	ST_TcOnlineLicensesInfoData	320
5.41	ST_TcRouterStatusInfo	321
5.42	ST_TimeZoneInformation	321
5.43	SYMINFOSTRUCT	322
5.44	T_Arg	323
5.45	T_FILETIME	323

5.46	T_FILETIME64	323
5.47	T_FIX16	324
5.48	T_HashTableEntry	325
5.49	T_HHASHTABLE	325
5.50	T_HLINKEDLIST	325
5.51	T_HUGE_INTEGER.....	326
5.52	T_LARGE_INTEGER.....	326
5.53	T_LinkedListEntry	326
5.54	T_UHUGE_INTEGER.....	326
5.55	T_ULARGE_INTEGER	327
5.56	TIMESTRUCT	327
6	Globale Konstanten	329
6.1	Bibliotheksversion	329
7	Globale Variablen	330
8	Beispiele	331
8.1	Beispiel: Kommunikation BC/BX<->PC/CX (F_SwapRealEx)	331
8.2	Beispiel: Dateisuche (FB_EnumFindFileEntry, FB_EnumFindFileList).....	335
8.3	Beispiel: Datei-Ring-FiFo (FB_FileRingBuffer)	337
8.4	Beispiel: Memory-Ring-FiFo (FB_MemRingBuffer).....	339
8.5	Beispiel: Memory-Ring-FiFo (FB_MemRingBufferEx)	340
8.6	Beispiel: Hash-Tabelle (FB_HashTableCtrl)	341
8.7	Beispiel: Verkettete-Liste (FB_LinkedListCtrl).....	345
8.8	Beispiel: Schreiben/lesen einer CSV-Datei	351
8.9	Beispiel: Software-Uhren (RTC, RTC_EX, RTC_EX2)	354
9	Anhang	356
9.1	Systemverhalten beim Schreiben von persistenten Daten	356
9.2	Formatspezifikation	356
9.3	Format-Fehlercodes	359
9.4	Scope-Server-Fehlercodes	360
9.5	ADS Return Codes.....	360
9.6	Win32 Error Codes.....	364

1 Vorwort

1.1 Hinweise zur Dokumentation

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

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

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

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

Disclaimer

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

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

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

Marken

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

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

Patente

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

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702

mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Deutschland.

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

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 SPS-Bibliothek Tc2_Utilities umfasst Funktionsbausteine und Funktionen zum Aufruf von TwinCAT-Systemfunktionen und Betriebssystemfunktionen sowie verschiedene Konvertierungsfunktionen.

- [Betriebssystemfunktionen](#) [► 16]
- [SPS-Funktionen](#) [► 17]
- [Checksumme/CRC-Funktionen](#) [► 17]
- [Systemfunktionen](#) [► 17]
- [Scope-View-Funktionen](#) [► 17]
- [Scope-Server-Funktionen](#) [► 18]
- [ADS-Monitor-Funktionen](#) [► 18]
- [Konvertierungsfunktionen](#) [► 18]
- [STRING-Funktionen](#) [► 19]
- [Erweiterte STRING-Funktionen](#) [► 19]
- [64-Bit-Funktionen \(unsigned\)](#) [► 20]
- [64-Bit-Funktionen \(signed\)](#) [► 21]
- [16-Bit-Festkommazahl-Funktionen \(signed\)](#) [► 21]
- [Byte-Order-Konvertierungsfunktionen](#) [► 22]
- [FLOAT-Funktionen](#) [► 22]
- [LCOMPLEX-Funktionen](#) [► 23]
- [P\[TYPE\] TO \[TYPE\]-Konvertierungsfunktionen](#) [► 23]
- [T_Arg-Hilfsfunktionen](#) [► 23]
- [CSV-Formatfunktionen](#) [► 24]
- [Lizenzierungsfunktionen](#) [► 24]
- [Andere Funktionen](#) [► 25]

Betriebssystemfunktionen

Name	Beschreibung
NT_Shutdown [► 127]	Betriebssystem herunterfahren (shutdown).
NT_AbortShutdown [► 121]	Shutdown-Prozess unterbrechen.
NT_Reboot [► 123]	Restart des Betriebssystems durchführen.
NT_GetTime [► 122]	Aktuelle lokale Windows-Systemzeit lesen.
NT_SetLocalTime [► 124]	Aktuelle lokale Windows-Systemzeit setzen.
NT_StartProcess [► 128]	Windows-Anwendungen aus der SPS starten.
NT_SetTimeToRTCTime [► 125]	Die lokale Windows-Systemzeit mit der Echtzeituhr des PCs synchronisieren.
FB_RegQueryValue [► 101]	Werte aus der Registrierung auslesen.
FB_RegSetValue [► 104]	Werte in die Registrierung schreiben.
FB_EnumFindFileEntry [► 51]	Sucht in einem Verzeichnis nach einer Datei oder einem Unterverzeichnis, dessen Name dem spezifizierten Namen gleicht. Die gefundenen Einträge können einzeln ausgelesen werden.
FB_EnumFindFileList [► 53]	Sucht in einem Verzeichnis nach einer Datei oder einem Unterverzeichnis, dessen Name dem spezifizierten Namen gleicht. Die gefundenen Einträge können Gruppenweise ausgelesen werden.
FB_GetAdaptersInfo [► 67]	Liest Netzwerkadapterinformationen.
FB_GetHostName [► 72]	Liest den Host-Namen des lokalen PCs.
FB_GetHostAddrByName [► 71]	Konvertiert den Host-Namen in die (IPv4) Internet Protokoll Netzwerkadresse.

Name	Beschreibung
FB_GetTimeZoneInformation [▶ 78]	Liest die Zeitzone-Konfiguration des Betriebssystems.
FB_SetTimeZoneInformation [▶ 111]	Setzt die Zeitzone-Konfiguration des Betriebssystems.
FB_LocalSystemTime [▶ 90]	Liefert die lokale Windows-Systemzeit und Sommerzeit-/Winterzeit-Info.

SPS-Funktionen

Name	Beschreibung
PLC_Reset [▶ 134]	Reset der SPS durchführen.
PLC_Start [▶ 134]	SPS starten.
PLC_Stop [▶ 135]	SPS stoppen.
PLC_ReadSymInfo [▶ 130]	Symbolinformationen der SPS lesen.
PLC_ReadSymInfoByName [▶ 131]	Symbolinformationen einer SPS-Variablen anhand des Symbolnamens lesen.
PLC_ReadSymInfoByNameEx [▶ 132]	Symbolinformationen einer SPS-Variablen anhand des Symbolnamens lesen. Der Kommentar wird abgeschnitten wenn die verfügbare Puffergröße nicht ausreichend ist.
Profiler [▶ 136]	Ausführungszeit vom SPS-Code ermitteln.
WritePersistentData [▶ 146]	Persistente Daten auf den Datenträger aus dem SPS-Programm heraus speichern.
FB_WritePersistentData [▶ 119]	Persistente Daten auf den Datenträger aus dem SPS-Programm heraus speichern (erweiterte Version).

Checksumme/CRC-Funktionen

Name	Beschreibung
F_CheckSum16 [▶ 264]	Berechnet die 16-Bit-Check-Summe.
F_DATA_TO_CRC16_CCITT [▶ 266]	Berechnet CRC16-CCITT (zyklische Redundanz Prüfung) von einem beliebigen Datentyp.
F_BYTE_TO_CRC16_CCITT [▶ 263]	Berechnet CRC16-CCITT (zyklische Redundanz Prüfung) eines einzelnen Datenbytes.

Systemfunktionen

Name	Beschreibung
TC_Restart [▶ 144]	TwinCAT-System Restart durchführen.
TC_Stop [▶ 144]	TwinCAT-System stoppen.
TC_Config [▶ 142]	TwinCAT-System in den CONFIG-Modus versetzen.
TC_CpuUsage [▶ 142]	Die CPU-Auslastung des TwinCAT-Systems ermitteln.
TC_SysLatency [▶ 145]	Die aktuelle und maximale Latenzzeit eines TwinCAT-Systems ermitteln.
GetRemotePCInfo [▶ 120]	Router-Informationen über die konfigurierten Remote-PCs lesen.
FB_GetLocalAmsNetId [▶ 76]	AmsNetId des lokalen TwinCAT-PCs lesen.
FB_GetRouterStatusInfo [▶ 77]	Router-Statusinformationen lesen.
FB_EnumRouteEntry [▶ 54]	Router-Verbindungsinformationen lesen.
FB_AddRouteEntry [▶ 41]	Neue Router-Verbindung hinzufügen.
FB_RemoveRouteEntry [▶ 107]	Vorhandene Router-Verbindung löschen.
ROUTETRANSPORT_TO_STRING [▶ 290]	Konvertiert die AMS Message Router Transportschichtkennung in einen String.
FB_GetDeviceIdentification [▶ 26]	Liest die Geräteidentifikation ein.
FB_GetDeviceIdentificationEx [▶ 69]	Liest die Geräteidentifikation ein. (Erlaubt längere Strings bei Hardware Model und Seriennummer.)
FB_GetLicences [▶ 73]	Liest die gültigen und ungültigen TwinCAT Lizenzen aus.
FB_GetSystemId [▶ 78]	Liest die SystemID als GUID aus.
FB_GetVolumId [▶ 80]	Liest die SystemID und die Volume-System-ID aus.

Scope-View-Funktionen

Scope-View-Funktionen werden von TwinCAT 3 nicht mehr unterstützt.

Scope-Server-Funktionen

Name	Beschreibung
FB_ScopeServerControl [▶ 108]	Steuert (start/speichern...) den Scope Server für data logging.

ADS-Monitor-Funktionen

Name	Beschreibung
FB_AmsLogger [▶ 43]	AMS Logger aus der SPS starten/stoppen.

Konvertierungsfunktionen

Name	Beschreibung
DT_TO_SYSTEMTIME [▶ 149]	DATE_AND_TIME in Windows-Systemzeit-Struktur konvertieren.
DT_TO_FILETIME [▶ 243]	DATE_AND_TIME in Windows-File-Time konvertieren.
SYSTEMTIME_TO_DT [▶ 159]	Windows Systemzeit-Struktur in DATE_AND_TIME konvertieren.
SYSTEMTIME_TO_FILETIME [▶ 246]	Windows Systemzeit-Struktur in File-Time konvertieren.
SYSTEMTIME_TO_STRING [▶ 160]	Windows Systemzeit-Struktur in String konvertieren.
STRING_TO_SYSTEMTIME [▶ 158]	String in Windows Systemzeit-Struktur konvertieren.
FILETIME_TO_DT [▶ 244]	Windows File-Time in DATE_AND_TIME konvertieren.
FILETIME_TO_SYSTEMTIME [▶ 245]	Windows File-Time in Systemzeit-Struktur konvertieren.
DEC_TO_BCD [▶ 39]	Dezimal-Zahlen in BCD-Zahlen konvertieren.
BCD_TO_DEC [▶ 32]	BCD-Zahlen in Dezimal-Zahlen konvertieren.
DEG_TO_RAD [▶ 257]	Grad-Winkel in Bogenmaß konvertieren.
RAD_TO_DEG [▶ 289]	Bogenmaß in Grad-Winkel konvertieren.
TIME_TO_OTSTRUCT [▶ 161]	TIME-Variable in eine Struktur mit aufgelösten Millisekunden, Sekunden, Minuten usw. konvertieren.
OTSTRUCT_TO_TIME [▶ 158]	Struktur mit aufgelösten Millisekunden, Sekunden, Minuten usw. in TIME-Variable konvertieren.
F_SwapRealEx [▶ 271]	Tauscht um das Hi- und Lo-Word einer REAL-Variablen.
BYTE_TO_LREAL [▶ 252]	Erlaubt eine explizite Konvertierung vom Typ BYTE in eine positive Fließkommazahl vom Typ LREAL.
DWORD_TO_LREAL [▶ 261]	Erlaubt eine explizite Konvertierung vom Typ DWORD in eine positive Fließkommazahl vom Typ LREAL.
UDINT_TO_LREAL [▶ 293]	Erlaubt eine explizite Konvertierung vom Typ UDINT in eine positive Fließkommazahl vom Typ LREAL.
UINT_TO_LREAL [▶ 294]	Erlaubt eine explizite Konvertierung vom Typ UINT in eine positive Fließkommazahl vom Typ LREAL.
ULINT_TO_ULARGE [▶ 295]	Konvertiert eine 64 Bit Zahl vom Typ ULINT in eine 64 Bit Zahl vom Typ T_ULARGE_INTEGER.
USINT_TO_LREAL [▶ 295]	Erlaubt eine explizite Konvertierung vom Typ USINT in eine positive Fließkommazahl vom Typ LREAL.
BYTEARR_TO_MAXSTRING [▶ 253]	Konvertiert Byte Array in einen String.
MAXSTRING_TO_BYTEARR [▶ 283]	Konvertiert String in ein Byte Array.
F_TranslateFileTimeBias [▶ 243]	Konvertiert UTC-Zeit in Lokalzeit und umgekehrt (by bias).
FB_TzSpecificLocalTimeToFileTime [▶ 29]	Konvertiert kontinuierliche Lokalzeit (file time format) in UTC-Zeit.
FB_TzSpecificLocalTimeToSystemTime [▶ 117]	Konvertiert kontinuierliche Lokalzeit (structured system time format) in UTC-Zeit.
FB_FileTimeToTzSpecificLocalTime [▶ 27]	Konvertiert UTC-Zeit (file time format) in Lokalzeit.
FB_SystemTimeToTzSpecificLocalTime [▶ 113]	Konvertiert UTC-Zeit (structured system time format) in Lokalzeit.
HEXASCNIBBLE_TO_BYTE [▶ 275]	Konvertiert den ASCII-Code eines hexadezimalen Halbbytezeichens in den Dezimalwert.
HEXCHRNibble_TO_BYTE [▶ 275]	Konvertiert ein hexadezimalen Halbbytezeichen in seinen dezimalen Wert.
GuidsEqualByVal [▶ 274]	Vergleicht zwei GUID-Werte

STRING-Funktionen

Name	Beschreibung
LREAL_TO_FMTSTR [▶ 278]	Konvertiert eine Fließkommazahl in einen String mit der gewünschten Anzahl der Nachkommastellen.
DWORD_TO_DECSTR [▶ 259]	Konvertiert Dezimalzahl in einen Dezimalstring.
DWORD_TO_HEXSTR [▶ 260]	Konvertiert Dezimalzahl in einen Hexadezimalstring.
DWORD_TO_OCTSTR [▶ 262]	Konvertiert Dezimalzahl in einen Oktalstring.
DWORD_TO_BINSTR [▶ 259]	Konvertiert Dezimalzahl in einen Binärstring.
LWORD_TO_DECSTR [▶ 281]	Konvertiert Dezimalzahl in einen Dezimalstring.
LWORD_TO_HEXSTR [▶ 282]	Konvertiert Dezimalzahl in einen Hexadezimalstring.
LWORD_TO_OCTSTR [▶ 283]	Konvertiert Dezimalzahl in einen Oktalstring.
LWORD_TO_BINSTR [▶ 280]	Konvertiert Dezimalzahl in einen Binärstring.
PVOID_TO_DECSTR [▶ 285]	Konvertiert Adresse (Pointer) in einen Dezimalstring.
PVOID_TO_HEXSTR [▶ 286]	Konvertiert Adresse (Pointer) in einen Hexadezimalstring.
PVOID_TO_OCTSTR [▶ 287]	Konvertiert Adresse (Pointer) in einen Oktalstring.
PVOID_TO_BINSTR [▶ 284]	Konvertiert Adresse (Pointer) in einen Binärstring.
PVOID_TO_STRING [▶ 288]	Konvertiert Adresse (Pointer) in einen String.
STRING_TO_PVOID [▶ 292]	Konvertiert einen String in eine Adresse (Pointer).
LINT_TO_DECSTR [▶ 277]	Konvertiert vorzeichenbehaftete Dezimalzahl (64 Bit) in einen Dezimalstring.
DINT_TO_DECSTR [▶ 258]	Konvertiert vorzeichenbehaftete Dezimalzahl (32 Bit) in einen Dezimalstring.
F.FormatArgToStr [▶ 266]	Konvertiert und formatiert eine Dezimalzahl oder eine Fließkommazahl in einen String.
BYTE_TO_BINSTR [▶ 250]	Konvertiert eine Dezimalzahl vom Typ Byte in einen Binärstring.
BYTE_TO_DECSTR [▶ 250]	Konvertiert eine Dezimalzahl in einen Dezimalstring.
BYTE_TO_HEXSTR [▶ 251]	Konvertiert eine Dezimalzahl in einen Hexadezimalstring.
BYTE_TO_OCTSTR [▶ 253]	Konvertiert eine Dezimalzahl in einen Oktalstring.
WORD_TO_BINSTR [▶ 296]	Konvertiert eine Dezimalzahl vom Typ Word in einen Binärstring.
WORD_TO_DECSTR [▶ 297]	Konvertiert eine Dezimalzahl vom Typ Word in einen Dezimalstring.
WORD_TO_HEXSTR [▶ 297]	Konvertiert eine Dezimalzahl vom Typ Word in einen Hexadezimalstring.
WORD_TO_OCTSTR [▶ 299]	Konvertiert eine Dezimalzahl vom Typ Word in einen Oktalstring.
FB.FormatString [▶ 64]	Konvertiert und formatiert bis zu 10 Argumente (Dezimal- oder Fließkommazahlen).
FB.EnumStringNumbers [▶ 56]	Durchsucht einen String nach Zahlen.
F.ToUCase [▶ 273]	Konvertiert Kleinbuchstaben in Großbuchstaben in einem String.
F.ToLCase [▶ 272]	Konvertiert Großbuchstaben in Kleinbuchstaben in einem String.
F.LTrim [▶ 270]	Entfernt Leerzeichen am Anfang eines Strings.
F.RTrim [▶ 270]	Entfernt Leerzeichen am Ende eines Strings.
DATA_TO_HEXSTR [▶ 256]	Konvertiert Binärdaten in Hexadezimal-String.
HEXSTR_TO_DATA [▶ 276]	Konvertiert Hexadezimal-String in Binärdaten.
GUID_TO_STRING [▶ 274]	Konvertiert eine strukturierte GUID-Variable in eine GUID-Stringvariable.
GUID_TO_REGSTRING [▶ 273]	Konvertiert eine strukturierte GUID-Variable in eine Registry-GUID-Stringvariable
REGSTRING_TO_GUID [▶ 289]	Konvertiert eine Registry-GUID-Stringvariable in eine strukturierte GUID-Variable.

Erweiterte STRING-Funktionen

Name	Beschreibung
CHAR_TO_WCHAR [▶ 162]	Konvertiert ein Zeichen vom Typ STRING in ein Zeichen vom Typ WSTRING (mit Null-Terminierung).
CONCAT2 [▶ 162]	Verkettet zwei beliebig lange Zeichenketten vom Typ STRING.
DELETE2 [▶ 164]	Entfernt nLen Zeichen einer Zeichenfolge beginnend an der Position nPos.

Name	Beschreibung
F_StringIsASCII [▶ 164]	Prüft, ob eine Zeichenfolge nur ASCII-Zeichen (0x000 bis 0x7F) enthält und gibt die Anzahl der ASCII-Zeichen zurück.
FIND2 [▶ 165]	Findet eine ggf. mehrfach vorkommende Zeichenfolge in einer anderen Zeichenfolge.
FindAndDelete [▶ 165]	Findet eine ggf. mehrfach vorkommende Zeichenfolge in einer anderen Zeichenfolge und entfernt diese.
FindAndDeleteChar [▶ 166]	Findet ein ggf. mehrfach vorkommendes Zeichen in einer Zeichenfolge und entfernt dieses.
FindAndReplace [▶ 167]	Findet eine ggf. mehrfach vorkommende Zeichenfolge in einer anderen Zeichenfolge und ersetzt diese durch eine andere Zeichenfolge.
FindAndReplaceChar [▶ 168]	Findet ein ggf. mehrfach vorkommendes Zeichen in einer Zeichenfolge und ersetzt dieses durch ein anderes Zeichen.
INSERT2 [▶ 172]	Fügt eine Zeichenfolge in eine andere Zeichenfolge nach der Position nPos ein.
LEN2 [▶ 173]	Gibt die Anzahl der Zeichen in einer Zeichenfolge zurück.
REPLACE2 [▶ 174]	Ersetzt nLen Zeichen einer Zeichenfolge durch eine andere Zeichenfolge beginnend an der Position nPos.
sLiteral_TO_UTF8 [▶ 174]	Konvertiert eine beliebige Zeichenfolge vom Datentyp STRING in eine Zeichenfolge im UTF-8-Format.
STRING_TO_UTF8 [▶ 175]	Konvertiert eine beliebige Zeichenfolge einer Variablen vom Datentyp STRING in eine Zeichenfolge im UTF-8-Format.
STRING_TO_WSTRING2 [▶ 176]	Konvertiert eine Variable vom Datentyp STRING in eine Variable vom Datentyp WSTRING.
STRNCPY [▶ 177]	Kopiert die Zeichenfolge einer Variablen vom Datentyp STRING und prüft, ob die Zeichenfolge vollständig kopiert wurde.
UTF8_TO_STRING [▶ 177]	Konvertiert eine Zeichenfolge im UTF8-Format in eine Zeichenfolge vom Datentyp STRING.
UTF8_TO_WSTRING [▶ 178]	Konvertiert eine Zeichenfolge im UTF-8-Format in eine Zeichenfolge vom Datentyp WSTRING.
UTF8Len [▶ 179]	Gibt die Anzahl der Zeichen einer UTF-8-Zeichenfolge zurück.
WCHAR_TO_CHAR [▶ 180]	Konvertiert eine Variable vom Datentyp WSTRING in eine Variable vom Datentyp STRING (mit Null-Terminierung).
WCONCAT2 [▶ 180]	Verkettet zwei beliebig lange Zeichenfolgen vom Datentyp WSTRING.
WLEN2 [▶ 181]	Gibt die Anzahl der Zeichen in einer Unicode-Zeichenfolge vom Datentyp WSTRING zurück.
wsLiteral_TO_UTF8 [▶ 182]	Konvertiert eine beliebige Zeichenfolge vom Datentyp WSTRING in eine Zeichenfolge im UTF-8-Format.
WSTRING_TO_STRING2 [▶ 182]	Konvertiert eine Variable vom Datentyp WSTRING in eine Variable vom Datentyp STRING.
WSTRING_TO_UTF8 [▶ 183]	Konvertiert eine Zeichenfolge einer Variablen vom Datentyp WSTRING in eine Zeichenfolge im UTF-8-Format.
WSTRNCPY [▶ 184]	Kopiert eine Zeichenfolge einer Variablen vom Datentyp WSTRING und prüft, ob die Zeichenfolge vollständig kopiert wurde.

64-Bit-Funktionen (unsigned)

Name	Beschreibung
ULARGE_INTEGER [▶ 229]	Initialisiert/setzt eine 64-Bit-Zahl.
UInt64Add64 [▶ 220]	Addiert zwei 64-Bit-Zahlen.
UInt64Add64Ex [▶ 220]	Addiert zwei 64-Bit-Zahlen (mit Overflow check).
UInt64Sub64 [▶ 228]	Subtrahiert zwei 64-Bit-Zahlen.
UInt64Cmp64 [▶ 221]	Vergleicht zwei 64-Bit-Zahlen.
UInt32x32To64 [▶ 218]	Multipliziert zwei 32-Bit-Zahlen. Das Ergebnis ist eine 64-Bit-Zahl.
UInt64Mul64 [▶ 225]	Multipliziert zwei 64-Bit-Zahlen. Das Ergebnis ist eine 64-Bit-Zahl.
UInt64Mul64Ex [▶ 225]	Multipliziert zwei 64-Bit-Zahlen. Das Ergebnis ist eine 64-Bit-Zahl (mit Overflow check).
UInt64Div64 [▶ 222]	Division zweier 64-Bit-Zahlen.
UInt64Div64Ex [▶ 222]	Division zweier 64-Bit-Zahlen (mit Restergebnis).
UInt64Div16Ex [▶ 221]	Division einer 64-Bit-Zahl durch eine 16-Bit-Zahl. Das Ergebnis ist eine 64-Bit-Zahl.

Name	Beschreibung
UInt64Mod64 [▶ 224]	Modulo-Division zweier 64-Bit-Zahlen.
UInt64And [▶ 220]	Bitweise AND zweier 64-Bit-Zahlen.
UInt64Or [▶ 226]	Bitweise OR zweier 64-Bit-Zahlen.
UInt64Not [▶ 226]	Bitweise NOT einer 64-Bit-Zahlen.
UInt64Xor [▶ 228]	Bitweise XOR zweier 64-Bit-Zahlen.
UInt64Rol [▶ 226]	Bitweise Linksrotation einer 64-Bit-Zahl.
UInt64Ror [▶ 227]	Bitweise Rechtsrotation einer 64-Bit-Zahl.
UInt64Shl [▶ 227]	Bitweises Links-Shift einer 64-Bit-Zahl.
UInt64Shr [▶ 228]	Bitweises Rechts-Shift einer 64-Bit-Zahl.
UInt64Min [▶ 224]	Minimumfunktion
UInt64Max [▶ 224]	Maximumfunktion
UInt64Limit [▶ 223]	Limitierung
UInt64isZero [▶ 223]	Prüft ob der Wert der 64-Bit-Zahl ist Null.
UINT64 TO STRING [▶ 219]	Konvertiert 64-Bit-Zahl in einen STRING.
UINT64 TO LREAL [▶ 219]	Konvertiert 64-Bit-Zahl in einen LREAL.
STRING TO UINT64 [▶ 218]	Konvertiert einen STRING in eine 64-Bit-Zahl.
LREAL TO UINT64 [▶ 217]	Konvertiert LREAL in eine 64-Bit-Zahl.
LWORD TO ULARGE [▶ 218]	Konvertiert eine 64-Bit-Zahl vom Typ LWORD in eine 64-Bit-Zahl vom Typ T_ULARGE_INTEGER.
ULARGE TO LWORD [▶ 230]	Konvertiert eine 64-Bit-Zahl vom Typ T_ULARGE_INTEGER in eine 64-Bit-Zahl vom Typ LWORD.
ULARGE TO ULINT [▶ 229]	Konvertiert eine 64-Bit-Zahl vom Typ T_ULARGE_INTEGER in eine 64 Bit Zahl ULINT.

64-Bit-Funktionen (signed)

Name	Beschreibung
LARGE_INTEGER [▶ 215]	Initialisiert/setzt eine 64-Bit-Zahl.
Int64Add64 [▶ 212]	Addiert zwei 64-Bit-Zahlen.
Int64Add64Ex [▶ 212]	Addiert zwei 64-Bit-Zahlen (mit Overflow check).
Int64Sub64 [▶ 215]	Subtrahiert zwei 64-Bit-Zahlen.
Int64Cmp64 [▶ 212]	Vergleicht zwei 64-Bit-Zahlen.
Int64Div64Ex [▶ 213]	Division zweier 64-Bit-Zahlen (mit Restergebnis).
Int64Not [▶ 214]	Bitweise NOT einer 64-Bit-Zahl.
Int64isZero [▶ 213]	Prüft ob der Wert der 64-Bit-Zahl ist Null.
Int64Negate [▶ 214]	Negiert eine 64-Bit-Zahl.
INT64 TO LREAL [▶ 211]	Konvertiert 64-Bit-Zahl in einen LREAL.
LREAL TO INT64 [▶ 217]	Konvertiert LREAL in eine 64 Bit Zahl.
LARGE TO ULARGE [▶ 216]	Konvertiert eine vorzeichenbehaftete 64-Bit-Zahl in eine vorzeichenlose 64-Bit-Zahl.
ULARGE TO LARGE [▶ 217]	Konvertiert eine vorzeichenlose 64-Bit-Zahl in eine vorzeichenbehaftete 64-Bit-Zahl.
LARGE TO LINT [▶ 215]	Konvertiert eine vorzeichenbehaftete 64-Bit-Zahl vom Typ LINT in eine vorzeichenbehaftete 64-Bit-Zahl vom Typ T_LARGE_INTEGER).
LINT TO LARGE [▶ 216]	Konvertiert eine vorzeichenbehaftete 64-Bit-Zahl vom Typ T_LARGE_INTEGER in eine vorzeichenbehaftete 64-Bit-Zahl vom Typ LINT.

16-Bit-Festkommazahl-Funktionen (signed)

Name	Beschreibung
FIX16Add [▶ 206]	Addiert zwei Festkommazahlen.
FIX16Align [▶ 207]	Ändert die Auflösung einer Festkommazahl.
FIX16Sub [▶ 209]	Subtrahiert zwei Festkommazahlen.
FIX16Div [▶ 208]	Dividiert zwei Festkommazahlen.
FIX16Mul [▶ 208]	Multipliziert zwei Festkommazahlen.

Name	Beschreibung
LREAL TO FIX16 [▶ 210]	Konvertiert LREAL in eine Festkommazahl.
WORD TO FIX16 [▶ 211]	Konvertiert WORD in eine Festkommazahl.
FIX16 TO LREAL [▶ 205]	Konvertiert eine Festkommazahl in LREAL.
FIX16 TO WORD [▶ 206]	Konvertiert eine Festkommazahl in WORD.

Byte-Order-Konvertierungsfunktionen

Name	Beschreibung
HOST TO BE16 [▶ 185]	Host-To-Network Konvertierung (16-Bit-Zahl)
HOST TO BE32 [▶ 185]	Host-To-Network Konvertierung (32-Bit-Zahl)
HOST TO BE64 [▶ 186]	Host-To-Network Konvertierung (64-Bit-Zahl, „legacy“-Typ: T_ULARGE_INTEGER)
HOST TO BE64EX [▶ 186]	Host-To-Network Konvertierung (64-Bit-Zahl, „native“-Typ: LWORD)
HOST TO BE128 [▶ 187]	Host-To-Network Konvertierung (128-Bit-Zahl, „legacy“-Typ: T_UHUGE_INTEGER)
BE16 TO HOST [▶ 187]	Network-To-Host Konvertierung (16-Bit-Zahl)
BE32 TO HOST [▶ 188]	Network-To-Host Konvertierung (32-Bit-Zahl)
BE64 TO HOST [▶ 188]	Network-To-Host Konvertierung (64-Bit-Zahl, „legacy“-Typ: T_ULARGE_INTEGER)
BE64 TO HOSTEX [▶ 188]	Network-To-Host Konvertierung (64-Bit-Zahl, „native“-Typ: LWORD)
BE128 TO HOST [▶ 189]	Network-To-Host Konvertierung (128-Bit-Zahl, „legacy“-Typ: T_UHUGE_INTEGER)

FLOAT-Funktionen

Name	Beschreibung
BOOL TO FLOAT [▶ 189]	Konvertiert eine Variable vom Typ BOOL in eine Variable vom Typ LREAL.
DINT TO FLOAT [▶ 190]	Konvertiert eine Variable vom Typ DINT in eine Variable vom Typ FLOAT.
FLOAT TO BOOL [▶ 190]	Konvertiert eine Variable vom Typ FLOAT in eine Variable vom Typ BOOL.
FLOAT TO DINT [▶ 190]	Konvertiert eine Variable vom Typ FLOAT in eine Variable vom Typ DINT.
FLOAT TO INT [▶ 190]	Konvertiert eine Variable vom Typ FLOAT in eine Variable vom Typ INT.
FLOAT TO SINT [▶ 191]	Konvertiert eine Variable vom Typ FLOAT in eine Variable vom Typ SINT.
FLOAT TO STRING [▶ 191]	Konvertiert eine Variable vom Typ FLOAT in eine Variable vom Typ STRING.
FLOAT TO TIME [▶ 191]	Konvertiert eine Variable vom Typ FLOAT in eine Variable vom Typ TIME.
FLOAT TO UDINT [▶ 192]	Konvertiert eine Variable vom Typ FLOAT in eine Variable vom Typ UDINT.
FLOAT TO UINT [▶ 192]	Konvertiert eine Variable vom Typ FLOAT in eine Variable vom Typ UINT.
INT TO FLOAT [▶ 192]	Konvertiert eine Variable vom Typ INT in eine Variable vom Typ FLOAT.
SINT TO FLOAT [▶ 192]	Konvertiert eine Variable vom Typ SINT in eine Variable vom Typ FLOAT.
TIME TO FLOAT [▶ 193]	Konvertiert eine Variable vom Typ TIME in eine Variable vom Typ FLOAT.
UDINT TO FLOAT [▶ 193]	Konvertiert eine Variable vom Typ UDINT in eine Variable vom Typ FLOAT.
UINT TO FLOAT [▶ 193]	Konvertiert eine Variable vom Typ UINT in eine Variable vom Typ FLOAT.
LrealsFinite [▶ 194]	Liefert TRUE, wenn das Argument vom Typ LREAL einen endlichen Wert besitzt.
LrealsNaN [▶ 194]	Liefert TRUE, wenn das Argument vom Typ LREAL einen undefinierten Wert besitzt (NaN).

LCOMPLEX-Funktionen

LcomplexIsNaN [► 196]	Liefert TRUE, wenn das Argument vom Typ LCOMPLEX einen undefinierten Wert besitzt (NaN).
LcomplexAbs [► 196]	Liefert den Absolutwert zur übergebenen komplexen Zahl.

P[TYPE]_TO_[TYPE]-Konvertierungsfunktionen

Name	Beschreibung
PBOOL_TO_BOOL [► 196]	Liefert den Inhalt einer BOOL-Pointervariablen.
PBYTE_TO_BYTE [► 197]	Liefert den Inhalt einer BYTE-Pointervariablen.
PDATE_TO_DATE [► 197]	Liefert den Inhalt einer DATE-Pointervariablen.
PDINT_TO_DINT [► 197]	Liefert den Inhalt einer DINT-Pointervariablen.
PDT_TO_DT [► 198]	Liefert den Inhalt einer DT-Pointervariablen.
PDWORD_TO_DWORD [► 198]	Liefert den Inhalt einer DWORD-Pointervariablen.
PHUGE_TO_HUGE [► 198]	Liefert den Inhalt einer T_HUGE_INTEGER-Pointervariablen.
PINT_TO_INT [► 199]	Liefert den Inhalt einer INT-Pointervariablen.
PLARGE_TO_LARGE [► 199]	Liefert den Inhalt einer T_LARGE_INTEGER-Pointervariablen.
PLINT_TO_LINT [► 199]	Liefert den Inhalt einer LINT-Pointervariablen.
PLREAL_TO_LREAL [► 200]	Liefert den Inhalt einer LREAL-Pointervariablen.
PLWORD_TO_LWORD [► 200]	Liefert den Inhalt einer LWORD-Pointervariablen.
PMAXSTRING_TO_MAXSTRING [► 201]	Liefert den Inhalt einer T_MaxString-Pointervariablen.
PREAL_TO_REAL [► 201]	Liefert den Inhalt einer REAL-Pointervariablen.
PSINT_TO_SINT [► 201]	Liefert den Inhalt einer SINT-Pointervariablen.
PSTRING_TO_STRING [► 202]	Liefert den Inhalt einer STRING-Pointervariablen.
PTIME_TO_TIME [► 202]	Liefert den Inhalt einer TIME-Pointervariablen.
PTOD_TO_TOD [► 202]	Liefert den Inhalt einer TOD-Pointervariablen.
PUDINT_TO_UDINT [► 203]	Liefert den Inhalt einer UDINT-Pointervariablen.
PUHUGE_TO_UHUGE [► 203]	Liefert den Inhalt einer T_UHUGE_INTEGER-Pointervariablen.
PUINT_TO_UINT [► 203]	Liefert den Inhalt einer UINT-Pointervariablen.
PULARGE_TO_ULARGE [► 204]	Liefert den Inhalt einer T_ULARGE_INTEGER-Pointervariablen.
PULINT_TO_ULINT [► 204]	Liefert den Inhalt einer ULINT-Pointervariablen.
PUSINT_TO_USINT [► 204]	Liefert den Inhalt einer USINT-Pointervariablen.
PWORD_TO_WORD [► 205]	Liefert den Inhalt einer WORD-Pointervariablen.
PUINT64_TO_UINT64 [► 205]	Liefert den Inhalt einer T_ULARGE_INTEGER-Pointervariablen.

T_Arg-Hilfsfunktionen

Name	Beschreibung
F_ARGCMP [► 230]	Vergleicht zwei Variablen vom Typ T_Arg
F_ARGCPY [► 231]	Kopiert den Wert einer Variablen vom Typ T_Arg in eine andere Variable und liefert die Anzahl der erfolgreich kopierten Datenbytes zurück.
F_ARGISZERO [► 232]	Liefert TRUE, wenn eine der T_Arg-Membervariablen den Wert Null hat oder nicht initialisiert wurde.
F_BIGTYPE [► 232]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer Struct- oder Array-Variablen.
F_BOOL [► 233]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer BOOL-Variablen.
F_BYTE [► 233]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer BYTE-Variablen.
F_DINT [► 233]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer DINT-Variablen.
F_DWORD [► 234]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer DWORD-Variablen.
F_HUGE [► 234]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer T_HUGE_INTEGER-Variablen.
F_INT [► 235]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer INT-Variablen.

Name	Beschreibung
F_LARGE [► 235]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer T_LARGE_INTEGER-Variablen.
F_LINT [► 235]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer LINT-Variablen.
F_LREAL [► 236]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer LREAL-Variablen.
F_LWORD [► 236]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer LWORD-Variablen.
F_REAL [► 237]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer REAL-Variablen.
F_SINT [► 237]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer SINT-Variablen.
F_STRING [► 237]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer T_MaxString-Variablen.
F_UDINT [► 238]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer UDINT-Variablen.
F_UHUGE [► 239]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer T_UHUGE_INTEGER-Variablen.
F_UINT [► 239]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer UINT-Variablen.
F_ULARGE [► 239]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer T_ULARGE_INTEGER-Variablen.
F_ULINT [► 240]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer ULINT-Variablen.
F_USINT [► 240]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer USINT-Variablen.
F_WORD [► 240]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer WORD-Variablen.
F_PVOID [► 241]	Liefert in einer Struktur vom Typ T_Arg Informationen zu einer PVOID-Variablen.

CSV-Formatfunktionen

Name	Beschreibung
CSVFIELD_TO_STRING [► 254]	Konvertiert den Wert eines Strings mit einem Datenfeld im CSV-Format in eine SPS-Stringvariable.
STRING_TO_CSVFIELD [► 290]	Konvertiert den Wert einer SPS-Stringvariablen in einen String mit einem Datenfeld im CSV-Format.
CSVFIELD_TO_ARG [► 253]	Konvertiert einen Bytepuffer mit einem Datenfeld im CSV-Format in einen Wert einer beliebigen SPS-Variablen.
ARG_TO_CSVFIELD [► 247]	Konvertiert den Wert einer beliebigen SPS-Variablen in einen Bytepuffer mit einem Datenfeld im CSV-Format.
FB_CSVMemBufferReader [► 49]	Teilt Datensätze im CSV-Format die in einem Bytepuffer vorliegen in einzelne Datenfelder.
FB_CSVMemBufferWriter [► 50]	Erzeugt aus einzelnen Datenfeldern einzelne oder mehrere Datensätze in einem Bytepuffer

Lizenzierungsfunktionen

Name	Beschreibung
FB_LicFileGetStorageInfo [► 86]	Liest die StorageInfo des Lizenz-Dongles und das Dateiverzeichnis aus.
FB_LicFileCreate [► 84]	Erzeugt eine Datei auf dem Lizenz-Dongle.
FB_LicFileDelete [► 85]	Löscht eine Datei auf dem Lizenz-Dongle.
FB_LicFileRead [► 87]	Liest eine Datei vom Lizenz-Dongle in einen bereitgestellten Buffer.
FB_LicFileCopyToDongle [► 83]	Kopiert eine Datei von der Festplatte auf den Lizenz-Dongle.
FB_LicFileCopyFromDongle [► 82]	Kopiert eine Datei vom Lizenz-Dongle auf die Festplatte.
FB_CheckLicense [► 48]	Ermittelt den TwinCAT-3-Lizenzstatus für eine gegebene Lizenz-ID.
FB_GetDongleSystemId [► 70]	Liest die System-ID und die Volume-ID des TwinCAT-3-Lizenz-Dongles als GUID aus.
FB_GetLicenseDongle [► 73]	Ermittelt die Anzahl angeschlossener Lizenz-Dongles und liefert Adresse und Status.

Name	Beschreibung
FB_GetLicenses [▶ 73]	Liest die gültigen und ungültigen TwinCAT-Lizenzen aus.
FB_GetLicensesEx [▶ 75]	Ermittelt den Status aller TwinCAT-3-Lizenzen und OEM-Lizenzen.

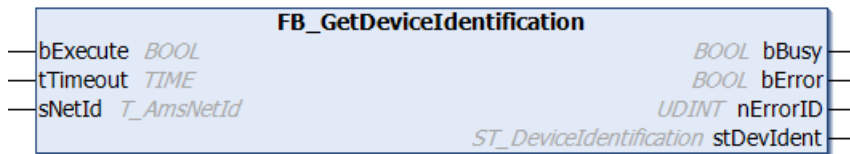
Andere Funktionen

Name	Beschreibung
FB_BasicPID [▶ 45]	Einfacher PID controller
F_GetVersionTcUtilities [▶ 246]	Versionsinformationen der Bibliothek lesen.
IsFinite [▶ 241]	Überprüft die Formatierung einer Gleitkommazahl nach der IEEE.
F_YearIsLeapYear [▶ 155]	Ermittelt ob ein Jahr Schaltjahr ist.
F_GetMaxMonthDays [▶ 152]	Ermittelt die maximale Anzahl der Monatstage.
F_GetDOYOfYearMonthDay [▶ 151]	Ermittelt die Nummer des Tages im Jahr.
F_GetMonthOfDOY [▶ 153]	Ermittelt den Monat anhand der Nummer des Tages im Jahr.
F_GetDayOfWeek [▶ 151]	Ermittelt die Nummer des Wochentages.
F_GetWeekOfYear [▶ 153]	Ermittelt die Kalenderwoche.
F_GetDayOfMonthEx [▶ 150]	Ermittelt das Datum des ersten, zweiten usw. Wochentages in einem bestimmten Monat und Jahr.
F_GetWeekOfYear [▶ 153]	Liefert die Nummer der Kalenderwoche zu einem vorgegebenen Datum.
RTC [▶ 138]	"Software"-RTC (Real Time Clock)
RTC_EX [▶ 139]	"Software"-RTC (Real Time Clock)
RTC_EX2 [▶ 140]	"Software"-RTC (Real Time Clock)
FB_FileRingBuffer [▶ 59]	Schreibt/liest Datensätze in oder von der Datei (FIFO).
FB_MemRingBuffer [▶ 96]	Schreibt/liest Datensätze in oder von einer Puffervariable (FIFO).
FB_MemRingBufferEx [▶ 97]	Schreibt/liest Datensätze in oder von einer Puffervariable (FIFO).
FB_StringRingBuffer [▶ 112]	Schreibt/liest Strings in oder von einer Puffervariable (FIFO).
FB_MemStackBuffer [▶ 99]	Schreibt/liest Datensätze in oder von einer Puffervariable (LIFO).
FB_MemBufferMerge [▶ 93]	Fügt einzelne kleinere Datensegmente zu einem größeren Datensegment zusammen.
FB_MemBufferSplit [▶ 95]	Teilt einen Speicherbereich (Datenpuffer) in mehrere kleinere Segmente auf.
FB_HashTableCtrl [▶ 80], F_CreateHashTableHnd [▶ 264]	Einfache Hash-Tabelle.
FB_LinkedListCtrl [▶ 88], F_CreateLinkedListHnd [▶ 265]	Einfache verkettete Liste (doppelt verkettet).
DCF77_TIME [▶ 32]	Einfacher DCF77-Dekoder.
DCF77_TIME_EX [▶ 36]	DCF77-Dekoder mit Plausibilitätsprüfung von zwei aufeinanderfolgenden Telegrammen und Zeitzoneinformation.

3 Funktionsbausteine

3.1 [veraltet]

3.1.1 FB_GetDeviceIdentification



Der Baustein liest die Geräteidentifikation ein.

i Veraltete Funktionalität

Für längere Strings bei Hardware Modell und Hardware Seriennummer muss der [FB_GetDeviceIdentificationEx \[► 69\]](#) verwendet werden.

VAR_INPUT

```
VAR_INPUT
  bExecute   : BOOL;
  tTimeout   : TIME := DEFAULT_ADS_TIMEOUT;
  sNetId     : T_AmsNetId;
END_VAR
```

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

sNetId: Hier kann die AmsNetId des TwinCAT-Rechners angegeben werden, dessen Geräteidentifikation gelesen werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorId   : UDINT;
  stDevIdent : ST_DeviceIdentification;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

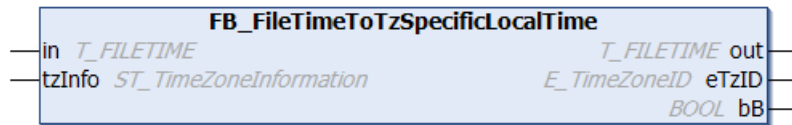
nErrorId: Liefert bei einem gesetzten bError-Ausgang die [ADS-Fehlernummer \[► 360\]](#).

stDevIdent: Liefert die Geräteidentifikation (Typ: [ST_DeviceIdentification \[► 313\]](#))

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.1.2 FB_FileTimeToTzSpecificLocalTime



Veraltete Funktion

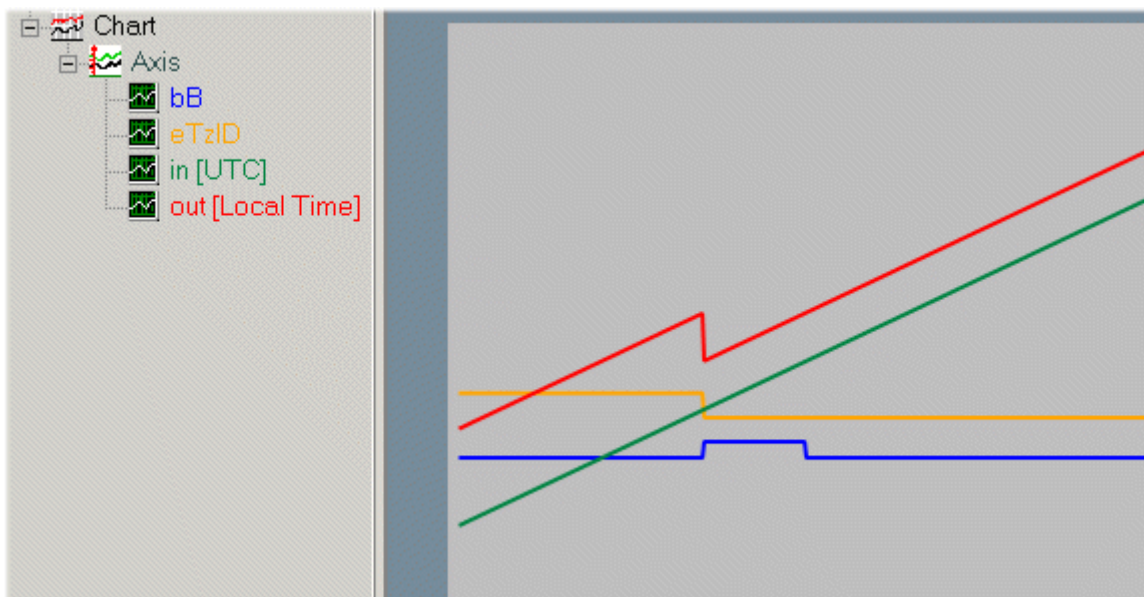
i Dieser Funktionsbaustein ist veraltet. Verwenden Sie stattdessen den Funktionsbaustein [FB_FileTime64ToTzSpecificLocalTime](#) [► 62].

Der Funktionsbaustein konvertiert die UTC-Zeit (file time format) in Lokalzeit (file time format) unter der Berücksichtigung der angegebenen Zeitzoneinformation. Der Funktionsbaustein: [FB_SystemTimeToTzSpecificLocalTime](#) [► 113] besitzt eine ähnliche Funktionalität, mit dem Unterschied, dass er ein anderes Zeitformat (structured system time format) umrechnet.

Der Baustein eignet sich nur zur Konvertierung von **kontinuierlichen** UTC-Zeitstempelinformation. Anhand der Zeitzoneinformation errechnet der Funktionsbaustein die erforderlichen Zeitsprünge in der Lokalzeit (Sommer-/Winterzeit Umstellung). Zeitsprünge in der UTC-Eingangszeit sind nicht zulässig und führen zur fehlerhaften Konvertierung. Der Grund: Die zuletzt konvertierte Zeit wird im Baustein intern gespeichert, um aus dem Verlauf der UTC-Eingangszeit und dem gespeicherten Wert die B-Zeiten (siehe unten) beim Zurückstellen der Lokalzeit erkennen zu können.

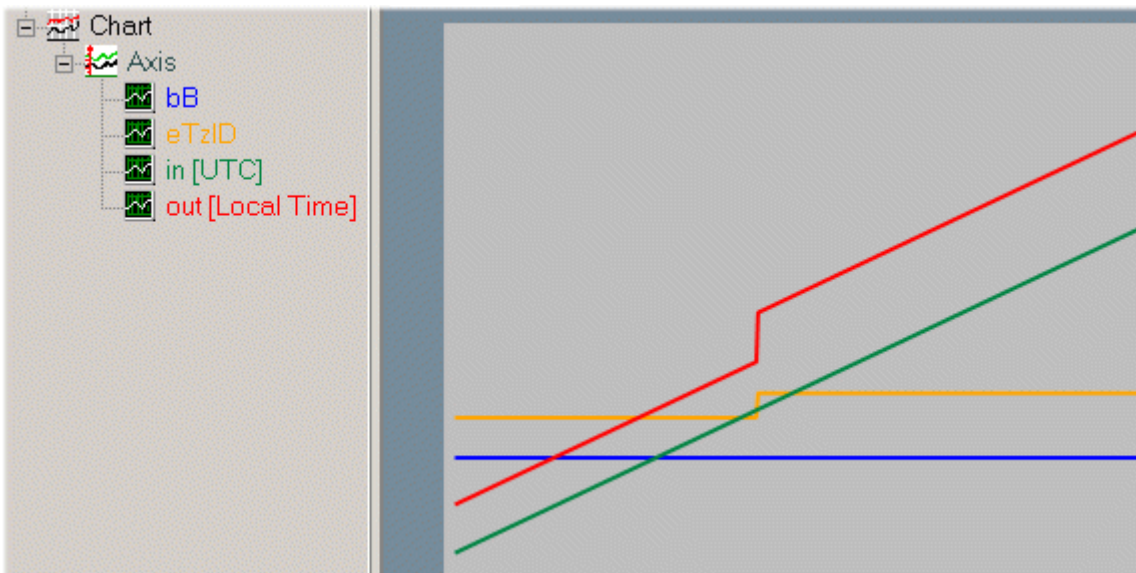
Der Baustein besitzt eine Aktion: A_Reset(). Ein Aufruf dieser Aktion setzt die Ausgänge des Bausteins und die intern gespeicherte (zuletzt konvertierte Zeit) auf Null zurück.

1. Grafische Darstellung des zeitlichen Verhaltens beim Übergang von Sommerzeit zur Winterzeit (hier tzInfo = WEST_EUROPE_TZI):



Die UTC-Eingangszeit (grün) verläuft kontinuierlich. Die Lokalzeit (rot) macht einen Sprung zurück. Der Lokalzeit: **02h:59m:59s:999ms..** folgt unmittelbar die Zeit: **02h:00m:00s:000ms..** Die Zeiten zwischen 2h und 3h sind zweifach vorhanden. Die doppelte Zeit vor der Zeitumstellung wird z. B. als **02:05:00 MESZ A** bezeichnet und die Zeit nach der Umstellung als **02:05:00 MEZ B**. Die Ausgangsvariable *bB* gibt Information darüber, ob es sich um den ersten oder zweiten *Durchgang* handelt. Beim zweiten *Durchgang* ist die *bB*-Ausgangsvariable (blau) auf TRUE gesetzt. Die *bB*-Ausgangsvariable wird automatisch zurückgesetzt, nachdem die doppelte Zeit überschritten wurde. Die Zeitzone-ID (orange) wechselt von *eTimeZoneID_Daylight* (Sommerzeit) zu *eTimeZoneID_Standard* (Winterzeit).

2. Grafische Darstellung des zeitlichen Verhaltens beim Übergang von Winterzeit zur Sommerzeit (hier tzInfo = WEST_EUROPE_TZI):



Die UTC-Eingangszeit (grün) verläuft kontinuierlich. Die Lokalzeit (grün) macht einen Sprung nach vorne. Der Lokalzeit: **2h:59m:59s:999ms..** folgt unmittelbar die Zeit: **3h:00m:00s:000ms..** Die Zeitzonen-ID (orange) wechselt von *eTimeZoneID_Standard* (Winterzeit) zu *eTimeZoneID_Daylight* (Sommerzeit).

VAR_INPUT

```
VAR_INPUT
  in      : T_FILETIME;
  tzInfo  : ST_TimeZoneInformation;
END_VAR
```

in: UTC-Zeit (file time format), die konvertiert werden soll (Typ: [T_FILETIME](#) [[▶ 323](#)]).

tzInfo: Strukturvariable mit der aktuellen Zeitzone-Information des Betriebssystems (Typ: [ST_TimeZoneInformation](#) [[▶ 321](#)]).

VAR_OUTPUT

```
VAR_OUTPUT
  out      : T_FILETIME;
  eTzID    : E_TimeZoneID := eTimeZoneID_Unknown;
  bB       : BOOL;
END_VAR
```

out: Konvertierte Lokalzeit (file time format, Typ: [T_FILETIME](#) [[▶ 323](#)]).

eTzID: Zusätzliche Sommer-/ Winterzeit-Information (Typ: [E_TimeZoneID](#) [[▶ 308](#)]).

bB: TRUE => B-Zeit (z. B.: **02:05:00 MEZ B**), FALSE => Übrige Zeit (z. B.: **02:05:00 MESZ A**). Bei einem Sprung der Lokalzeit zurück wird dieser Ausgang gesetzt und beim Überschreiten der doppelten Lokalzeit wieder zurückgesetzt.

Beispiel:

Die UTC-Zeit: DT#2011-09-02-09:01:31 wird in Lokalzeit konvertiert. Das Ergebnis ist: DT#2011-09-02-11:01:31.

```
PROGRAM MAIN
VAR
  in      : DT := DT#2011-09-02-09:01:31; (* UTC time *)
  out     : DT; (* Local time *)
  fbToLocal : FB_FileTimeToTzSpecificLocalTime;
END_VAR

fbToLocal( in := DT_TO_FILETIME( in ), tzInfo := WEST_EUROPE_TZI );
out := FILETIME_TO_DT( fbToLocal.out );
```

Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

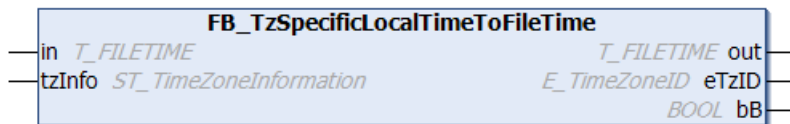
- [FB_TzSpecificLocalTimeToSystemTime](#) [[▶ 117](#)]

- [FB_TzSpecificLocalTimeToFileTime](#) [► 29]
- [FB_SystemTimeToTzSpecificLocalTime](#) [► 113]
- [FB_GetTimeZoneInformation](#) [► 78]
- [FB_SetTimeZoneInformation](#) [► 111]
- [NT_SetLocalTime](#) [► 124]
- [NT_GetTime](#) [► 122]
- [NT_SetTimeToRTCTime](#) [► 125]
- [F_TranslateFileTimeBias](#) [► 243]
- [FB_LocalSystemTime](#) [► 90]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.1.3 FB_TzSpecificLocalTimeToFileTime



Veraltete Funktion



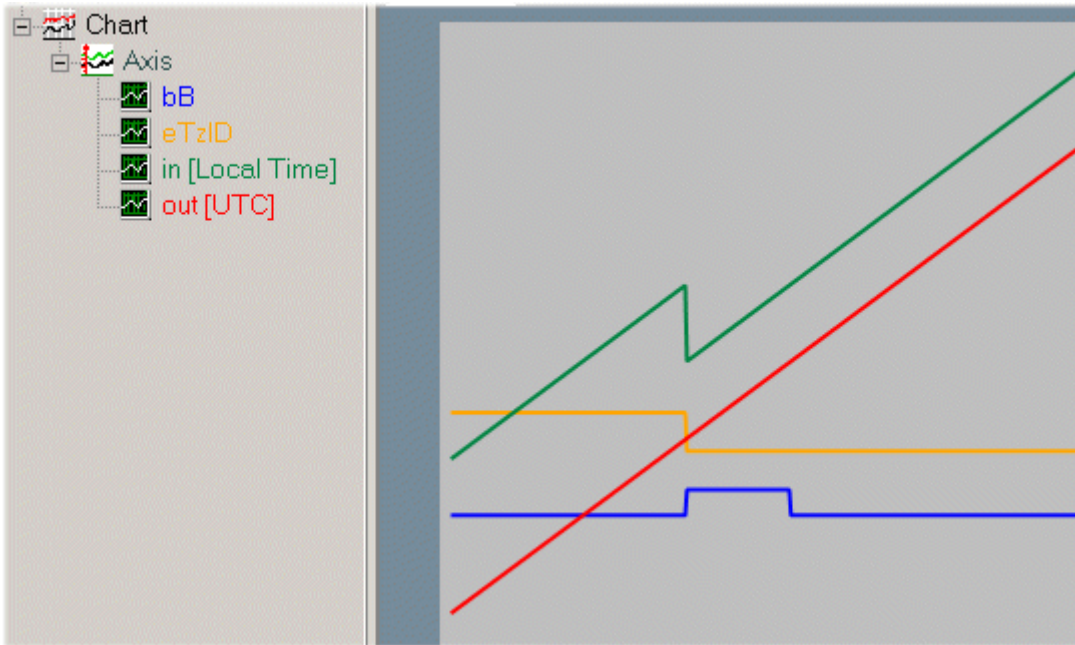
Dieser Funktionsbaustein ist veraltet. Verwenden Sie stattdessen den Funktionsbaustein [FB_TzSpecificLocalTimeToFileTime64](#) [► 115].

Der Funktionsbaustein konvertiert die Lokalzeit (file time format) in UTC-Zeit (file time format) unter der Berücksichtigung der angegebenen Zeitzoneinformation. Der Funktionsbaustein: [FB_TzSpecificLocalTimeToSystemTime](#) [► 117] besitzt eine ähnliche Funktionalität, mit dem Unterschied, dass er ein anderes Zeitformat (structured system time format) umrechnet.

Der Baustein eignet sich nur zur Konvertierung von **kontinuierlichen** Lokalzeit-Zeitstempelinformation. Die Lokalzeitsprünge durch die Sommer-/Winterzeit-Umstellung sind zulässig und werden vom Baustein erkannt. Lokalzeiten, die sich beliebig ändern führen zur fehlerhaften Konvertierung. Der Grund: Die zuletzt konvertierte Zeit wird im Baustein intern gespeichert, um die Sommerzeit-/Winterzeit-Information und die B-Zeiten (siehe unten) beim Zurückstellen der Lokalzeit erkennen zu können. Der Baustein besitzt eine Aktion: `A_Reset()`. Ein Aufruf dieser Aktion setzt die Ausgänge des Bausteins und die intern gespeicherte (zuletzt konvertierte Zeit) auf Null zurück.

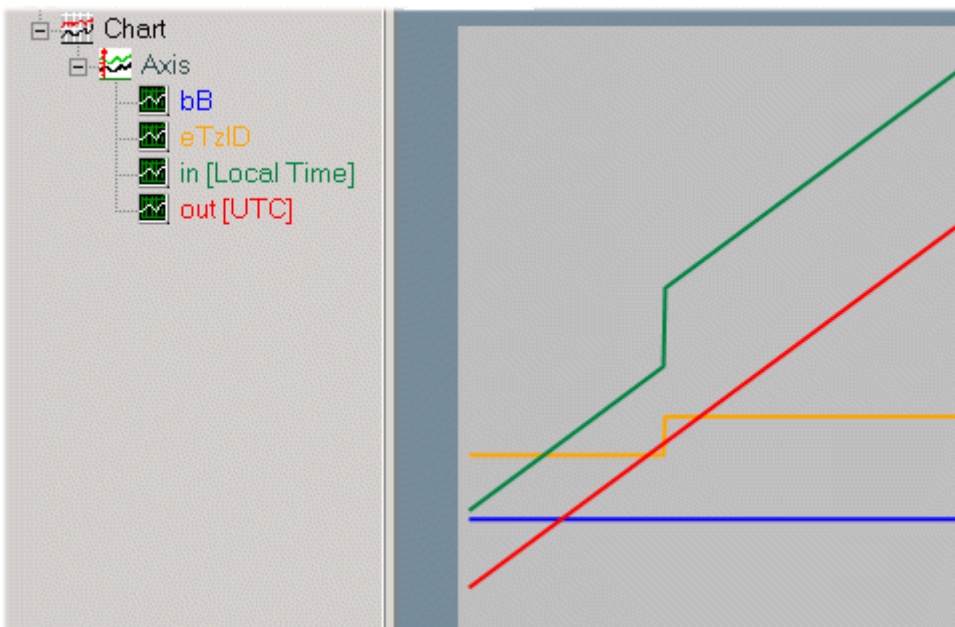
Die Sprünge in der Lokalzeit stellen hier ein Problem dar, da sie in eine lineare UTC-Zeit konvertiert werden müssen. Es empfiehlt sich daher, für Zeitstempelungsaufgaben die (kontinuierliche) UTC-Zeit zu verwenden, und diese erst zur optischen Anzeige der Werte (z. B. in einer Visualisierung) in die jeweilige Lokalzeit zu konvertieren.

1. Grafische Darstellung des zeitlichen Verhaltens beim Übergang von Sommerzeit zur Winterzeit (hier tzInfo = WEST_EUROPE_TZI):



Die Lokalzeit (grün) macht einen Sprung zurück. Die UTC-Ausgangszeit (rot) verläuft kontinuierlich weiter. Der Lokalzeit: **02h:59m:59s:999ms..** folgt unmittelbar die Zeit: **02h:00m:00s:000ms..** Die Zeiten zwischen 2h und 3h sind zweifach vorhanden. Die doppelte Zeit vor der Zeitumstellung wird z. B. als **02:05:00 MESZ A** bezeichnet und die Zeit nach der Umstellung als **02:05:00 MEZ B**. Die Ausgangsvariable *bB* gibt Information darüber ob es sich um den ersten oder zweiten *Durchgang* handelt. Beim zweiten *Durchgang* ist die *bB*-Ausgangsvariable (blau) auf TRUE gesetzt. Die *bB*-Ausgangsvariable wird automatisch zurückgesetzt nach dem die doppelte Zeit überschritten wurde. Die Zeitzonen-ID (orange) wechselt von *eTimeZoneID_Daylight* (Sommerzeit) zu *eTimeZoneID_Standard* (Winterzeit).

2. Grafische Darstellung des zeitlichen Verhaltens beim Übergang von Winterzeit zur Sommerzeit (hier tzInfo = WEST_EUROPE_TZI):



Die Lokalzeit (grün) macht einen Sprung nach vorne. Die UTC-Ausgangszeit (rot) verläuft kontinuierlich weiter. Der Lokalzeit: **2h:59m:59s:999ms..** folgt unmittelbar die Zeit: **3h:00m:00s:000ms..** Die Zeitzonen-ID (orange) wechselt von *eTimeZoneID_Standard* (Winterzeit) zu *eTimeZoneID_Daylight* (Sommerzeit).

VAR_INPUT

```
VAR_INPUT
  in      : T_FILETIME;
  tzInfo  : ST_TimeZoneInformation;
END_VAR
```

in: Lokalzeit (file time format) die konvertiert werden soll (Typ: [T_FILETIME \[▶ 323\]](#)).

tzInfo: Strukturvariable mit der aktuellen Zeitzone-Information des Betriebssystems (Typ: [ST_TimeZoneInformation \[▶ 321\]](#)).

VAR_OUTPUT

```
VAR_OUTPUT
  out      : T_FILETIME;
  eTzID    : E_TimeZoneID := eTimeZoneID_Unknown;
  bB       : BOOL;
END_VAR
```

out: Konvertierte UTC-Zeit (file time format) (Typ: [T_FILETIME \[▶ 323\]](#)).

eTzID: Zusätzliche Sommer-/ Winterzeit-Information (Typ: [E_TimeZoneID \[▶ 308\]](#)).

bB: TRUE => B-Zeit (Z.B.: **02:05:00 MEZ B**), FALSE => Übrige Zeit (Z.B.: **02:05:00 MESZ A**). Bei einem Sprung der Lokalzeit zurück wird dieser Ausgang gesetzt und beim Überschreiten der doppelten Lokalzeit wieder zurückgesetzt.

Beispiel:

Die Lokalzeit: DT#2011-09-02-11:01:31 wird in UTC-Zeit: DT#2011-09-02-09:01:31 konvertiert.

```
PROGRAM MAIN
VAR
  in      : DT := DT#2011-09-02-11:01:31; (* Local time *)
  out     : DT; (* UTC time *)
  fbToUTC : FB_TzSpecificLocalTimeToFileTime;
END_VAR

fbToUTC( in := DT_TO_FILETIME( in ), tzInfo := WEST_EUROPE_TZI );
out := FILETIME_TO_DT( fbToUTC.out );
```

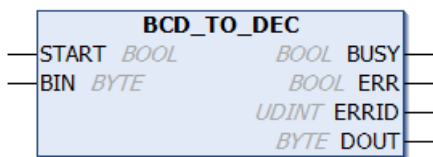
Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

- [FB_TzSpecificLocalTimeToSystemTime \[▶ 117\]](#)
- [FB_SystemTimeToTzSpecificLocalTime \[▶ 113\]](#)
- [FB_FileTimeTimeToTzSpecificLocalTime \[▶ 27\]](#)
- [FB_GetTimeZoneInformation \[▶ 78\]](#)
- [FB_SetTimeZoneInformation \[▶ 111\]](#)
- [NT_SetLocalTime \[▶ 124\]](#)
- [NT_GetTime \[▶ 122\]](#)
- [NT_SetTimeToRTCTime \[▶ 125\]](#)
- [F_TranslateFileTimeBias \[▶ 243\]](#)
- [FB_LocalSystemTime \[▶ 90\]](#)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.2 BCD_TO_DEC



Mit dem Funktionsbaustein "BCD_TO_DEC" können BCD-Zahlen in Dezimalformat konvertiert werden. Die zu konvertierende BCD-Zahl wird auf Zulässigkeit der Werte überprüft.

VAR_INPUT

```
VAR_INPUT
    START : BOOL;
    BIN   : BYTE;
END_VAR
```

START: Über eine positive Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

BIN: Die zu konvertierende BCD-Zahl.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY : BOOL;
    ERR  : BOOL;
    ERRID : UDINT;
    DOUT : BYTE;
END_VAR
```

BUSY: Beim Starten der Konvertierung wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Konvertierung abgeschlossen wurde. Nachdem der BUSY-Ausgang zurückgesetzt wurde, steht der Dezimalwert am DOUT-Ausgang zur Verfügung.

ERR: Beim Fehler wird die Variable auf TRUE gesetzt.

ERRID: Fehlercode.

DOUT: Beim Erfolg steht an diesem Ausgang die konvertierte Variable im Dezimalformat.

Fehlercodes:

Fehlercode	Fehlerbeschreibung
0	Kein Fehler
0x000F	Unzulässiger Wert im Low-Nibble der BCD-Zahl
0x00F0	Unzulässiger Wert im High-Nibble der BCD-Zahl

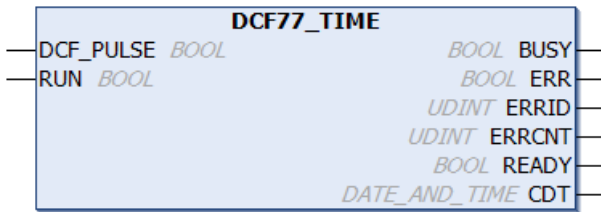
Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.3 DCF77_TIME



Dieser Funktionsbaustein wird durch den Funktionsbaustein [DCF77_TIME_EX](#) [▶ 36] ersetzt.



Mit dem Funktionsbaustein "DCF77_TIME" kann das DCF-77 Funkuhr-Signal dekodiert werden. Über eine steigende Flanke am RUN-Eingang wird die Dekodierung gestartet fortgesetzt, solange der RUN-Eingang gesetzt ist. Der Funktionsbaustein braucht im schlechtesten Fall max. 1 Minute um sich zu synchronisieren und eine weitere Minute um Daten für die nächste Minute zu dekodieren. Während dieser Zeit wird auf die fehlende 59. Sekundenmarke gewartet. Intern findet bei dem Funktionsbaustein eine Abtastung des DCF-77 Signals statt. Um die Flanken fehlerfrei abtasten zu können, sollte der Funktionsbaustein in jedem Zyklus der SPS einmal aufgerufen werden. Bei einer Zykluszeit <= 25ms können zufriedenstellende Ergebnisse erzielt werden. Bei einem fehlenden oder fehlerhaften DCF-77 Signal wird der ERR-Ausgang auf TRUE gesetzt und ein entsprechender Fehlercode am ERRID-Ausgang gesetzt. Bei dem nächsten fehlerfreien Empfang werden die ERR- und ERRID-Ausgänge zurückgesetzt. Manche der Empfänger liefern ein invertiertes DCF-77 Signal. In so einem Fall muss das Signal zuerst invertiert werden und dann an den DCF_PULSE-Eingang geleitet werden. Bei einem fehlerfreien Betrieb wird die aktuelle Zeit im Minutentakt am CDT-Ausgang aktualisiert. Dabei wird für einen Zyklus der SPS bei der Nullten Sekunde der READY-Ausgang auf TRUE gesetzt. In dieser Zeit ist die DCF-77 Zeit am CDT-Ausgang gültig und kann im SPS-Programm ausgewertet werden. Der READY-Ausgang wird nur dann gesetzt, wenn kein Fehler in den Daten für die kommende Minute erkannt wurde. Die Fehlererkennung wird mit Hilfe der übertragenen Paritätsbits durchgeführt. Bei schlechten Empfangsverhältnissen kann eine zu 100% fehlerfreie Erkennung nicht gewährleistet werden. D.h. bei zwei fehlerhaften (invertierten) Bits kann der Funktionsbaustein keinen Fehler erkennen und setzt den READY-Ausgang ebenfalls auf TRUE. Um eine zuverlässige Zeitinformation zu erhalten müssen zusätzliche Sicherungsmechanismen implementiert werden z.B. die Auswertung der Redundanz der Zeitinformation in aufeinanderfolgenden Minuten.

In dem DCF77_TIME-Funktionsbaustein ist eine einfache Plausibilitätsprüfung von zwei aufeinanderfolgenden Telegrammen implementiert. Diese Funktionalität kann über eine globale boolesche Variable für alle Instanzen des DCF77_TIME-Bausteins aktiviert werden. Bei der Aktivierung der Plausibilitätsprüfung verlängert sich die erste Synchronisierung um eine weitere Minute auf max. 3 Minuten.

```
GLOBAL_DCF77_SEQUENCE_CHECK : BOOL := FALSE;
(* TRUE = Enable plausibility check (two telegrams are checked), FALSE = Disable check *)
```

Die auftretenden Fehler während des Empfangs werden von dem Funktionsbaustein registriert. Der ERRCNT-Ausgang ist ein Fehlerzähler. Der Zähler gibt Auskunft über die Anzahl der aufgetretenen Fehler seit dem letzten fehlerfreien Empfang. Bei dem nächsten fehlerfreien Empfang wird dieser Zähler zurückgesetzt.

Zeitcode

Während jeder Minute werden die Nummern von Jahr, Monat, Tag, Wochentag, Stunde, Minute BCD-codiert durch Impulsmodulation der Sekundenmarken übertragen. Die übertragenen Informationen gelten jeweils für die nachfolgende Minute. Jede Sekunde wird eine der Sekundenmarken übertragen. Dabei entspricht eine Sekundenmarke mit der Dauer von 0.1s einer binären Null und eine mit der Dauer von 0.2s einer binären Eins. Die Informationen werden mit 3 Prüfbits ergänzt. Bei der 59. Sekunde fehlt die Sekundenmarke und diese "Lücke" kann zur Synchronisation des Empfängers benutzt werden.

Über eine globale Variable kann die Länge des kurzen und langen Impulssignals konfiguriert werden. Bei einem gestörten Signal sind die Impulsbreiten immer kleiner. In der Empfängerspezifikation sind meistens Angaben über minimale und maximale Impulsbreiten der beiden logischen Signale zu finden, wobei breitere Impulse bei höheren Feldstärken zu erwarten sind und schmale Impulse bei niedrigen Feldstärken oder entsprechenden Störverhältnissen. In der Nähe des Senders (wo die Feldstärke sehr groß ist) können ebenfalls Probleme entstehen wenn die Impulsbreite der logischen Null zu groß wird. Deshalb wird abhängig von der Spezifikation des Empfängers eine feste Grenze zur Unterscheidung zwischen Null und Eins festgelegt. **Prüfen Sie deshalb die Spezifikation des eingesetzten Empfängers und konfigurieren entsprechend die Impulslänge.**

```
GLOBAL_DCF77_PULSE_SPLIT : TIME := T#140ms; (* 0 == pulse < 140ms, 1 == pulse > 140 *)
```

Z.B.: in der Spezifikation von Atmel T4227 (Time Code Receiver) wird folgende Pulslänge angegeben:
 100ms Pulse (Null): Min: 70ms, Typical: 95ms, Max: **130ms**
 200ms Pulse (Eins): Min **170ms**, Typical 195ms, Max 235ms
 Für diesen IC wäre ein Grenzwert von 150ms optimal = $130 + ((170ms - 130ms) / 2)$.

Tipp:

Wenn der konfigurierte Grenzwert für die Impulslänge einen zu kleinen Wert hat dann werden kurze Impulse als lange erkannt. Umgekehrt gilt, wenn der konfigurierte Grenzwert zu groß ist dann werden lange Impulse als kurze erkannt. Der Empfänger kann bei einer passenden Checksumme diese Fehler nicht erkennen. Im ersten Fall liefert der Empfänger möglicherweise Zeiten die weit in der Zukunft liegen und im zweiten Fall Zeiten aus der Vergangenheit.

VAR_INPUT

```
VAR_INPUT
    DCF_PULSE : BOOL;
    RUN       : BOOL;
END_VAR
```

DCF_PULSE: Das DCF-77 Signal.

RUN: Eine steigende Flanke an diesem Eingang initialisiert den Funktionsbaustein und startet die Dekodierung des DCF-77 Signals. Wird dieser Eingang zurückgesetzt, dann wird die Dekodierung gestoppt.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY   : BOOL;
    ERR    : BOOL;
    ERRID  : UDINT;
    ERRCNT: UDINT;
    READY  : BOOL;
    CDT    : DATE_AND_TIME;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt.

ERR: Ist ein Fehler bei der Dekodierung aufgetreten, dann wird dieser Ausgang gesetzt.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die Fehlernummer.

ERRCNT: Anzahl der aufgetretenen Fehler seit dem letzten fehlerfreien Empfang.

READY: Ist dieser Ausgang gesetzt, dann sind die Daten am CDT-Ausgang gültig.

CDT: Die DCF-77 Zeit im DATE_AND_TIME-Format.

Fehlercodes	Fehlerbeschreibung
0	Kein Fehler
0x100	Timeout-Fehler. Möglicherweise kein DCF-77 Signal erkannt.
0x200	Parity-Fehler. In den empfangenen Daten wurden fehlerhafte Bits erkannt.
0x300	Fehlerhafte Daten empfangen. Da bei der Parity Prüfung nur ein falsches Bit erkannt werden kann, werden die empfangenen Daten noch auf Gültigkeit überprüft (bei Monat = 13 wird z.B. dieser Fehlercode gesetzt).
0x400	Der letzte Dekodierungszyklus war zu lang. Dieser Fehler kann bei einem schlechten Empfang auftreten (zu wenig Sekundenmarken wurden empfangen).
0x500	Der letzte Dekodierungszyklus war zu kurz. Dieser Fehler kann bei einem schlechten Empfang auftreten (zusätzliche Flanken wurden empfangen).

Beispiel:

In der Beispielanwendung wird bei einem fehlerfreien Empfang eine TwinCAT Software-Uhr (RTC) mit der Funkzeit synchronisiert.

```
PROGRAM P_DCF77_TIME
VAR
  bDcfPulse      : BOOL;
  fbDcf          : DCF77_TIME;
  bBusy         : BOOL;
  bError        : BOOL;
  nErrID        : UDINT;
  nErrCnt       : UDINT;
  bDcfValid     : BOOL;
  tDcfDT        : DT;
  fbRtc         : RTC;
  bRtcValid     : BOOL;
  tRtcDT        : DT;
END_VAR
```

Online Ansicht:

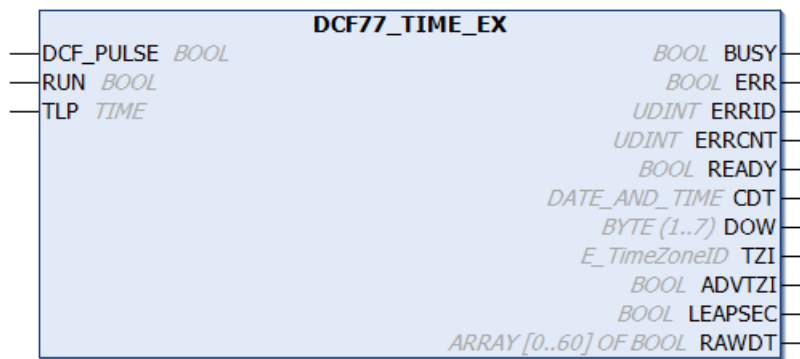
Expression	Type	Value	Prepared value	Address	Comment
bDcfPulse	BOOL	TRUE			
fbDcf	DCF77_TIME				
bBusy	BOOL	TRUE			
bError	BOOL	FALSE			
nErrID	UDINT	16#00000000			
nErrCnt	UDINT	16#00000000			
bDcfValid	BOOL	FALSE			
tDcfDT	DATE_AND_TIME	DT#2014-9-26-14:3:0			
fbRtc	RTC				
bRtcValid	BOOL	TRUE			
tRtcDT	DATE_AND_TIME	DT#2014-9-26-14:3:11			

Siehe auch in der Beschreibung des [DCF77_TIME_EX \[▶ 36\]](#)-Funktionsbausteins.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.4 DCF77_TIME_EX



Mit dem Funktionsbaustein "DCF77_TIME_EX" kann das DCF-77 Funkuhr-Signal dekodiert werden. Im Gegensatz zu dem "DCF77 TIME [▶ 32]"-Funktionsbaustein werden bei diesem Baustein standardmäßig zwei aufeinanderfolgende Telegramme auf Plausibilität geprüft.

Über eine steigende Flanke am RUN-Eingang wird die Dekodierung gestartet und fortgesetzt, solange der RUN-Eingang gesetzt ist. Der Funktionsbaustein braucht im ungünstigsten Fall maximal eine Minute um sich zu synchronisieren, und zwei weitere Minuten um Daten für die nächste Minute zu dekodieren. Während dieser Zeit wird auf die fehlende 59. Sekundenmarke gewartet. Intern findet bei dem Funktionsbaustein eine Abtastung des DCF-77 Signals statt. Um die Flanken fehlerfrei abtasten zu können, sollte der Funktionsbaustein in jedem Zyklus der SPS einmal aufgerufen werden. Bei einer Zykluszeit $\leq 25\text{ms}$ können zufriedenstellende Ergebnisse erzielt werden. Bei einem fehlenden oder fehlerhaften DCF-77 Signal wird der ERR-Ausgang auf TRUE gesetzt und ein entsprechender Fehlercode am ERRID-Ausgang gesetzt. Bei dem nächsten fehlerfreien Empfang werden die ERR- und ERRID-Ausgänge zurückgesetzt. Manche der Empfänger liefern ein invertiertes DCF-77 Signal. In so einem Fall muss das Signal zuerst invertiert werden und dann an den DCF_PULSE-Eingang geleitet werden. Bei einem fehlerfreien Betrieb wird die aktuelle Zeit im Minutentakt am CDT-Ausgang aktualisiert. Dabei wird für einen Zyklus der SPS bei der Nullten Sekunde der READY-Ausgang auf TRUE gesetzt. In dieser Zeit ist die DCF-77 Zeit am CDT-Ausgang gültig und kann im SPS-Programm ausgewertet werden. Der READY-Ausgang wird nur dann gesetzt, wenn die Daten für die kommende Minute fehlerfrei empfangen werden konnten. Die Fehlererkennung wird mit Hilfe der übertragenen Paritätsbits durchgeführt und die zwei letzten Telegramme auf Plausibilität überprüft. Bei schlechten Empfangsverhältnissen kann eine zu 100% fehlerfreie Erkennung nicht gewährleistet werden. D.h. bei zwei fehlerhaften (invertierten) Bits in zwei danach folgenden Telegrammen kann der Funktionsbaustein keinen Fehler erkennen und setzt den READY-Ausgang ebenfalls auf TRUE. Durch die Plausibilitätsprüfung ist die Wahrscheinlichkeit, dass die passenden Bits verfälscht werden und ein solcher Fehler nicht erkannt wird recht gering.

Die auftretenden Fehler während des Empfangs werden von dem Funktionsbaustein registriert. Der ERRCNT-Ausgang ist ein Fehlerzähler. Der Zähler gibt Auskunft über die Anzahl der aufgetretenen Fehler seit dem letzten fehlerfreien Empfang. Bei dem nächsten fehlerfreien Empfang wird dieser Zähler zurückgesetzt.

Zeitcode

Während jeder Minute werden die Nummern von Jahr, Monat, Tag, Wochentag, Stunde, Minute BCD-codiert durch Impulsmodulation der Sekundenmarken übertragen. Die übertragenen Informationen gelten jeweils für die nachfolgende Minute. Jede Sekunde wird eine der Sekundenmarken übertragen. Dabei entspricht eine Sekundenmarke mit der Dauer von 0.1s einer binären Null und eine mit der Dauer von 0.2s einer binären Eins. Die Informationen werden mit 3 Prüfbits ergänzt. Bei der 59. Sekunde fehlt die Sekundenmarke und diese "Lücke" kann zur Synchronisation des Empfängers benutzt werden.

VAR_INPUT

```
VAR_INPUT
  DCF_PULSE  : BOOL;
  RUN        : BOOL;
  TLP        : TIME := 140ms;
END_VAR
```

DCF_PULSE: Das DCF-77 Signal.

RUN: Eine steigende Flanke an diesem Eingang initialisiert den Funktionsbaustein und startet die Dekodierung des DCF-77 Signals. Wird dieser Eingang zurückgesetzt, dann wird die Dekodierung gestoppt.

TLP: Über diesen Eingang wird abhängig von der Spezifikation des Empfängers eine feste Grenze zur Unterscheidung zwischen Null und Eins festgelegt. Bei einem gestörten Signal sind die Impulsbreiten immer kleiner. In den Empfängerspezifikationen finden sich meistens Angaben über minimale und maximale Impulsbreiten der beiden logischen Signale, wobei breitere Impulse bei höheren Feldstärken zu erwarten sind und schmale Impulse bei niedrigen Feldstärken oder entsprechenden Störverhältnissen. In der Nähe des Senders (wo die Feldstärke sehr groß ist) können ebenfalls Probleme entstehen, wenn die Impulsbreite der logischen Null zu groß wird. **Prüfen Sie deshalb die Spezifikation des eingesetzten Empfängers und konfigurieren entsprechend die Impulslänge.**

Z.B.: in der Spezifikation von Atmel T4227 (Time Code Receiver) wird folgende Pulslänge angegeben:
 100ms Pulse (Null): Min: 70ms, Typical: 95ms, Max: **130ms**
 200ms Pulse (Eins): Min **170ms**, Typical 195ms, Max 235ms
 Für diesen IC wäre ein Grenzwert von 150ms optimal = $130 + ((170ms - 130ms) / 2)$.



Wenn der konfigurierte Grenzwert für die Impulslänge einen zu kleinen Wert hat, dann werden kurze Impulse als lange erkannt. Umgekehrt gilt, wenn der konfigurierte Grenzwert zu groß ist, dann werden lange Impulse als kurze erkannt. Der Empfänger kann bei einer passenden Checksumme diese Fehler nicht erkennen. Im ersten Fall liefert der Empfänger möglicherweise Zeiten die weit in der Zukunft liegen und im zweiten Fall Zeiten aus der Vergangenheit.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY      : BOOL;
  ERR       : BOOL;
  ERRID     : UDINT;
  ERRCNT    : UDINT;
  READY     : BOOL;
  CDT       : DATE_AND_TIME;
  DOW       : BYTE(1..7); (* ISO 8601 day of week: 1 = Monday.. 7 = Sunday *)
  TZI       : E_TimeZoneID; (* time zone information *)
  ADVTZI    : BOOL; (* MEZ->MESZ or MESZ->MEZ time change notification *)
  LEAPSEC   : BOOL; (* TRUE = Leap second *)
  RAWDT     : ARRAY[0..60] OF BOOL; (* Raw decoded data bits *)
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt.

ERR: Ist ein Fehler bei der Dekodierung aufgetreten, dann wird dieser Ausgang gesetzt.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die Fehlernummer.

ERRCNT: Anzahl der aufgetretenen Fehler seit dem letzten fehlerfreien Empfang.

READY: Ist dieser Ausgang gesetzt, dann sind die Daten am CDT-Ausgang gültig.

CDT: Die DCF-77 Zeit im DATE_AND_TIME-Format.

DOW: Wochentag nach ISO 8601: 1 = Montag... 7 = Sonntag.

TZI: Zeitzoneinformation (Sommer-/Winterzeit).

ADVTZI: Ankündigung eines MEZ -> MESZ oder MESZ -> MEZ Wechsels. Am Ende dieser Stunde wird MESZ/MEZ umgestellt (siehe Telegrammbeispiele).

LEAPSEC: Ankündigung einer Schaltsekunde. Am Ende dieser Stunde wird eine Schaltsekunde eingefügt (siehe Telegrammbeispiele).

RAWDT: Zuletzt decodierte (rohe) Bit-Information. Bitte beachten Sie, dass nur die Paritätsbits der Zeitinformation überprüft werden. Die Paritätsbits der Wetterdaten werden nicht ausgewertet!

Fehlercodes	Fehlerbeschreibung
0	Kein Fehler
0x100	Timeout-Fehler. Möglicherweise kein DCF-77 Signal erkannt.

Fehlercodes	Fehlerbeschreibung
0x200	Parity-Fehler. In den empfangenen Daten wurden fehlerhafte Bits erkannt.
0x300	Fehlerhafte Daten wurden empfangen. Da bei der Parity Prüfung nur ein falsches Bit erkannt werden kann, werden die empfangenen Daten noch auf Gültigkeit überprüft (bei Monat = 13 wird z.B. dieser Fehlercode gesetzt).
0x400	Der letzte Dekodierungszyklus war zu lang. Dieser Fehler kann bei einem schlechten Empfang auftreten (zu wenig Sekundenmarken wurden empfangen).
0x500	Der letzte Dekodierungszyklus war zu kurz. Dieser Fehler kann bei einem schlechten Empfang auftreten (zusätzliche Flanken wurden empfangen).

Telegrammbeispiele:

CEST -> CET (daylight-saving time -> standard time)

```
'0 01110100100111 011001 00011011 0100001 011001 111 00001 000100000': Sunday, 26.10.08 02:58:00, TZI = eTimeZoneID_Daylight, ADVTZI = TRUE
'0 11110001101110 011001 10011010 0100001 011001 111 00001 000100000': Sunday, 26.10.08 02:59:00, TZI = eTimeZoneID_Daylight, ADVTZI = TRUE
'0 01000001001110 010101 00000000 0100001 011001 111 00001 000100000': Sunday, 26.10.08 02:00:00, TZI = eTimeZoneID_Standard, ADVTZI = TRUE
'0 01111110100000 000101 10000001 0100001 011001 111 00001 000100000': Sunday, 26.10.08 02:01:00, TZI = eTimeZoneID_Standard, ADVTZI = FALSE
```

CET -> CEST (standard time -> daylight-saving time)

```
'0 01000110111010 010101 00011011 1000001 000011 111 11000 000100000': Sunday, 30.03.08 01:58:00, TZI = eTimeZoneID_Standard, ADVTZI = TRUE
'0 01000010100111 010101 10011010 1000001 000011 111 11000 000100000': Sunday, 30.03.08 01:59:00, TZI = eTimeZoneID_Standard, ADVTZI = TRUE
'0 10000111100011 011001 00000000 1100000 000011 111 11000 000100000': Sunday, 30.03.08 03:00:00, TZI = eTimeZoneID_Daylight, ADVTZI = TRUE
'0 01010000010110 001001 10000001 1100000 000011 111 11000 000100000': Sunday, 30.03.08 03:01:00, TZI = eTimeZoneID_Daylight, ADVTZI = FALSE
```

Leap second

```
'0 10110000100001 000111 10011010 0000000 100000 001 10000 100100001': Thursday, 01.01.09 00:59:00, TZI = eTimeZoneID_Standard, LEAPSEC = TRUE
'0 11010010111000 000111 00000000 1000001 100000 001 10000 100100001': Thursday, 01.01.09 01:00:00, TZI = eTimeZoneID_Standard, LEAPSEC = TRUE
'0 01000110011101 000101 10000001 1000001 100000 001 10000 100100001': Thursday, 01.01.09 01:01:00, TZI = eTimeZoneID_Standard, LEAPSEC = FALSE
```

- █: LEAPSEC bit;
- █: CET/CEST-Information;
- █: ADVTZI bit;

Beispiel:

In der Beispielanwendung wird bei einem fehlerfreien Empfang eine TwinCAT Software-Uhr (RTC) mit der Funkzeit synchronisiert.

```
PROGRAM MAIN
VAR
    bDcfPulse AT%I* : BOOL;
    fbDcf          : DCF77_TIME_EX;
    bBusy         : BOOL;
    bError        : BOOL;
    nErrID        : UDINT;
    nErrCnt       : UDINT;
    bDcfValid     : BOOL;
    tDcfDt        : DT;
    nDow          : BYTE(1..7);
    eTzi          : E_TimeZoneID; (* time zone information *)
    bAdvTzi       : BOOL; (* MEZ->MESZ or MESZ->MEZ time change notification *)
    bLeapSec      : BOOL; (* TRUE = Leap second *)
    arRawDt       : ARRAY[0..60] OF BOOL;
    fbRtc         : RTC;
    bRtcValid     : BOOL;
    tRtcDt        : DT;
END_VAR

fbDcf( DCF_PULSE:= NOT bDcfPulse, RUN:= TRUE, TLP:= T#140MS,
      BUSY=>bBusy, ERR=>bError, ERRID=>nErrID, ERRCNT=>nErrCnt,
      READY=>bDcfValid, CDT=>tDcfDt, DOW=>nDow, TZI=>eTzi,
      ADVTZI=>bAdvTzi, LEAPSEC=>bLeapSec, RAWDT=>arRawDt );

fbRtc( EN := bDcfValid, PDT := tDcfDt, Q=>bRtcValid, CDT=>tRtcDt );
```

Online Ansicht:

MAIN [Online] x P_DCF77_TIME [Online] Source Control Explorer

TwinCAT_Device.Untitled1.MAIN

Expression	Type	Value	Prepared value	Address	Comment
bDcfPulse	BOOL	TRUE		%I*	
fbDcf	DCF77_TIME_EX				
bBusy	BOOL	TRUE			
bError	BOOL	FALSE			
nErrID	UDINT	16#00000000			
nErrCnt	UDINT	16#00000000			
bDcfValid	BOOL	FALSE			
tDcfDt	DATE_AND_TIME	DT#2014-9-26-14:4:0			
nDow	BYTE (BYTE#1..7)	16#05			
eTzi	E_TIMEZONEID	eTimeZoneID_Daylight			time zone information
bAdvTzi	BOOL	FALSE			MEZ->MESZ or MESZ->time change notifi...
bLeapSec	BOOL	FALSE			TRUE = Leap second
arRawDt	ARRAY [0..60] OF B...				
fbRtc	RTC				
bRtcValid	BOOL	TRUE			
tRtcDt	DATE_AND_TIME	DT#2014-9-26-14:4:4			

```

1  fbDcf( DCF_PULSE FALSE != NOT bDcfPulse TRUE, RON TRUE := TRUE, ILF T#140ms, := T#140MS,
2  BUSY TRUE =>bBusy TRUE, ERR FALSE =>bError FALSE, ERRID 16#00000000 =>nErrID 16#00000000, ERRCNT 16#00000000 =>nErrCnt 16#00000000,
3  READY FALSE =>bDcfValid FALSE, CDI DT#2014-9-26-14:4:0 =>tDcfDt DT#2014-9-26-14:4:0, DOW 16#05 =>nDow 16#05, TZI[eTimeZone] =>eTzi[eTimeZone],
4  ADVTZI FALSE =>bAdvTzi FALSE, LEAPSEC FALSE =>bLeapSec FALSE, RAWDT =>arRawDt );
5
6  fbRtc( EN FALSE := bDcfValid FALSE, PDI DT#2014-9-26-14:4:0 := tDcfDt DT#2014-9-26-14:4:0, TRUE =>bRtcValid TRUE, CDI DT#2014-9-26-14:4:4 =>tRtcDt DT#2014-9-26-14:4:4 );
7

```

Siehe auch in der Beschreibung des DCF77_TIME [▶ 32]-Funktionsbausteins.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.5 DEC_TO_BCD



Mit dem Funktionsbaustein "DEC_TO_BCD" können Dezimalzahlen in BCD-Format konvertiert werden. Die zu konvertierende Zahl wird auf Zulässigkeit der Werte überprüft.

VAR_INPUT

```

VAR_INPUT
    START : BOOL;
    DIN : BYTE;
END_VAR

```

START: Über eine positive Flanke an diesem Eingang wird der Funktionsbaustein aktiviert.

DIN: Die zu konvertierende Dezimalzahl.

VAR_OUTPUT

```

VAR_OUTPUT
    BUSY : BOOL;
    ERR : BOOL;
    ERRID : UDINT;
    BOUT : BYTE;
END_VAR

```

BUSY: Beim Starten der Konvertierung wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Konvertierung abgeschlossen wurde. Nachdem der BUSY-Ausgang zurückgesetzt wurde, steht die BCD-Zahl am BOUT-Ausgang zur Verfügung.

ERR: Beim Fehler wird die Variable auf TRUE gesetzt.

ERRID: Fehlercode.

BOUT: Beim Erfolg steht an diesem Ausgang die konvertierte Variable im BCD-Format.

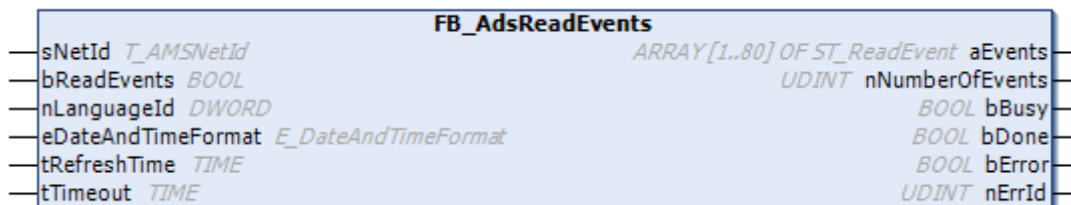
Fehlercodes:

Fehlercode	Fehlerbeschreibung
0	Kein Fehler
0x00FF	Unzulässiger Dezimalwert der zu konvertierenden Variablen

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.6 FB_AdsReadEvents



Der Funktionsbaustein erfragt die aktiven Meldungen des EventLoggers via ADS und stellt sie in Form eines Arrays `aEvents` zur Verfügung. Wenn die Meldungen im Visualisierungselement Event-Tabelle angezeigt werden sollen, muss das Array `aEvents` in dessen Eigenschaft `Meldungsdatenarray` eingetragen werden.

Meldungen mit einer Textlänge kleiner gleich 255 Zeichen können am Ausgang vollständig ausgegeben werden. Meldungen mit einer Textlänge größer 255 Zeichen und kleiner gleich 1023 Zeichen werden mit abgeschnittenem Text ausgegeben. Meldungen mit einer Textlänge größer 1023 Zeichen können nicht ausgegeben werden und der Funktionsbaustein liefert einen Fehler.

VAR_INPUT

```
VAR_INPUT
  sNetId      : T_AMSNetId;
  bReadEvents : BOOL;
  nLanguageId : DWORD;
  eDateAndTimeFormat : E_DateAndTimeFormat;
  tRefreshTime : TIME;
  tTimeout     : TIME;
END_VAR
```

sNetId: AmsNetId des Geräts, von dem die Meldungen des EventLoggers abgefragt werden sollen. Falls die Meldungen lokal ausgelesen werden sollen, kann ein Leerstring angegeben werden.

bReadEvents: Mit dem Eingang kann die Freigabe zum Auslesen der Meldungen gegeben werden. Mit Rücknahme der Freigabe werden auch die Fehlerausgänge (`bError` und `nErrId`) zurückgesetzt.

nLanguageId: (Sprach-ID) Definiert, welche Übersetzung des Meldungstextes abgefragt werden soll.

eDateAndTimeFormat: Definiert, welches Format die Zeitstempel haben sollen. Zur Auswahl stehen:

- `de_De` – deutsche Schreibweise: `dd.MM.yyyy hh:mm:ss` (24 h)
- `en_GB` – britische Schreibweise: `dd/MM/yyyy hh:mm:ss` (12 h)
- `en_US` – amerikanische Schreibweise: `MM/dd/yyyy hh:mm:ss` (12 h)

tRefreshTime: Definiert die Zeitspanne, nach der die Abfrage der Meldungen wiederholt wird.

tTimeout: Definiert die Zeitspanne, nach der ein Zeitüberschreitungsfehler ausgelöst wird.

VAR_OUTPUT

```
VAR_OUTPUT
  aEvents      : ARRAY[1..80] OF ST_ReadEvent;
  nNumberOfEvents : UDINT;
  bBusy        : BOOL;
  bDone        : BOOL;
  bError       : BOOL;
  nErrorId     : UDINT;
END_VAR
```

aEvents: Über dieses Array stellt der Funktionsbaustein die ausgelesenen Meldungen zur Verfügung. Maximal können in dem Array 80 Meldungen gespeichert werden. (siehe [ST_ReadEvent](#) [▶ 318])

nNumberOfEvents: Gibt an, wie viele Meldungen aktuell im Array `aEvents` gespeichert sind.

bBusy: Gibt an, ob der Baustein gerade arbeitet.

bDone: TRUE, wenn der Baustein gerade nicht arbeitet, aber mindestens einmal gearbeitet hat.

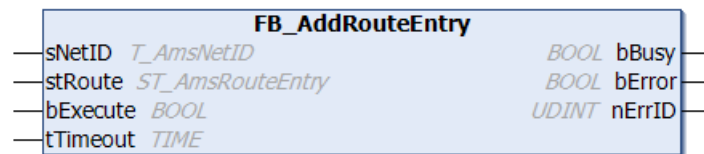
bError: Gibt an, ob ein Fehler aufgetreten ist.

nErrorId: Gibt die Fehlernummer an.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.7 FB_AddRouteEntry



Mit dem Funktionsbaustein kann eine neue AMS Router Verbindung (Remote Route) zu einem TwinCAT System hinzugefügt werden.

AMS Router Verbindungsliste

Beide Kommunikationspartner einer AMS Router Verbindung besitzen eine AMS Router Verbindungsliste. Diese Listen enthalten AMS Router Verbindungen. Die Funktionsfähigkeit einer AMS Router Verbindung ist gegeben, wenn beide Kommunikationspartner sich gegenseitig in ihrer Verbindungsliste eingetragen haben.

Bei Verwendung des Funktionsbaustein wird nur bei einem Kommunikationspartner dessen Liste erweitert.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetID;
  stRoute     : ST_AmsRouteEntry;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden auf dem die AMS Router Verbindungsliste um eine neue Verbindung ergänzt werden soll (Typ: `T_AmsNetID`). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

stRoute: Strukturelement mit Parametern der neuen Verbindung (Typ: [ST_AmsRouteEntry](#) [▶ 311]).

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy : BOOL;
  bError : BOOL;
  nErrId : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

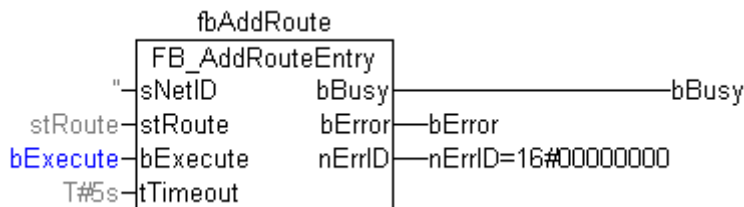
nErrId: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [► 360].

Beispiel:

Auf dem lokalen TwinCAT System soll eine neue AMS Router Verbindung mit dem Verbindungsnamen: "TEST", TwinCAT Netzwerkadresse: "172.16.6.111.1", IP-Adresse: "172.16.6.111" und dem Transportweg: "TCP/IP" hinzugefügt werden.

```
PROGRAM P_TEST3
VAR
  fbAddRoute : FB_AddRouteEntry;
  bExecute : BOOL;
  bBusy : BOOL;
  bError : BOOL;
  nErrID : UDINT;
  stRoute : ST_AmsRouteEntry := ( sName := 'TEST',
                                  sNetID := '172.16.6.111.1.1',
                                  sAddress := '172.16.6.111',
                                  eTransport := eRouteTransport_TCP_IP );
END_VAR
```

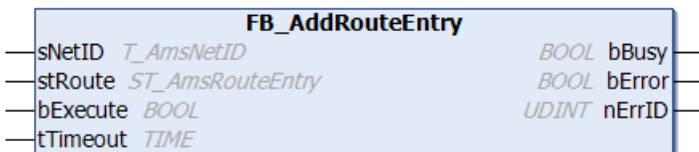
Die benötigten Verbindungsparameter werden bereits im Deklarationsteil initialisiert. Bei einer steigenden Flanke an der *bExecute*-Variablen wird die neue Verbindung hinzugefügt.



Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.8 FB_AddRouteEntryEx



Mit dem Funktionsbaustein kann eine neue AMS Router Verbindung (Remote Route) zu einem TwinCAT System hinzugefügt werden.

Der Funktionsbaustein unterstützt mit der virtuellen AmsNetId die AmsNAT- Funktionalität. Diese ermöglicht das Herstellen von Routen zu zwei oder mehreren Steuerungen, welche die gleiche AmsNetId besitzen.

AMS Router Verbindungsliste

I Beide Kommunikationspartner einer AMS Router Verbindung besitzen eine AMS Router Verbindungsliste. Diese Listen enthalten AMS Router Verbindungen. Die Funktionsfähigkeit einer AMS Router Verbindung ist gegeben, wenn beide Kommunikationspartner sich gegenseitig in ihrer Verbindungsliste eingetragen haben.
Bei Verwendung des Funktionsbausteins wird nur bei einem Kommunikationspartner dessen Liste erweitert.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetID;
  stRoute     : ST_AmsRouteEntryEx;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden auf dem die AMS Router Verbindungsliste um eine neue Verbindung ergänzt werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

stRoute: Strukturelement mit Parametern der neuen Verbindung (Typ: [ST_AmsRouteEntryEx](#) [[▶ 312](#)]).

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrId      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

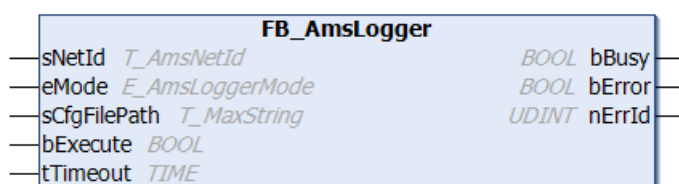
bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten *bError*-Ausgang die [ADS-Fehlernummer](#) [[▶ 360](#)].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= v3.3.41.0

3.9 FB_AmsLogger



Der "TwinCAT AMS Logger" gehört zu den Komponenten des "TwinCAT ADS Monitors" (..\TwinCAT\AdsMonitor\Logger\TcAmsLog.exe). Der Logger zeichnet die AMS/ADS-Kommandos auf den Datenträger auf. Die Aufzeichnung kann später z.B. bei einer Fehlersuche mit dem "TwinCAT AMS ADS Viewer" angezeigt und analysiert werden.

Mit dem FB_AmsLogger-Funktionsbaustein kann die Aufzeichnung aus einem SPS-Programm gestartet bzw. gestoppt werden. Der FB_AmsLogger-Funktionsbaustein kann nur mit einer bereits existierenden/laufenden Instanz der TcAmsLog.exe kommunizieren. D.h. die TcAmsLog.exe muss vorher (z.B. manuell aus dem Start-Menü) oder mit Hilfe des [NT_StartProcess \[▶ 128\]](#)-Bausteins gestartet worden sein.

VAR_INPUT

```
VAR_INPUT
  sNetId      : T_AmsNetId := '';
  eMode       : E_AmsLoggerMode := AMSLOGGER_RUN;
  sCfgFilePath : T_MaxString := '';
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem der Status des "TwinCAT AMS Loggers" geändert werden soll (Typ: T_AmsNetID). Für den Logger auf dem lokalen Rechner kann auch ein Leerstring angegeben werden.

eMode: Der neue Status in den der "TwinCAT AMS Logger" versetzt werden soll (Typ: E_AmsLoggerMode [\[▶ 300\]](#), Aufzeichnung starten/stoppen).

sCfgFilePath: (Optional) Pfad für eine "TwinCAT AMS Logger"-Konfigurationsdatei (Typ: T_MaxString). Zur Zeit noch nicht implementiert und reserviert für zukünftige Anwendungen (geben Sie bitte ein Leerstring an).

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrId     : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten bError-Ausgang die [ADS-Fehlernummer \[▶ 360\]](#).

Beispiel:

Beim SPS-Programmstart wird auf einem lokalen System eine Instanz von TcAmsLog.exe gestartet. Beim Setzen der bRecord-Variablen auf TRUE wird die Aufzeichnung AMS/ADS-Kommandos gestartet und beim zurücksetzen auf FALSE gestoppt.

Deklarationsteil:

```
PROGRAM MAIN
VAR
  bRecord      : BOOL := TRUE; (* TRUE => start recording, FALSE => stop recording *)
  fbStartProcess : NT_StartProcess := ( NETID := '', PATHSTR := 'c:
\TwinCAT\AdsMonitor\Logger\TcAmsLog.exe',
                                     DIRNAME := 'c:
\TwinCAT\AdsMonitor\Logger', COMNDLINE := '', TMOUT := DEFAULT_ADS_TIMEOUT );
  fbAmsLogger   : FB_AmsLogger := ( sNetID := '', eMode := AMSLOGGER_STOP, sCfgFilePath := '', tT
imeout := DEFAULT_ADS_TIMEOUT );
  state        : BYTE;
  bBusy        : BOOL;
  bError       : BOOL;
  nErrID       : UDINT;
  eCurrMode    : E_AmsLoggerMode := AMSLOGGER_STOP; (* Current mode/state *)
```

```
eNewMode      : E_AmsLoggerMode := AMSLOGGER_STOP; (* New mode/state *)
timer         : TON := ( PT := T#5s );
END_VAR
```

Implementierung:

```
CASE state OF
0: (* Start instance of TcAmsLogger.exe *)
  fbStartProcess( START := FALSE );
  fbStartProcess( START:= TRUE );
  state := 1;

1: (* Wait until command execution started *)
  fbStartProcess( START := FALSE, BUSY=>bBusy, ERR=>bError, ERRID=>nErrID );
  IF NOT bBusy THEN
    IF NOT bError THEN (* Success *)
      state := 2;
    ELSE (* Error *)
      state := 100;
    END_IF
  END_IF

2: (*Wait until instance started or new AMS logger mode/state set *)
  timer( IN := TRUE );
  IF timer.Q THEN
    timer( IN := FALSE );
    state := 3;
  END_IF

3: (* Change TcAmsLog.exe mode/state *)
  eNewMode := SEL( bRecord, AMSLOGGER_STOP, AMSLOGGER_RUN);
  IF ( eNewMode <> eCurrMode ) THEN
    fbAmsLogger( bExecute := FALSE );
    fbAmsLogger( eMode:= eNewMode, bExecute := TRUE );
    state := 4;
  END_IF

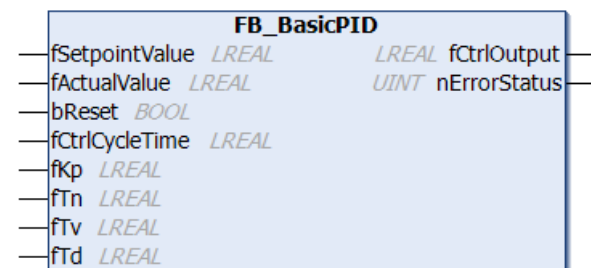
4: (* Wait until command execution started *)
  fbAmsLogger( bExecute := FALSE, bBusy=>bBusy, bError=>bError, nErrID=>nErrID );
  IF NOT bBusy THEN
    IF NOT bError THEN (* Success *)
      eCurrMode := eNewMode;
      state := 2;
    ELSE (* Error *)
      state := 100;
    END_IF
  END_IF

100: (* Error state *)
  ;
END_CASE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

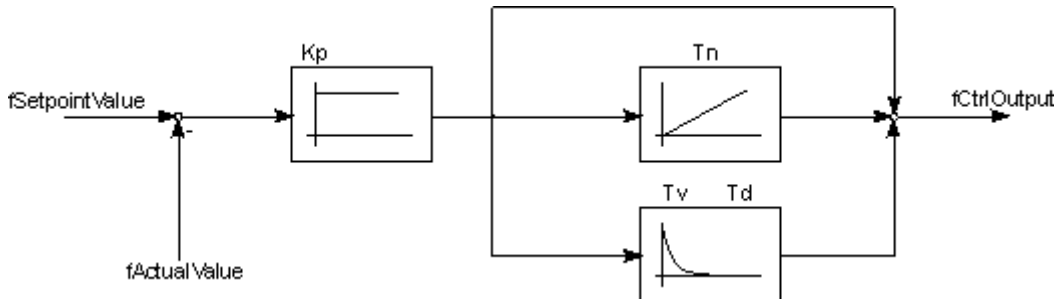
3.10 FB_BasicPID



Der Funktionsbaustein stellt ein einfaches diskretisiertes PID-Glied dar.

Übertragungsfunktion:

$$G(s) = K_p \left(1 + \frac{1}{T_n s} + \frac{T_v s}{1 + T_d s} \right)$$

Wirkungsplan:**VAR_INPUT**

```

VAR_INPUT
  fSetpointValue : LREAL; (* setpoint value *)
  fActualValue   : LREAL; (* actual value *)
  bReset         : BOOL;
  fCtrlCycleTime : LREAL; (* controller cycle time in seconds [s] *)
  fKp            : LREAL; (* proportional gain Kp (P) *)
  fTn            : LREAL; (* integral gain Tn (I) [s] *)
  fTv            : LREAL; (* derivative gain Tv (D-T1) [s] *)
  fTd            : LREAL; (* derivative damping time Td (D-T1) [s] *)
END_VAR

```

fSetpointValue : Sollwert der Regelgröße.

fActualValue : Istwert der Regelgröße.

bReset : Ein TRUE an diesem Eingang setzt die internen Zustandsgrößen sowie den Ausgang des Reglers zurück.

fCtrlCycleTime : Zykluszeit, mit der der Funktionsbaustein aufgerufen wird und der Regelkreis bearbeitet wird [s].

Hier **muss** zwingend die Zykluszeit der SPS-Task angegeben werden, wenn der Baustein in jedem SPS-Zyklus aufgerufen wird,

anderenfalls das entsprechende Vielfache der SPS-Task-Zykluszeit.

fKp : Reglerverstärkung / Reglerbeiwert

fTn : Nachstellzeit [s]

fTv : Vorhaltzeit [s]

fTd : Dämpfungszeit [s]

VAR_OUTPUT

```

VAR_OUTPUT
  fCtrlOutput : LREAL;
  nErrorStatus : UINT
END_VAR

```

fCtrlOutput : Ausgang des PID-Gliedes.

nErrorStatus : Liefert die Fehlernummer, wenn ein Fehler vorliegt (nErrorStatus <> 0).

Fehlercodes:

Wert	Konstante	Fehlerbeschreibung
0	nERR_NOERROR	Kein Fehler
1	nERR_INVALIDPARAM	Ungültige Parameter
2	nERR_INVALIDCYCLETIME	Ungültige Zykluszeit.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.11 FB_CalcHashValue

Dieser Funktionsbaustein berechnet einen Hash-Wert.

Hierzu werden die Methoden start(), update() und finish() verwendet.

Die Methoden entkoppeln das Anfügen von Eingangsdaten von der eigentlichen Berechnung des Hash-Wertes und ermöglichen zudem Eingangsdaten stückweise in mehreren Schritten anzufügen. Sollte kein mehrfaches Hinzufügen von Eingangsdaten notwendig sein, so wird die Verwendung der Funktion [F_GenerateHashValue\(\)](#) [[▶ 269](#)] anstatt dem Funktionsbaustein empfohlen.

Methode start()

Diese Methode initialisiert die Hash-Berechnung mit dem angegebenen Hash Mode.

```
METHOD start : BOOL
VAR_INPUT
    hashMode : E_HashMode;
END_VAR
```

hashMode: Hier wird ein Hash Mode, wie beispielsweise SHA 512, angegeben. Siehe [E_HashMode](#) [[▶ 303](#)].

Methode update()

Diese Methode kann einmalig oder mehrfach aufgerufen werden. Bei jedem Aufruf werden Eingangsdaten für die Hash-Berechnung hinzugefügt.

```
METHOD update : BOOL
VAR_INPUT
    pData : PVOID;
    nData : UDINT;
END_VAR
```

pData: Hier wird die Adresse der Eingangsdaten angegeben.

nData: Hier wird die Größe der Eingangsdaten in Bytes angegeben.

Methode finish()

Diese Methode führt die Hash-Berechnung aus und gibt den berechneten Hash-Wert aus.

```
VAR_INPUT
    pHash : PVOID;
    nHash : UDINT;
END_VAR
```

pHash: Hier wird die Adresse des Puffers angegeben, wohin der Hash-Wert gespeichert werden soll.

nHash: Hier wird die Größe des Puffers für den Hash-Wert angegeben. Die Größe ist abhängig vom Hash Mode, siehe auch [E_HashMode](#) [[▶ 303](#)].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024.29	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.51.0

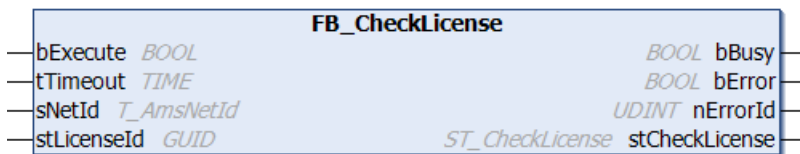
Sehen Sie dazu auch

- 📖 ST_FindFileEntry [▶ 316]
- 📖 ADS Return Codes [▶ 360]
- 📖 F_GenerateHashValue [▶ 269]

3.12 FB_CheckLicense

● **Nutzung von OEM-Lizenzen = Bootprojekt verschlüsseln!**

i Denken Sie daran, dass die per FB_CheckLicense abgefragte License-ID im Binärcode mit einem Hex-Editor leicht gefunden und (mit einem gewissen Aufwand) manipuliert werden kann. Arbeiten Sie daher unbedingt mit einer Verschlüsselung des Bootprojektes (am sichersten), oder verschleiern Sie zumindest die abgefragte License-ID im Quellcode bestmöglich.



Der Funktionsbaustein FB_CheckLicense ermittelt den TwinCAT-3-Lizenzstatus für eine gegebene Lizenz-ID.

VAR_INPUT

```
VAR_INPUT
    bExecute      : BOOL;
    tTimeout      : TIME;
    sNetId        : T_AmsNetId;
    stLicenseId   : GUID;
END_VAR
```

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Timeout-Zeit, die bei Ausführung des Befehls nicht überschritten werden darf.

sNetId: AmsNetId (AMS-Netzwerkennung) des TwinCAT-Rechners, dessen Lizenzstatus ausgelesen werden soll (Typ: T_AmsNetId). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

stLicenseId: Lizenz-ID (Typ: GUID)

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy         : BOOL;
    bError        : BOOL;
    nErrorId      : UDINT;
    stCheckLicense : ST_CheckLicense
END_VAR
```

bBusy: TRUE, solange der Baustein aktiv ist.

bError: TRUE, wenn bei der Ausführung des Kommandos ein Fehler auftritt.

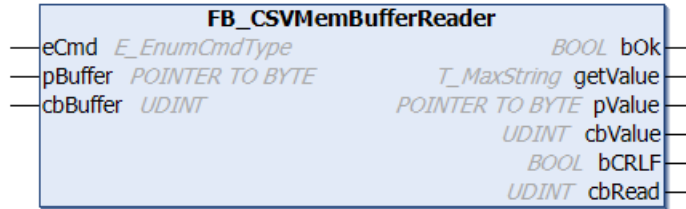
nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

stCheckLicense: Struktur mit Lizenzdaten (Typ: ST_CheckLicense)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022	PC oder CX (x64, x86)	Tc2_Utilities >= 3.3.24.0

3.13 FB_CSVMemBufferReader



Mit diesem Funktionsbaustein können Datensätze die in einem externen Puffer liegen in einzelne Datenfelder zerlegt/interpretiert werden. Die Pufferdaten könnten z.B. zuerst mit Hilfe der Funktionsbausteine für den Dateizugriff aus einer Datei gelesen werden. Der Funktionsbaustein liest das erste oder das nächste Datenfeld und gibt dessen Wert entweder als String am *getValue*-Ausgang oder als Adresse/Bytegröße am *pValue/cbValue*-Ausgang zurück.

Die Daten im Puffer müssen einem bestimmten Format entsprechen damit sie von dem Funktionsbaustein richtig interpretiert werden können. Zur Trennung der Datensätze muss das Datensatztrennzeichen CRLF (CR=Carriage Return, LF=Line Feed) verwendet werden. Der letzte Datensatz muss ebenfalls mit einem CRLF abschließen. Einzelne Datenfelder müssen mit dem Datenfeldtrennzeichen getrennt worden sein. In der Standardeinstellung wird dafür das Datenfeldtrennzeichen Semikolon verwendet. Über die globale SPS-Variable **DEFAULT_CSV_FIELD_SEP** kann das Trennzeichen vom Semikolon auf Komma konfiguriert werden.

VAR_INPUT

```

VAR_INPUT
    eCmd      : E_EnumCmdType := eEnumCmd_First;
    pBuffer   : POINTER TO BYTE;
    cbBuffer  : UDINT;
END_VAR
    
```

eCmd: Steuerparameter für den Pufferbaustein (Typ: [E_EnumCmdType](#) | 303]). *eEnumCmd_First* liest das erste Datenfeld, *eEnumCmd_Next* liest das nächste Datenfeld. Andere Parameterwerte werden nicht benutzt.

pBuffer: Adresse (Pointer) auf die Quellpuffervariable. Die Adresse kann mit dem ADR-Operator ermittelt werden. Dieser Puffer beinhaltet die zu lesenden Datensatz/Datenfeld-Daten.

cbBuffer: Die Bytegröße der zu interpretierenden Daten im Quellpuffer (Datensatz/Datenfeld-Daten). Die Puffergröße kann viel größer sein als die reine zu interpretierende Datenmenge. Bitte geben Sie hier nur die tatsächliche Länge der zu interpretierenden Daten an.

VAR_OUTPUT

```

VAR_OUTPUT
    bOk       : BOOL;
    getValue  : T_MaxString := '';
    pValue    : POINTER TO BYTE := 0;
    cbValue   : UDINT := 0;
    bCRLF     : BOOL := FALSE;
    cbRead    : UDINT := 0;
END_VAR
    
```

bOk: TRUE = Success, FALSE = Fehlerhafte Daten/fehlerhafte Eingangsparameter oder wenn das Ende der Daten erreicht wurde und kein weiteres Datenfeld gelesen werden konnte.

getValue: Das zuletzt gelesene Datenfeld als String (Typ: *T_MaxString*). Bei Datenfeldern ohne Steuerzeichen und Binärdaten liefert dieser Ausgang das vollständige Datenfeld als Nullterminierter String. Datenfelder mit Steuerzeichen oder Binärdaten können aber an diesem Ausgang einen unvollständigen String zurückliefern. In diesem Fall werden die Ausgänge *pValue/cbValue* für den Zugriff auf das zuletzt gelesene Datenfeld benutzt.

pValue: Adresse (Pointer) auf das erste Datenbyte des Datenfelds. Bitte beachten Sie: Leere Datenfelder werden nicht wie bei einem SPS -String üblich mit einer Null terminiert und besitzen daher keine Daten. Die Adresse ist in diesem Fall Null.

cbValue: Datenfeldlänge in Bytes. Bitte beachten Sie: Leere Datenfelder werden nicht wie bei einem SPS-String üblich mit einer Null terminiert und besitzen daher keine Daten. Die Länge ist in diesem Fall auch Null. Die maximale Größe kann mit dem globalen Parameter `cMaxCSVFieldValueSize` angegeben werden.

bCRLF: Dieser Ausgang wird gesetzt, wenn beim letzten Lesebefehl das Datensatzende erreicht wurde. Das zuletzt gelesene Datenfeld gehört noch zum vorherigen Datensatz. Das nächste Datenfeld gehört dann zu einem neuen Datensatz.

cbRead: Anzahl der erfolgreich gelesenen/interpretieren Datenbytes. Diese Zahl kann größer sein als die Datenfeldlänge am `cbValue`-Ausgang. Die Länge am `cbRead`-Ausgang beinhaltet auch die interpretierten Datenfeld-/Datensatztrennzeichen.

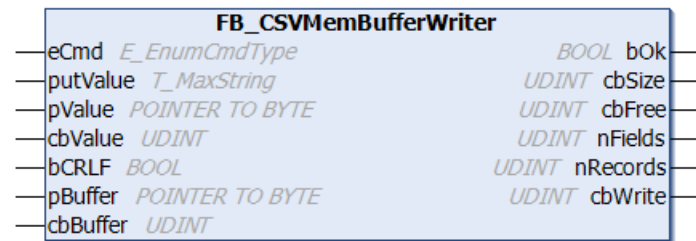
Beispiel:

Siehe: [Beispiel: Schreiben/lesen einer CSV-Datei \[▶ 351\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.14 FB_CSVMemBufferWriter



Mit diesem Funktionsbaustein können Datensätze im CSV-Format aus einzelnen Datenfeldern in einem externen Puffer erzeugt werden. Der Inhalt des Puffers kann danach z.B. mit Hilfe der Bausteine für den Dateizugriff in eine Datei geschrieben werden. Das neue Datenfeld kann entweder über die `putValue`-Variable (String) oder über die optionalen `pValue`- und `cbValue`-Variablen an den Baustein übergeben werden. Dies hängt davon ab ob Sie Datenfelder ohne Steuerzeichen (String) oder Datenfelder mit Steuerzeichen oder Binärdaten in den Datensatz schreiben wollen. Der Funktionsbaustein kann mehrere Datensätze im Puffer erzeugen bis die maximal verfügbare Puffergröße erreicht ist. Das Ende eines Datensatzes (letztes Datenfeld im aktuellen Datensatz) wird an das Datenfeld automatisch angehängt wenn beim Schreiben des Datenfeldes die `bCRLF`-Variable auf TRUE gesetzt wurde. Die Datenfeldtrennzeichen werden vom Baustein automatisch hinzugefügt. In der Standardeinstellung wird dafür das Datenfeldtrennzeichen Semikolon verwendet. Über die globale SPS-Variablen `DEFAULT_CSV_FIELD_SEP` kann das Trennzeichen vom Semikolon auf Komma konfiguriert werden.

VAR_INPUT

```
VAR_INPUT
    eCmd      : E_EnumCmdType := eEnumCmd_First;
    putValue  : T_MaxString := '';
    pValue    : POINTER TO BYTE := 0;
    cbValue   : UDINT := 0;
    bCRLF     : BOOL := FALSE;
    pBuffer   : POINTER TO BYTE;
    cbBuffer  : UDINT;
END_VAR
```

eCmd: Steuerparameter für den Pufferbaustein (Typ: `E_EnumCmdType` [▶ 303]). `eEnumCmd_First` fügt das erste Datenfeld in den Puffer, `eEnumCmd_Next` fügt das nächste Datenfeld hinzu. Andere Parameterwerte werden nicht benutzt.

putValue: Ein neues Datenfeld als String (Typ: T_MaxString). Dieser Eingang muss ein Leerstring sein wenn statt dieses Eingangs die optionalen Parameter *pValue* und *cbValue* benutzt werden.

pValue: Optional: Adresse eines externen Bytepuffers in dem sich das neue Datenfeld befindet. Zusammen mit dem *cbValue*-Parameter kann dieser Eingang dann benutzt werden, wenn z.B. ein Datenfeld mit Steuerzeichen oder Binärdaten in den Datensatz geschrieben werden soll. Die Steuerzeichen oder Binärdaten im Datenfeld könnten den *putValue*-String an einer ungewünschten Stelle abschneiden und werden deshalb als Bytepuffer übergeben. Dieser Eingang muss den Wert Null haben, wenn er nicht benutzt wird.

cbValue: Optional: Länge der Datenfelddaten im externen Bytepuffer. Dieser Eingang muss den Wert Null haben, wenn er nicht benutzt wird. Die maximale Größe kann mit dem globalen Parameter *cMaxCSVFieldValueSize* angegeben werden.

bCRLF: Wenn dieser Eingang gesetzt ist, dann wird das neue Datenfeld mit einem CRLF Datensatztrennzeichen abgeschlossen. Datenfelder die danach geschrieben werden gehören dann zum neuen Datensatz.

pBuffer: Adresse (Pointer) auf die Zielpuffervariable. Die Adresse kann mit dem ADR-Operator ermittelt werden. In diesem Puffer erzeugt der Funktionsbaustein die Datensätze im CSV-Format.

cbBuffer: Maximal verfügbare Größe (in Byte) der Zielpuffervariablen. Die Größe kann mit dem SIZEOF-Operator ermittelt werden.

VAR_OUTPUT

```
VAR_OUTPUT
  bOk      : BOOL;
  cbSize   : UDINT;
  cbFree   : UDINT;
  nFields  : UDINT;
  nRecords : UDINT;
  cbWrite  : UDINT;
END_VAR
```

bOk: TRUE = Success, FALSE = Pufferüberlauf oder fehlerhafte Eingangsparameter.

cbSize: Der aktuelle Puffer-Füllstatus (Anzahl der im Puffer erzeugten Datenbytes).

cbFree: Anzahl der freien Datenbytes im Puffer.

nFields: Anzahl der geschriebenen Datenfelder.

nRecords: Anzahl der geschriebenen Datensätze.

cbWrite: Anzahl der zuletzt geschriebenen Datenbytes (die Länge des letzten Datenfelds + eventuelle Datensatz- oder Datenfeld-Trennzeichen).

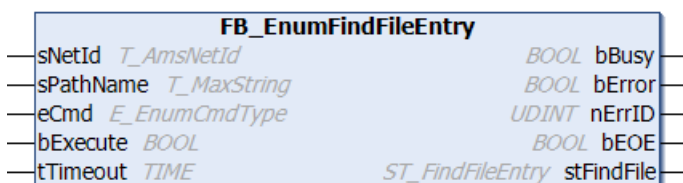
Beispiel:

Siehe: [Beispiel: Schreiben/lesen einer CSV-Datei](#) [► 351].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.15 FB_EnumFindFileEntry



Dieser Funktionsbaustein durchsucht ein Verzeichnis nach einer Datei oder nach einem Unterverzeichnis, dessen Name dem spezifizierten Namen gleicht. Die gefundenen Einträge können einzeln ausgelesen werden. Siehe auch die Beschreibung des FB_EnumFindFileList-Funktionsbausteins. Mit dem Eingangsparameter *eCmd* kann durch die Liste der Einträge navigiert werden. Der *eCmd*-Eingang bestimmt z. B., ob der erste oder nächste Eintrag gelesen werden soll.

Wichtige Hinweise:

Eine neue Suche darf nur dann gestartet werden, wenn die vorherige Suche komplett abgeschlossen wurde. Für eine komplette Suche muss die Bausteininstanz eventuell mehrere Male aktiviert werden (durch steigende Flanke am *bExecute*-Eingang). Die Suche ist nur dann komplett abgeschlossen wenn *bEOE* = *TRUE* erreicht wurde oder wenn die Suche mit *eCmd* = *eEnumCmd_Abort* vorzeitig abgebrochen wurde.

Für das TwinCAT System ist die Suche möglicherweise noch nicht abgeschlossen, wenn von der SPS-Applikation die gesuchte Datei oder das Verzeichnis bereits gefunden wurde.

Wenn nicht alle Einträge ausgelesen werden sollen (d. h. *bEOE=TRUE* wird nicht erreicht), muss der Funktionsbaustein anschließend mit dem Eingangsparameter *eCmd* = *eEnumCmd_Abort* aufgerufen werden. Dies ist nötig, um den Suchvorgang abzuschließen und alle internen Ressourcen (Datei-Handles) freizugeben. Wenn *bEOE=TRUE* erreicht wurde, oder ein Fehler auftritt, wird das *eEnumCmd_Abort* intern automatisch ausgeführt.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetID;
  sPathName   : T_MaxString;
  eCmd        : E_EnumCmdType := eEnumCmd_First;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Verzeichnis durchsucht werden soll (Typ: *T_AmsNetID*). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

sPathName: Ein gültiger Verzeichnisname oder Verzeichnis mit Dateinamen als String (Typ: *T_MaxString*). Der String kann folgende Platzhalter (* und ?) enthalten. Wenn der Pfad mit einem Platzhalter, Punkt oder dem Verzeichnisnamen endet muss der Benutzer Zugriffsrechte auf diesen Pfad und dessen Unterverzeichnisse haben.

eCmd: Kommandoparameter für den Aufzählungsbaustein (Typ: [E_EnumCmdType](#) [[▶ 303](#)]).

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrId      : UDINT;
  bEOE        : BOOL;
  stFindFile  : ST_FindFileEntry;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten *bError*-Ausgang die [ADS-Fehlernummer](#) [[▶ 360](#)].

bEOE: Ende der Aufzählung wurde erreicht (end of enumeration). Beim ersten Versuch einen nicht existierenden Eintrag zu lesen wird dieser Ausgang auf *TRUE* gesetzt. D.h. so lange *bEOE* = *FALSE* und *bError* = *FALSE* ist, sind auch die gelesenen Einträge gültig.

stFindFile: Bei Erfolg liefert diese Strukturvariable Informationen zur gefundenen Datei (Typ: *ST_FindFileEntry* [▶ 316]).

Beispiel:

Siehe: Beispiel: Dateisuche (FB_EnumFindFileEntry, FB_EnumFindFileList). [▶ 335]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/ BSD: TC RT x64, TC OS ARMT2)	Tc2_Uilities (System)

3.16 FB_EnumFindFileList



Dieser Funktionsbaustein durchsucht ein Verzeichnis nach einer Datei oder nach einem Unterverzeichnis, dessen Name dem spezifizierten Namen gleicht. Die gefundenen Einträge können einzeln ausgelesen werden. Siehe auch die Beschreibung des *FB_EnumFindFileEntry*-Funktionsbausteins. Mit dem Eingangsparameter *eCmd* kann durch die Liste der Einträge navigiert werden. Der *eCmd*-Eingang bestimmt z. B., ob der erste oder nächste Eintrag gelesen werden soll.

Wichtiger Hinweis:

Eine neue Suche darf nur dann gestartet werden, wenn die vorherige Suche komplett abgeschlossen wurde. Für eine komplette Suche muss die Bausteininstanz eventuell mehrere Male aktiviert werden (durch steigende Flanke am *bExecute*-Eingang). Die Suche ist nur dann komplett abgeschlossen, wenn *bEOE* = *TRUE* erreicht wurde oder wenn die Suche mit *eCmd* = *eEnumCmd_Abort* vorzeitig abgebrochen wurde.

Für das TwinCAT System ist die Suche möglicherweise noch nicht abgeschlossen, wenn von der SPS-Applikation die gesuchte Datei oder das Verzeichnis bereits gefunden wurde.

Wenn nicht alle Einträge ausgelesen werden sollen (d. h. *bEOE* = *TRUE* wird nicht erreicht), muss der Funktionsbaustein anschließend mit dem Eingangsparameter *eCmd* = *eEnumCmd_Abort* aufgerufen werden. Dies ist nötig, um den Suchvorgang abzuschließen und alle internen Ressourcen (Datei-Handles) freizugeben. Wenn *bEOE* = *TRUE* erreicht wurde, oder ein Fehler auftritt, wird das *eEnumCmd_Abort* intern automatisch ausgeführt.

VAR_INPUT

```

VAR_INPUT
  sNetID      : T_AmsNetID;
  sPathName   : T_MaxString;
  eCmd        : E_EnumCmdType := eEnumCmd_First;
  pFindList   : POINTER TO ST_FindFileEntry;
  cbFindList  : UDINT;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
    
```

sNetID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Verzeichnis durchsucht werden soll (Typ: *T_AmsNetID*). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

sPathName: Ein gültiger Verzeichnisname oder Verzeichnis mit Dateinamen als String (Typ: `T_MaxString`). Der String kann folgende Platzhalter (* und ?) enthalten. Wenn der Pfad mit einem Platzhalter, Punkt oder dem Verzeichnisnamen endet, muss der Benutzer Zugriffsrechte auf diesen Pfad und dessen Unterverzeichnisse haben.

eCmd: Steuerkommando für den Aufzählungsbaustein (Typ: `E_EnumCmdType` [▶ 303]).

pFindList: Adresse (Pointervariable) einer Arrayvariablen von Typ: `ST_FindFileEntry` [▶ 316].

cbFindList: Bytegröße der Arrayvariablen vom Typ: `ST_FindFileEntry` [▶ 316].

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrId    : UDINT;
  bEOE      : BOOL;
  nFindFiles : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [▶ 360].

bEOE: Ende der Aufzählung wurde erreicht (end of enumeration). Beim ersten Versuch einen nicht existierenden Eintrag zu lesen wird dieser Ausgang auf TRUE gesetzt. D.h. so lange *bEOE* = FALSE und *bError* = FALSE ist, sind auch die gelesenen Einträge gültig.

nFindFiles: Anzahl der gültigen Einträge im Puffer.

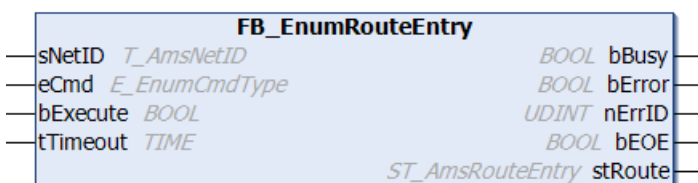
Beispiel:

Siehe: Beispiel: Dateisuche (FB_EnumFindFileEntry, FB_EnumFindFileList) [▶ 335].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Uilities (System)

3.17 FB_EnumRouteEntry



Mit dem Funktionsbaustein können Informationen über die AMS Router Verbindungen zu anderen TwinCAT Systemen (Remote Routes) ausgelesen werden. Bei mehreren Verbindungen muss der Funktionsbaustein mehrere Male aufgerufen werden und pro Aufruf kann maximal ein Eintrag ausgelesen werden. Mit dem Eingangsparameter *eCmd* kann durch die Liste der Einträge navigiert werden. Der *eCmd*-Eingang bestimmt ob der erste oder nächste Eintrag gelesen werden soll.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetID;
  eCmd        : E_EnumCmdType := eEnumCmd_First;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen AMS Router Verbindungen gelesen werden sollen (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

eCmd: Steuerkommando für den Aufzählungsbaustein (Typ: E_EnumCmdType [► 303]).

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrId      : UDINT;
  bEOE       : BOOL;
  stRoute     : ST_AmsRouteEntry;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [► 360].

bEOE: Ende der Aufzählung wurde erreicht (end of enumeration). Beim ersten Versuch einen nicht existierenden Eintrag zu lesen wird dieser Ausgang auf TRUE gesetzt. D.h. solange *bEOE* = FALSE und *bError* = FALSE sind, sind auch die gelesenen Einträge gültig.

stRoute: Strukturelement mit zuletzt gelesenen Verbindungsparametern (Typ: ST_AmsRouteEntry [► 311]).

Beispiel:

Auf dem lokalen TwinCAT System sollen die konfigurierten AMS Router Verbindungen ausgelesen und als Meldungen in die TwinCAT System Manager Logger Ausgabe geschrieben werden.

```
PROGRAM P_EnumRouteEntries
VAR
  fbEnum : FB_EnumRouteEntry := ( sNetID := '', tTimeout := T#5s );
  bEnum  : BOOL := TRUE;
  nState : BYTE := 0;
  sInfo  : T_MaxString;
END_VAR
```

Bei einer steigenden Flanke an der *bEnum*-Variablen werden die Verbindungsinformationen ausgelesen.

```
CASE nState OF
  0:
  IF bEnum THEN (* flag set ? *)
    bEnum := FALSE; (* reset flag *)
    fbEnum.eCmd := eEnumCmd_First; (* enum first entry *)
    nState := 1;
  END_IF
```

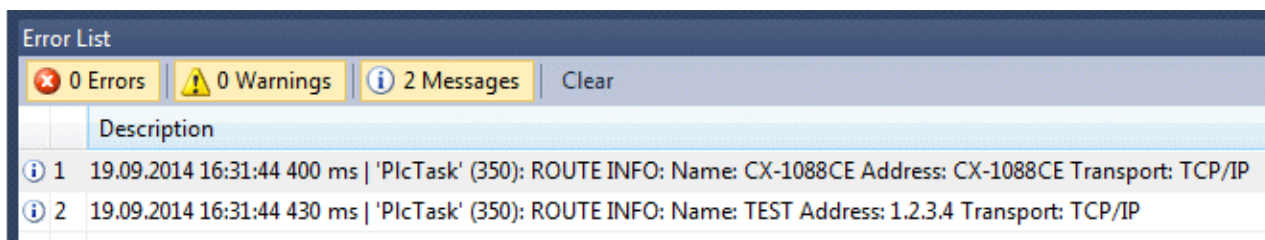


```

1: (* enum one entry *)
fbEnum( bExecute := FALSE );
fbEnum( bExecute := TRUE );
nState := 2;

2: (* wait until function block not busy *)
fbEnum( bExecute := FALSE );
IF NOT fbEnum.bBusy THEN
    IF NOT fbEnum.bError THEN
        IF NOT fbEnum.bEOE THEN
            sInfo := CONCAT( 'Name: ', fbEnum.stRoute.sName );
            sInfo := CONCAT( sInfo, ' Address: ' );
            sInfo := CONCAT( sInfo, fbEnum.stRoute.sAddress );
            sInfo := CONCAT( sInfo, ' Transport: ' );
            sInfo := CONCAT( sInfo, ROUTETransportToString( fbEnum.stRoute.eTransport ) );
            ADSLOGSTR( ADSLOG_MSGTYPE_HINT OR ADSLOG_MSGTYPE_LOG, 'ROUTE INFO: %s', sInfo );
            fbEnum.eCmd := eEnumCmd_Next; (* enum next entry *)
            nState := 1;
        ELSE (* no more route entries *)
            nState := 0;
        END_IF
    ELSE (* log error *)
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'FB_EnumRouteEntry error:
%s', DWORD_TO_HEXSTR( fbEnum.nErrID, 0, FALSE ) );
        nState := 0;
    END_IF
END_IF
END_CASE
    
```

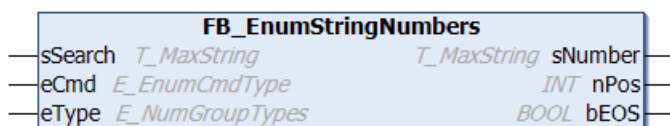
Die geschriebenen Logmeldungen in der TwinCAT System Manager Logger Ausgabe:



Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.18 FB_EnumStringNumbers



Mit diesem Funktionsbaustein kann ein String in einer REPEAT- oder WHILE-Schleife nach Zahlen durchsucht werden. Der String kann mehrere Zahlen beinhalten. Die gefundenen Zahlen werden als Teilstrings am Baustein-Ausgang ausgegeben. Es wird ab der aktuellen Position, nach dem ersten Zeichen, das als Zahlenzeichen interpretiert werden kann, gesucht. Die Suche wird beim ersten nicht als Zahl interpretierbaren Zeichen abgebrochen. Der eCmd-Parameter bestimmt, ob nach der ersten oder der nächsten Zahl gesucht werden soll. Der eType-Parameter bestimmt das Zahlenformat der Zahlen im Suchstring.

VAR_INPUT

```

VAR_INPUT
    sSearch : T_MaxString;
    eCmd    : E_EnumCmdType := eEnumCmd_First;
    eType   : E_NumGroupTypes := eNumGroup_Float;
END_VAR
    
```

sSearch: Suchstring in dem nach Zahlen gesucht werden soll (Typ: T_MaxString).

eCmd: Steuerkommando für den Aufzählungsbaustein (Typ: E_EnumCmdType [► 303]).

eType: Zahlenformat der gesuchten Zahl (Typ: E_NumGroupTypes [► 305]). Dieser Parameter bestimmt welche Zeichen ignoriert und welche als Zahlenzeichen interpretiert werden:

Wert	Bedeutung
eNumGroup_Float	Zahlen '0' bis '9', '+', '-' (Vorzeichen) und das 'e' oder 'E' (Exponentenzeichen) werden als gültige Zahlenzeichen interpretiert.
eNumGroup_Unsigned	Zahlen '0' bis '9' werden als gültige Zahlenzeichen interpretiert. '+', '-', 'e', 'E'-Zeichen werden als Zahlenzeichen ignoriert.
eNumGroup_Signed	Zahlen '0' bis '9' '+', '-' (Vorzeichen) werden als gültige Zahlenzeichen interpretiert. 'e' und 'E'-Zeichen werden als Zahlenzeichen ignoriert.

Beinhaltet der String z.B. Zahlen in der Exponentialdarstellung dann muss eType = eNumGroup_Float gesetzt werden (default).

VAR_OUTPUT

```
VAR_OUTPUT
    sNumber      : T_MaxString;
    nPos         : INT;
    bEOS        : BOOL;
END_VAR
```

sNumber: Die zuletzt gefundene Zahl als String (Typ: T_MaxString).

nPos: Diese Variable liefert immer die Position hinter dem zuletzt gefundenen und richtig formatierten Zahlenzeichen. D.h. an dieser Stelle beginnt der Baustein beim nächsten Aufruf nach neuen Zahlenzeichen zu suchen. nPos ist Null wenn die abschließende Null des sSearch-Strings erreicht wurde. Das erste Zeichen im String hat die Positionsnummer = 1 (non-zero based position).

bEOS: Diese Variable ist FALSE wenn das Ende des Strings noch nicht erreicht wurde und eine neue Zahl gefunden wurde. In diesem Fall liefert sNumber eine gültige Zahl als String. Diese Variable ist TRUE wenn keine weitere Zahl gefunden wurde. Eine weitere Suche muss in diesem Fall abgebrochen werden (sNumber liefert keinen gültigen Wert).

Beispiel:

Im folgenden Beispiel wird die sNumber-Variable nach gültigen Zahlen durchsucht. Die gefundenen Teilstrings werden in der Array-Variable arrNums abgespeichert.

```
TYPE ST_ScanRes :
STRUCT
    sNumber      : T_MaxString;
    nPos         : INT;
    sRemain      : T_MaxString;
END_STRUCT
END_TYPE

PROGRAM MAIN
VAR
    sSearch      : T_MaxString := 'Some numbers in string: +-12e-34, -56, +78';
    fbEnum       : FB_EnumStringNumbers := ( eType := eNumGroup_Float (* eNumGroup_Signed, eNumGroup_Unsigned *) );
    arrNums      : ARRAY[1..MAX_SCAN_NUMS] OF ST_ScanRes;
    idx          : INT;
    length       : INT;
    bEnum        : BOOL := TRUE;
END_VAR
VAR_CONSTANT
    MAX_SCAN_NUMS : INT := 10;
END_VAR

IF bEnum THEN
    bEnum := FALSE;

    MEMSET( ADR( arrNums ), 0, SIZEOF( arrNums ) );
```

```

idx := 0;
length := LEN( sSearch );

fbEnum( sSearch := sSearch, eCmd := eEnumCmd_First );
WHILE NOT fbEnum.bEOS DO
  IF idx < MAX_SCAN_NUMS THEN
    idx := idx + 1;

    arrNums[idx].sNumber:= fbEnum.sNumber;
    arrNums[idx].nPos := fbEnum.nPos;
    IF fbEnum.nPos <> 0 THEN
      arrNums[idx].sRemain:= RIGHT( sSearch, length - fbEnum.nPos + 1 );
    END_IF

  END_IF
  fbEnum( eCmd := eEnumCmd_Next );
END_WHILE
END_IF

```

Die gefundenen Strings:

- '-12e-34'
- '-56'
- '+78'

eType-Parameter eNumGroup_Signed liefert folgende Ergebnisse:

- '-12'
- '-34'
- '-56'
- '+78'

eType-Parameter eNumGroup_Unsigned liefert folgende Ergebnisse:

- '12'
- '34'
- '56'
- '78'

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.19 FB_FileProperties



Dieser Funktionsbaustein gibt die Dateieigenschaften einer ausgewählten Datei aus.

VAR_INPUT

```

VAR_INPUT
  sNetID      : T_AmsNetID;
  sPathName   : T_MaxString;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

```

sNetID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Verzeichnis durchsucht werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

sPathName: Hier wird der komplette Dateipfad inklusive Dateiname und Dateierdung angegeben als String (Typ: T_MaxString).

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrId     : UDINT;
  stProperties : ST_FindFileEntry;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

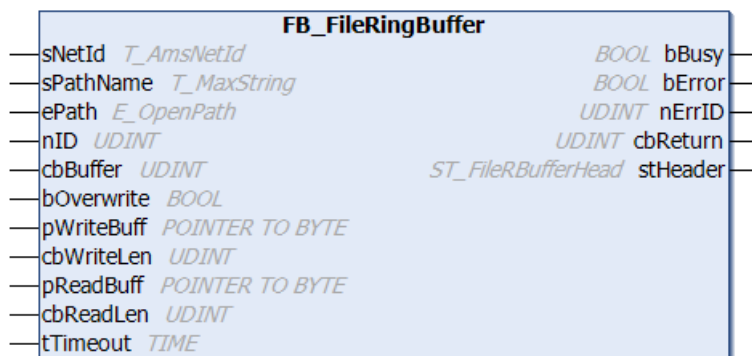
nErrId: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [▶ 360].

stProperties: Bei Erfolg liefert diese Strukturvariable Informationen zur gefundenen Datei (Typ: ST_FindFileEntry [▶ 316]).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/ BSD: TC RT x64, TC OS ARMT2)	Tc2_Uilities (System)

3.20 FB_FileRingBuffer



Mit dem Funktionsbaustein FB_FileRingBuffer können Datensätze unterschiedlicher Länge in eine Ringpufferdatei geschrieben oder die vorher geschriebenen Datensätze aus der Ringpufferdatei ausgelesen werden. Die geschriebenen Datensätze werden nach dem FIFO-Prinzip in der gleichen Reihenfolge ausgelesen in der sie vorher in die Ringpufferdatei geschrieben wurden. D.h. beim Lesen werden zuerst die ältesten Einträge ausgelesen. Das Öffnen/Schließen/Schreiben/Lesen der Datensätze wird durch Aktionsaufrufe gesteuert. Der Funktionsbaustein besitzt folgende Aktionen:

- **A_Open** (Öffnet eine vorhandene Ringpufferdatei zum Anhängen oder Erzeugen neuer Datensätze. Wenn die Datei bereits geöffnet ist wird kein Fehler zurückgemeldet.)
- **A_Close** (Schließt eine geöffnete Ringpufferdatei. Wenn die Datei bereits geschlossen ist wird kein Fehler zurückgemeldet.)

- **A_Create** (Öffnet eine neue Ringpufferdatei. Wenn die Datei bereits existiert wird sie überschrieben. Wenn die Datei bereits geöffnet ist wird kein Fehler zurückgeliefert)
- **A_AddTail** (Schreibt einen neuen Datensatz in die Ringpufferdatei.)
- **A_GetHead** (Liest den ältesten Datensatz aus der Ringpufferdatei, entfernt ihn aber nicht (der Dateizeiger wird nicht zum nächsten Datensatz bewegt).)
- **A_RemoveHead** (Liest und entfernt den ältesten Datensatz aus der Ringpufferdatei (der Dateizeiger wird zum nächsten Datensatz bewegt).)
- **A_Reset** (Löscht alle Datensätze in einer geöffneten Ringpufferdatei. Es werden lediglich die Dateizeiger und die Anzahl der Datensätze zurückgesetzt, die vorhandene physikalische Dateigröße wird nicht verändert (die alten Datensätze werden aber mit neuen überschrieben).

Beim Öffnen der Ringpufferdatei wird bei einer existierenden Datei zuerst der Dateiheder gelesen. Bei einer Ringpufferdatei die vorher fehlerfrei geschlossen wurde muss das Bit 0 im Header.status.Bit 0 = Null sein. Bei einem anderen Wert wird angenommen, dass die Datei vorher nicht richtig geschlossen wurde und die korrupte Datei durch eine neue leere Datei ersetzt sowie das Header.status.Bit 1 = 1 gesetzt (file corrupted). Beim Schließen der Datei wird das Header.status.Bit 0 = 0 gesetzt und der komplette Dateiheder in der Datei aktualisiert.

VAR_INPUT

```
VAR_INPUT
  sNetId      : T_AmsNetId := '';
  sPathName   : T_MaxString := 'c:\Temp\data.dat';
  ePath       : E_OpenPath := PATH_GENERIC;
  nID         : UDINT := 0;
  cbBuffer    : UDINT := 16#100000;
  bOverwrite  : BOOL := FALSE;
  pWriteBuff  : POINTER TO BYTE;
  cbWriteLen  : UDINT;
  pReadBuff   : POINTER TO BYTE;
  cbReadLen   : UDINT;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem die Pufferdatei geschrieben/gelesen werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

sPathName: Enthält den Pfad- und Dateinamen der zu öffnenden Pufferdatei (Typ: T_MaxString). **Hinweis:** Der Pfad kann nur auf das lokale File System des Rechners zeigen! Das bedeutet, Netzwerkpfade können hier nicht angegeben werden!

ePath: Über diesen Eingang kann ein TwinCAT - Systempfad auf dem Zielgerät zum Öffnen der Datei angewählt werden (Typ: E_OpenPath).

nID: Benutzerdefinierter 32 Bit Wert. Dieser Wert wird beim Öffnen einer neuen Datei in die Datei gespeichert und kann z.B. zur Versionsüberprüfung der Pufferdatei benutzt werden.

cbBuffer: Max. Bytegröße der zu öffnenden Pufferdatei. Dieser Parameter wird beim Erstellen der Datei in dem Dateiheder gespeichert und wird beim erneuten Öffnen derselben Datei überprüft. Sie können nur Dateien erneut öffnen, die mit der gleichen max. Puffergröße erstellt wurden. D.h. Sie können nicht eine Datei mit einer kleineren Puffergröße erstellen, mit Datensätzen füllen und mit einer größeren Puffergröße erneut öffnen. Wenn die Überprüfung der max. Puffergröße fehlschlägt wird automatisch eine neue Datei mit der neuen Puffergröße erstellt und geöffnet. Zusätzlich wird das Bit 1 (file corrupted) in dem Datei-Header-Status gesetzt.

bOverwrite: Schreibverhalten beim Erreichen der max. Dateipuffergröße. Bei TRUE an diesem Eingang werden die ältesten Einträge überschrieben wenn die max. Dateipuffergröße bereits erreicht wurde (es werden so lange Einträge gelöscht bis die freie Puffergröße ausreichend ist um den neuen Eintrag zu speichern). Beim FALSE an diesem Eingang wird ein Pufferüberlauf beim Erreichen der max. Dateipuffergröße als Fehler gemeldet.

pWriteBuff: Adresse der SPS-Variablen oder einer Puffervariablen, die die zu schreibenden Value-Daten enthält. Die Adresse kann mit dem ADR-Operator ermittelt werden. Der Programmierer ist selbst dafür verantwortlich die Puffervariable so zu dimensionieren, dass *cbWriteLen*-Datenbytes daraus entnommen werden können.

cbWriteLen: Anzahl der zu schreibenden Value-Datenbytes (Bei Stringvariablen inklusive der abschließenden Null).

pReadBuff: Adresse der SPS-Variablen oder einer Puffervariablen in welche die gelesenen Value-Daten hineinkopiert werden sollen. Die Adresse kann mit dem ADR-Operator ermittelt werden. Der Programmierer ist selbst dafür verantwortlich die Puffervariable so zu dimensionieren, dass diese cbReadLen-Datenbytes aufnehmen kann. Die Bytegröße der Puffervariablen muss größer oder gleich der Größe des zu lesenden Datensatzes sein.

cbReadLen: Anzahl der zu lesenden Value-Datenbytes. Bei einer zu kleinen Puffergröße werden keine Daten kopiert, der Funktionsbaustein meldet einen Puffer-Underflow-Fehler und die benötigte Puffergröße für den nächsten zu lesenden Datensatz wird am *cbReturn*-Ausgang zurückgeliefert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrId    : UDINT;
  cbReturn   : UDINT;
  stHeader   : ST_FileRBufferHead;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [▶ 360] oder den Befehlsspezifischen Fehlercode (Tabelle).

cbReturn: Anzahl der erfolgreich gelesenen Value-Datenbytes. Beim Lesebuffer-Underflow-Fehler liefert dieser Ausgang die benötigte Lesebuffer-Bytegröße.

stHeader: Ringpuffer-Dateiheader/-Status. (Typ: ST_FileRBufferHead [▶ 315])

Befehlsspezifischen Fehlercodes	Fehlerbeschreibung
0x8000	Schreiben: Dateipuffer ist leer. Lesen: Dateipuffer-Overflow.
0x8001	SPS-Applikation: Lesebuffer-Underflow (pReadBuff, cbReadLen) ist zu klein dimensioniert.
0x8002	Ringpufferdatei ist geschlossen und muss zuerst geöffnet werden.
0x8003	Falscher Wert des Eingangsparameters.

Die SPS-Applikation muss den binären Aufbau der Datei nicht zwingend kennen. Folgende Grafik zeigt aber den prinzipiellen Aufbau der verwendeten Ringpufferdatei:

Header (48 Byte) siehe in der Beschreibung von ST_FileRBufferHead	Länge Datensatz 1 (4 Byte)	Datensatz 1	Länge Datensatz 2 (4 Byte)	Datensatz 2	Länge Datensatz 3 (4 Byte)	Datensatz 3	Länge Datensatz n (4 Byte)	Datensatz n

Eine leere Ringpufferdatei beinhaltet nur den Header. Hinter dem Header folgt der eigentliche Puffer. Die Header.ptrFirst- und Header.ptrLast-Variable zeigt auf die nächste Position hinter dem Header. Beim Schreiben wird der ptrLast-Dateizeiger nach vorne bewegt. Der ptrFirst-Dateizeiger folgt dem ptrLast-Dateizeiger beim Lesen. Beim Erreichen der max. Puffergröße werden die Zeiger wieder zum Anfang des Puffers bewegt.

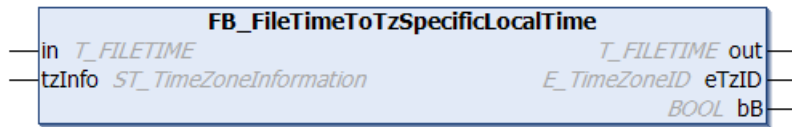
Beispiel:

Siehe: [Beispiel: Datei-Ring-FiFo \(FB_FileRingBuffer\) \[▶ 337\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.21 FB_FileTime64ToTzSpecificLocalTime



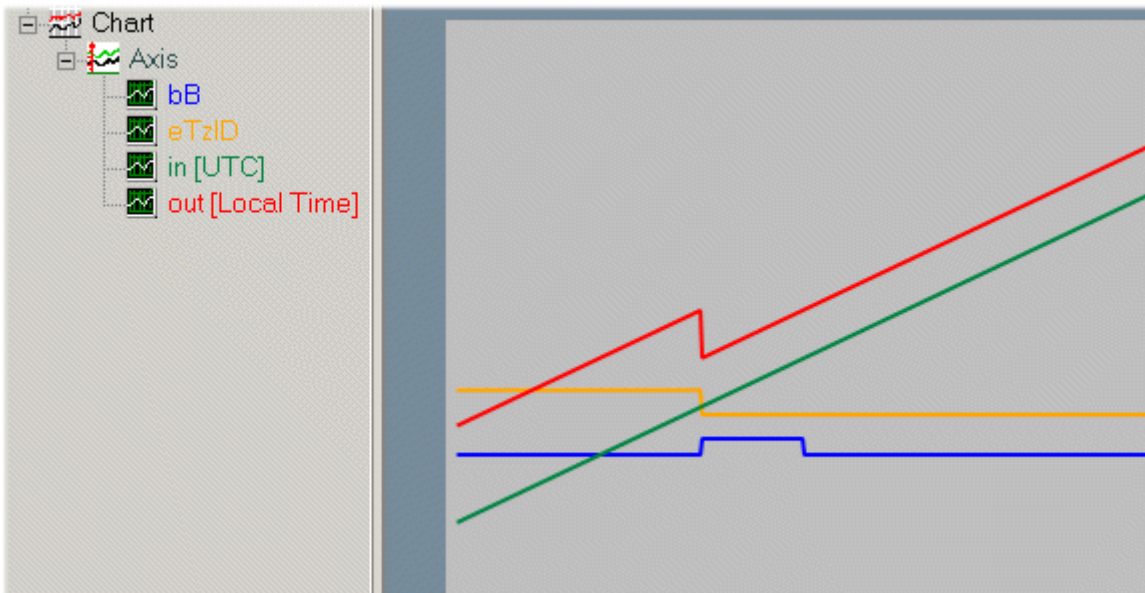
Der Funktionsbaustein konvertiert die UTC-Zeit (file time format) in Lokalzeit (file time format) unter der Berücksichtigung der angegebenen Zeitzoneninformation. Der Funktionsbaustein:

`FB_SystemTimeToTzSpecificLocalTime [▶ 113]` besitzt eine ähnliche Funktionalität, mit dem Unterschied dass er ein anderes Zeitformat (structured system time format) umrechnet.

Der Baustein eignet sich nur zur Konvertierung von **kontinuierlichen** UTC-Zeitstempelinformation. Anhand der Zeitzoneninformation errechnet der Funktionsbaustein die erforderlichen Zeitsprünge in der Lokalzeit (Sommer-/Winterzeit Umstellung). Zeitsprünge in der UTC-Eingangszeit sind nicht zulässig und führen zur fehlerhaften Konvertierung. Der Grund: Die zuletzt konvertierte Zeit wird im Baustein intern gespeichert um aus dem Verlauf der UTC-Eingangszeit und dem gespeicherten Wert die B-Zeiten (siehe unten) beim Zurückstellen der Lokalzeit erkennen zu können.

Der Baustein besitzt eine Aktion: `A_Reset()`. Ein Aufruf dieser Aktion setzt die Ausgänge des Bausteins und die intern gespeicherte (zuletzt konvertierte Zeit) auf Null zurück.

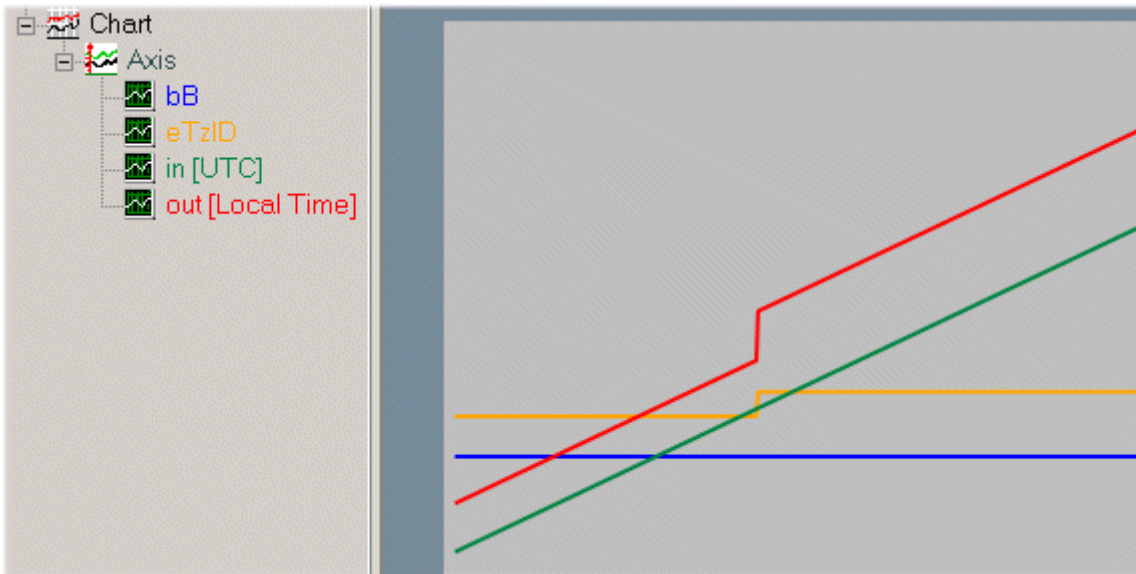
1. Grafische Darstellung des zeitlichen Verhaltens beim Übergang von Sommerzeit zur Winterzeit (hier `tzInfo = WEST_EUROPE_TZI`):



Die UTC-Eingangszeit (grün) verläuft kontinuierlich. Die Lokalzeit (rot) macht einen Sprung zurück. Der Lokalzeit: **02h:59m:59s:999ms..** folgt unmittelbar die Zeit: **02h:00m:00s:000ms..** Die Zeiten zwischen 2h und 3h sind zweifach vorhanden. Die doppelte Zeit vor der Zeitumstellung wird z.B. als **02:05:00 MEZ A** bezeichnet und die Zeit nach der Umstellung als **02:05:00 MEZ B**. Die Ausgangsvariable `bB` gibt Information darüber ob es sich um den ersten oder zweiten *Durchgang* handelt. Beim zweiten *Durchgang* ist die `bB-`

Ausgangsvariable (blau) auf TRUE gesetzt. Die *bB*-Ausgangsvariable wird automatisch zurückgesetzt nachdem die doppelte Zeit überschritten wurde. Die Zeitzonen-ID (orange) wechselt von *eTimeZoneID_Daylight* (Sommerzeit) zu *eTimeZoneID_Standard* (Winterzeit).

2. Grafische Darstellung des zeitlichen Verhaltens beim Übergang von Winterzeit zur Sommerzeit (hier tzInfo = WEST_EUROPE_TZI):



Die UTC-Eingangszeit (grün) verläuft kontinuierlich. Die Lokalzeit (grün) macht einen Sprung nach vorne. Der Lokalzeit: **2h:59m:59s:999ms..** folgt unmittelbar die Zeit: **3h:00m:00s:000ms..** Die Zeitzonen-ID (orange) wechselt von *eTimeZoneID_Standard* (Winterzeit) zu *eTimeZoneID_Daylight* (Sommerzeit).

VAR_INPUT

```
VAR_INPUT
  in      : T_FILETIME64;
  tzInfo  : ST_TimeZoneInformation;
END_VAR
```

in: UTC-Zeit (file time format), die konvertiert werden soll (Typ: [T_FILETIME64](#) [[▶ 323](#)]).

tzInfo: Strukturvariable mit der aktuellen Zeitzone-Information des Betriebssystems (Typ: [ST_TimeZoneInformation](#) [[▶ 321](#)]).

VAR_OUTPUT

```
VAR_OUTPUT
  out     : T_FILETIME64;
  eTzID   : E_TimeZoneID := eTimeZoneID_Unknown;
  bB      : BOOL;
END_VAR
```

out: Konvertierte Lokalzeit (file time format, Typ: [T_FILETIME64](#) [[▶ 323](#)]).

eTzID: Zusätzliche Sommer-/ Winterzeit-Information. (Typ: [E_TimeZoneID](#) [[▶ 308](#)])

bB: TRUE => B-Zeit (Z.B.: **02:05:00 MEZ B**), FALSE => Übrige Zeit (Z.B.: **02:05:00 MESZ A**). Bei einem Sprung der Lokalzeit zurück wird dieser Ausgang gesetzt und beim Überschreiten der doppelten Lokalzeit wieder zurückgesetzt.

Beispiel:

Die UTC-Zeit: DT#2011-09-02-09:01:31 wird in Lokalzeit konvertiert. Das Ergebnis ist: DT#2011-09-02-11:01:31.

```
PROGRAM MAIN
VAR
  in      : DT := DT#2011-09-02-09:01:31; (* UTC time *)
```

```

    out      : DT;(* Local time *)
    fbToLocal : FB_FileTime64ToTzSpecificLocalTime;
END_VAR

fbToLocal( in := DT_TO_FILETIME64( in ), tzInfo := WEST_EUROPE_TZI );
out := FILETIME64_TO_DT( fbToLocal.out );

```

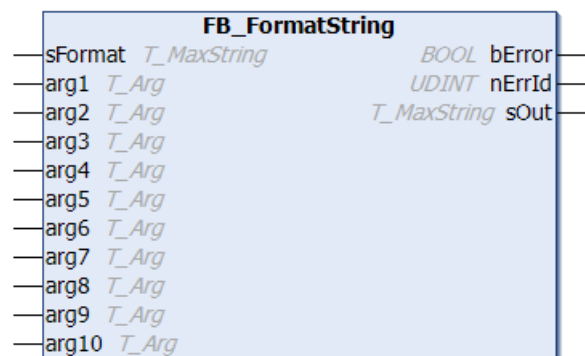
Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

- [FB_TzSpecificLocalTimeToSystemTime \[▶ 117\]](#)
- [FB_TzSpecificLocalTimeToFileTime64 \[▶ 115\]](#)
- [FB_SystemTimeToTzSpecificLocalTime \[▶ 113\]](#)
- [FB_GetTimeZoneInformation \[▶ 78\]](#)
- [FB_SetTimeZoneInformation \[▶ 111\]](#)
- [NT_SetLocalTime \[▶ 124\]](#)
- [NT_GetTime \[▶ 122\]](#)
- [NT_SetTimeToRTCTime \[▶ 125\]](#)
- [F_TranslateFileTime64Bias \[▶ 154\]](#)
- [FB_LocalSystemTime \[▶ 90\]](#)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.44.0

3.22 FB_FormatString



Mit dem Funktionsbaustein können bis zu 10 Argumente (ähnlich wie bei fprintf) entsprechend der [Formatspezifikation \[▶ 356\]](#) in einen String konvertiert und formatiert werden. Die Formatierung wird im gleichen SPS-Zyklus durchgeführt. D.h. der Ausgabestring ist sofort nach dem Aufruf des FBs verfügbar.

VAR_INPUT

```

VAR_INPUT
    sFormat : T_MaxString;
    arg1    : T_Arg;
    arg2    : T_Arg;
    arg3    : T_Arg;
    arg4    : T_Arg;
    arg5    : T_Arg;
    arg6    : T_Arg;
    arg7    : T_Arg;
    arg8    : T_Arg;
    arg9    : T_Arg;
    arg10   : T_Arg;
END_VAR

```

sFormat: Formatspezifikation als String (Typ: T_MaxString)(z.B. '%+20.5f' oder 'Measure X: %+ .10d, Y: %+ .10d').

arg1 bis arg10: Argumente, die formatiert werden sollen (Typ: T_Arg [▶ 323]). Folgende Hilfsfunktionen können benutzt werden um SPS-Variablen unterschiedlichsten Typs in den benötigten Datentyp T_Arg [▶ 323] zu konvertieren: F_BYTE [▶ 233], F_WORD [▶ 240], F_DWORD [▶ 234], F_LWORD [▶ 236], F_SINT [▶ 237], F_INT [▶ 235], F_DINT [▶ 233], F_LINT [▶ 235], F_USINT [▶ 240], F_UINT [▶ 239], F_UDINT [▶ 238], F_ULINT [▶ 240], F_STRING [▶ 237], F_REAL [▶ 237], F_LREAL [▶ 236].

VAR_OUTPUT

```
VAR_OUTPUT
  bError : BOOL;
  nErrId : UDINT;
  sOut   : T_MaxString;
END_VAR
```

bError: Sollte ein Fehler bei der Formatierung aufgetreten sein, dann wird dieser Ausgang gesetzt.

nErrId: Liefert bei einem gesetzten bError-Ausgang den Format Fehlercode. [▶ 359]

sOut: Bei Erfolg liefert dieser Ausgang den formatierten Ausgabestring (Typ: T_MaxString)

Beispiel:

```
PROGRAM MAIN
VAR
  fbFormat   : FB_FormatString;
  iY         : DINT;
  iX         : DINT;
  bError     : BOOL;
  nErrID     : UDINT;
  sOut       : T_MaxString;
END_VAR

iX := iX + 1;
iY := iY + 1;
fbFormat( sFormat := 'Measure X: %+10d, Y: %
+10d', arg1 := F_DINT( iX ), arg2 := F_DINT( iY ), sOut => sOut, bError => bError, nErrID => nErrID
);
```

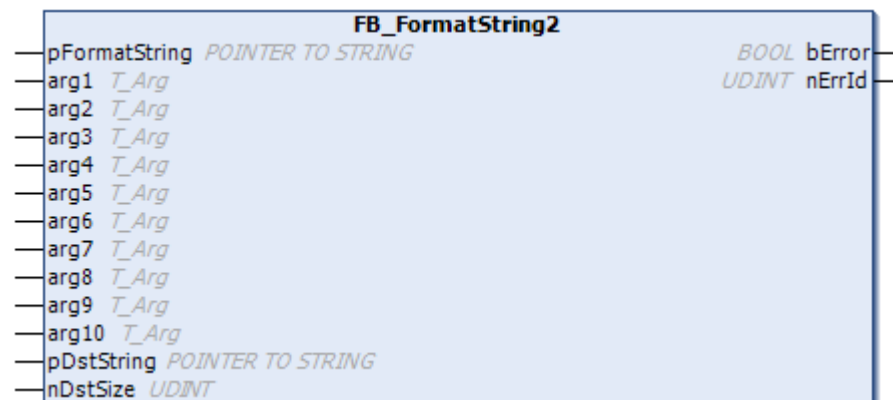
Das Ergebnis:

sOut = 'Measure X: +0000000130, Y: +0000000130'

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.23 FB_FormatString2



Mit dem Funktionsbaustein können bis zu 10 Argumente (ähnlich wie bei printf) entsprechend der [Formatspezifikation](#) [[▶ 356](#)] (z. B. '%+20.5f' oder 'Measure X: %+10d, Y: %+10d') in einen String konvertiert und formatiert werden. Die Formatierung wird im gleichen SPS-Zyklus durchgeführt. D. h. der Ausgabestring ist sofort nach dem Aufruf des Funktionsbausteins verfügbar.

Die Größen von Format-String und Ausgabestring sind, entgegen dem Funktionsbaustein [FB_FormatString](#) [[▶ 64](#)], nicht auf 255 Zeichen begrenzt. Jedes Argument ist jedoch limitiert auf eine Darstellung von maximal 250 Zeichen. Wenn die Zeichenanzahl überschritten wird oder der Ausgabestring zu klein sein für die formatierte Zeichenkette, so gibt der Funktionsbaustein einen Fehler aus.

VAR_INPUT

```
VAR_INPUT
  pFormatString : POINTER TO STRING;
  arg1          : T_Arg;
  arg2          : T_Arg;
  arg3          : T_Arg;
  arg4          : T_Arg;
  arg5          : T_Arg;
  arg6          : T_Arg;
  arg7          : T_Arg;
  arg8          : T_Arg;
  arg9          : T_Arg;
  arg10         : T_Arg;
  pDstString    : POINTER TO STRING;
  nDstSize      : UDINT;
END_VAR
```

pFormatString: Zeiger auf die Formatspezifikation als String. Die Adresse muss bei jedem Aufruf des Funktionsbausteins zugewiesen werden. Bei der Zuweisung kann der Operator ADR() verwendet werden.

arg1 bis arg10: Argumente, die formatiert werden sollen (Typ: [T_Arg](#) [[▶ 323](#)]). Folgende Hilfsfunktionen können benutzt werden, um SPS-Variablen unterschiedlichsten Typs in den benötigten Datentyp [T_Arg](#) [[▶ 323](#)] zu konvertieren: [F_BYTE](#) [[▶ 233](#)], [F_WORD](#) [[▶ 240](#)], [F_DWORD](#) [[▶ 234](#)], [F_LWORD](#) [[▶ 236](#)], [F_SINT](#) [[▶ 237](#)], [F_INT](#) [[▶ 235](#)], [F_DINT](#) [[▶ 233](#)], [F_LINT](#) [[▶ 235](#)], [F_USINT](#) [[▶ 240](#)], [F_UINT](#) [[▶ 239](#)], [F_UDINT](#) [[▶ 238](#)], [F_ULINT](#) [[▶ 240](#)], [F_STRING](#) [[▶ 237](#)], [F_REAL](#) [[▶ 237](#)], [F_LREAL](#) [[▶ 236](#)].

pDstString: Zeiger auf die resultierende STRING-Variable. Bei Erfolg wird hier die formatierte Zeichenkette geschrieben. Die Adresse muss bei jedem Aufruf des Funktionsbausteins zugewiesen werden. Bei der Zuweisung kann der Operator ADR() verwendet werden.

nDstSize: Größe der resultierenden STRING-Variablen in Bytes. Bei der Zuweisung kann der Operator SIZEOF() verwendet werden.

VAR_OUTPUT

```
VAR_OUTPUT
  bError : BOOL;
  nErrId : UDINT;
END_VAR
```

bError: TRUE, wenn bei der Formatierung ein Fehler aufgetreten ist.

nErrId: Liefert bei einem gesetzten bError-Ausgang den [Format-Fehlercode](#). [[▶ 359](#)]

Beispiel

```
PROGRAM MAIN
VAR
  fbFormat    : FB_FormatString2;
  sFormat     : STRING := 'Measure X: %+10d, Y: %+10d';
  iY          : DINT;
  iX          : DINT;
  bError      : BOOL;
  nErrID      : UDINT;
  sOut        : STRING(600);
END_VAR

iX := iX + 1;
iY := iY + 1;
fbFormat( pFormatString := ADR(sFormat), arg1 := F_DINT( iX ), arg2 := F_DINT( iY ), pDstString :=
ADR(sOut), nDstSize := SIZEOF(sOut), bError => bError, nErrID => nErrID );
```

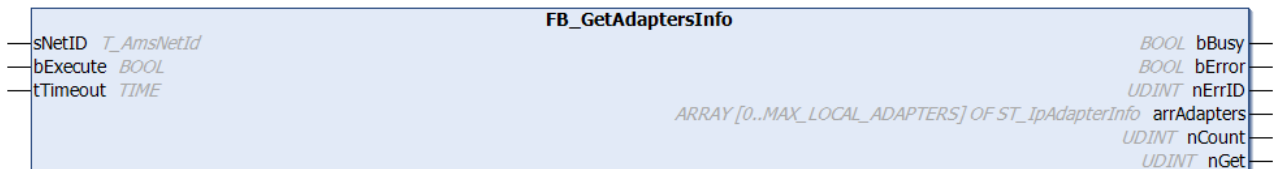
Ergebnis:

sOut = 'Measure X: +0000000130, Y: +0000000130'

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.35.0

3.24 FB_GetAdaptersInfo



Mit diesem Funktionsbaustein können Adapterinformationen eines TwinCAT PCs ausgelesen werden. Die maximale Anzahl der gelesenen Adapterinformationen ist derzeit auf MAX_LOCAL_ADAPTERS + 1 (Default = 6) begrenzt.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetID: Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Adapterinformationen gelesen werden sollen. Für den lokalen PC kann auch ein Leerstring angegeben werden (Typ: T_AmsNetID).

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrID      : UDINT;
  arrAdapters : ARRAY [0..MAX_LOCAL_ADAPTERS] OF ST_IpAdapterInfo;
  nCount      : UDINT;
  nGet        : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrID: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [► 360].

arrAdapters: Array-Variable mit zuletzt gelesenen Adapterinformationen. Jedes Arrayelement liefert Informationen eines Adapters (Typ: ST_IpAdapterInfo [► 317]).

nCount: Maximale Anzahl der gefundenen lokalen Adapter.

nGet: Anzahl der gültigen Einträge der *arrAdapters*-Ausgangsvariablen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64)	Tc2_Uilities (System)

3.25 FB_GetAdaptersInfoEx

Mit diesem Funktionsbaustein können Adapterinformationen eines TwinCAT PCs ausgelesen werden. Die maximale Anzahl der gelesenen Adapterinformationen ist auf 64 begrenzt.

Nach einem erfolgreichen Aufruf ist der Lesevorgang beendet und die gelesenen Adapterinformationen können mittels der Methode Get() kopiert werden.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetID: Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Adapterinformationen gelesen werden sollen. Für den lokalen PC kann auch ein Leerstring angegeben werden (Typ: T_AmsNetID).

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrorID    : UDINT;
  nAdapters   : UINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrorID: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [▶ 360].

nAdapters: Anzahl der gefundenen lokalen Adapter. Deren Adapterinformationen wurden ausgelesen und können mittels der Methode Get() kopiert werden.

Methode Get()

Nach erfolgreichem Aufruf des Funktionsbausteins ist der Lesevorgang beendet und die gelesenen Adapterinformationen können mittels der Methode Get() kopiert werden.

Es ist möglich die Informationen aller Adapter gemeinsam zu kopieren. Ebenso können die Informationen pro Adapter einzeln nacheinander kopiert werden.

Beim Aufruf wird ein lokales Array (Typ: ST_IpAdapterInfo [▶ 317]) angegeben. Jedes Array-Element liefert so Informationen eines Adapters.

```
METHOD Get : BOOL
VAR_INPUT
  pAdapters   : POINTER TO ST_IpAdapterInfo; // pointer to array of adapter info (variable ar
```

```
ray length)
  nAdaptersSize : UDINT; // size in bytes of array of adapter info
END_VAR
```

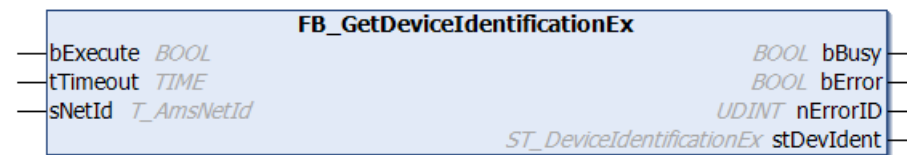
Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.48.0

Sehen Sie dazu auch

📄 ST_IPAdapterInfo [▶ 317]

3.26 FB_GetDeviceIdentificationEx



Der Baustein liest die Geräteidentifikation ein. Es werden längere Strings bei Hardware Model und Hardware Seriennummer erlaubt als beim [FB_GetDeviceIdentification](#) [▶ 26].

VAR_INPUT

```
VAR_INPUT
  bExecute : BOOL;
  tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
  sNetId   : T_AmsNetId;
END_VAR
```

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

sNetId: Hier kann die AmsNetId des TwinCAT-Rechners angegeben werden, dessen Geräteidentifikation gelesen werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorId   : UDINT;
  stDevIdent : ST_DeviceIdentificationEx;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

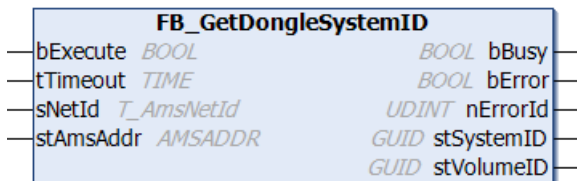
nErrorId: Liefert bei einem gesetzten bError-Ausgang die [ADS-Fehlernummer](#) [▶ 360].

stDevIdent: Liefert die Geräteidentifikation (Typ: [ST_DeviceIdentificationEx](#) [▶ 313]).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.27 FB_GetDongleSystemID



Der Funktionsbaustein FB_GetDongleSystemID liest die System-ID und die Volume-ID des TwinCAT-3-Lizenz-Dongles als GUID aus.

VAR_INPUT

```
VAR_INPUT
    bExecute   : BOOL;
    tTimeout   : TIME;
    sNetId     : T_AmsNetId;
    stAmsAddr  : AMSADDR;
END_VAR
```

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Timeout-Zeit, die bei Ausführung des Befehls nicht überschritten werden darf.

sNetId: AmsNetId (AMS-Netzwerkennung) des TwinCAT-Rechners, dessen Lizenzstatus ausgelesen werden soll (Typ: T_AmsNetId). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

stAmsAddr: Netzwerkadresse (AmsNetId und Port) des Lizenz-Dongles (Typ: AMSADDR)

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrorId   : UDINT;
    stSystemID : GUID;
    stVolumeID : GUID;
END_VAR
```

bBusy: TRUE, solange der Baustein aktiv ist.

bError: TRUE, wenn bei der Ausführung des Kommandos ein Fehler auftritt.

nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

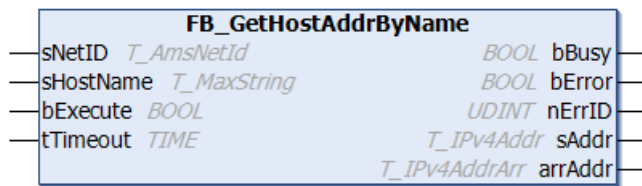
stSystemID: System-ID des Lizenz-Dongles (Typ: GUID)

stVolumeID: Volume-ID des Lizenz-Dongles (Typ: GUID)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022	PC oder CX (x64, x86)	Tc2_Uilities >= 3.3.24.0

3.28 FB_GetHostAddrByName



Mit diesem Funktionsbaustein kann die (IPv4) Internet Protokoll Netzwerkadresse für den angegebenen Hostnamen ausgelesen werden. Die Adresse wird als String und Byte-Array zurückgeliefert.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetId;
  sHostName   : T_MaxString := '';
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetID: Hier kann die Netzwerkadresse des TwinCAT Rechners angegeben werden, auf dem das Kommando ausgeführt werden soll (Typ: T_AmsNetID). Für den lokalen PC (default) kann auch ein Leerstring angegeben werden.

sHostName: Host name als String (Typ: T_MaxString). Z.B.: 'DataServer1'.

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrID      : UDINT;
  sAddr       : T_IPv4Addr := '';
  arrAddr     : T_IPv4AddrArr := [ 0, 0, 0, 0 ];
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrID: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [► 360].

sAddr: Internet Protokoll Netzwerkadresse (IPv4) als string (Typ: T_Ipv4Addr). Z.B.: '172.16.7.199'

arrAddr: Internet Protokoll Netzwerkadresse als byte array (Typ: T_Ipv4AddrArr).

Beispiel:

```
PROGRAM MAIN
VAR
  fbGet      : FB_GetHostAddrByName;
  bGet       : BOOL := TRUE;
  bError     : BOOL;
  nErrID     : UDINT;
  sIPv4      : T_IPv4Addr; (* Result: '87.106.8.100' *)
  arrIPv4    : T_IPv4AddrArr;
  state      : BYTE;
END_VAR

CASE state OF
0:
  IF bGet THEN
    bGet := FALSE;
    sIPv4 := '';
    fbGet( bExecute:= FALSE );
```

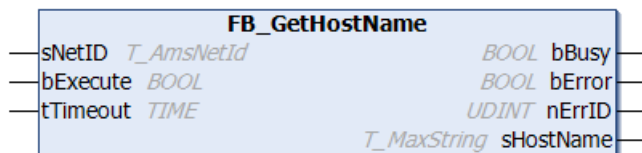
```

        fbGet( bExecute:= TRUE, sHostName := 'www.beckhoff.com' );
        state := 1;
    END_IF
1:
    fbGet( bExecute:= FALSE, bError=>bError, nErrID=>nErrID, sAddr=>sIPv4, arrAddr=>arrIPv4 );
    IF NOT fbGet.bBusy THEN
        state := 0;
    END_IF
END_CASE
    
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Uilities (System)

3.29 FB_GetHostName



Mit diesem Funktionsbaustein kann der Hostname eines TwinCAT PCs ausgelesen werden.

VAR_INPUT

```

VAR_INPUT
    sNetID    : T_AmsNetId;
    bExecute  : BOOL;
    tTimeout  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
    
```

sNetID: Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Hostname gelesen werden soll (Typ: T_AmsNetID). Für den lokalen PC kann auch ein Leerstring angegeben werden.

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```

VAR_OUTPUT
    bBusy     : BOOL;
    bError    : BOOL;
    nErrID    : UDINT;
    sHostName : T_MaxString;
END_VAR
    
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrID: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [► 360].

sHostName: Hostname als String (Typ: T_MaxString).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Uilities (System)

3.30 FB_GetLicenseDongles



Der Funktionsbaustein ermittelt die Anzahl angeschlossener Lizenz-Dongles und liefert Adresse und Status.

VAR_INPUT

```
VAR_INPUT
    bExecute : BOOL;
    tTimeout : TIME;
    sNetId   : T_AmsNetId;
END_VAR
```

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Timeout-Zeit, die bei Ausführung des Befehls nicht überschritten werden darf.

sNetId: AmsNetId (AMS-Netzwerkennung) des TwinCAT-Rechners, dessen Lizenzstatus ausgelesen werden soll (Typ: T_AmsNetId). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrorId   : UDINT;
    nLicenseDeviceDongles : UDINT;
    aLicenseDeviceDongles : ARRAY[1..nMaxLicenseDevices] OF ST_LicenseDongle;
END_VAR
```

bBusy: TRUE, solange der Baustein aktiv ist.

bError: TRUE, wenn bei der Ausführung des Kommandos ein Fehler auftritt.

nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

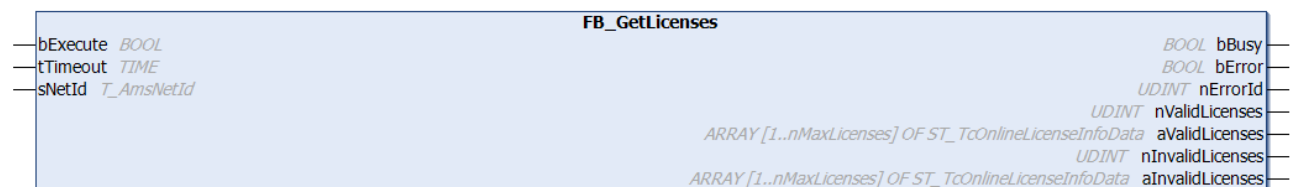
nLicenseDeviceDongles: Anzahl der Lizenz-Dongles

aLicenseDeviceDongles: Identifikationsdaten der angeschlossenen Lizenz-Dongle

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022	PC oder CX (x64, x86)	Tc2_Uilities >= 3.3.24.0

3.31 FB_GetLicenses



Der Funktionsbaustein liest die gültigen und ungültigen TwinCAT-Lizenzen aus.

VAR_INPUT

```
VAR_INPUT
    bExecute : BOOL;
    tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
    sNetId : T_AmsNetId;
END_VAR
```

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

sNetId: Hier kann die AmsNetId (Typ: T_AmsNetID) des TwinCAT-Rechners angegeben werden, dessen aktuelle TwinCAT-Lizenzen ausgelesen werden sollen. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy : BOOL;
    bError : BOOL;
    nErrorId : UDINT;
    nValidLicenses : UDINT;
    aValidLicenses : ARRAY [1..nMaxLicenses] OF ST_TcOnlineLicenseInfoData;
    nInvalidLicenses : UDINT;
    aInvalidLicenses : ARRAY [1..nMaxLicenses] OF ST_TcOnlineLicenseInfoData;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer [▶ 360].

nValidLicenses: Liefert die Anzahl der gültigen TwinCAT-Lizenzen.

aValidLicenses: Liefert eine Liste der gültigen TwinCAT-Lizenzen vom Datentyp ST_TcOnlineLicenseInfoData [▶ 320].

nInvalidLicenses: Liefert die Anzahl der ungültigen TwinCAT-Lizenzen.

aInvalidLicenses: Liefert eine Liste der ungültigen TwinCAT-Lizenzen vom Datentyp ST_TcOnlineLicenseInfoData [▶ 320].



Die Lizenzlisten beinhalten standardmäßig maximal 50 Einträge. Diese Grenze kann in der Parameterliste der Bibliothek über nMaxLicenses angepasst werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategorie-gruppe)
TwinCAT v3.1.4018	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) v3.3.9.0 oder höher

3.32 FB_GetLicensesEx

FB_GetLicensesEx	
bExecute <i>BOOL</i>	<i>BOOL</i> bBusy
tTimeout <i>TIME</i>	<i>BOOL</i> bError
sNetId <i>T_AmsNetId</i>	<i>UDINT</i> nErrorId
	<i>UDINT</i> nValidLicenses
	<i>ARRAY [1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx</i> aValidLicenses
	<i>UDINT</i> nPendingLicenses
	<i>ARRAY [1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx</i> aPendingLicenses
	<i>UDINT</i> nDemoLicenses
	<i>ARRAY [1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx</i> aDemoLicenses
	<i>UDINT</i> nOemLicenses
	<i>ARRAY [1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx</i> aOemLicenses
	<i>UDINT</i> nFailedLicenses
	<i>ARRAY [1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx</i> aFailedLicenses
	<i>UDINT</i> nInvalidLicenses
	<i>ARRAY [1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx</i> aInvalidLicenses

Der Funktionsbaustein FB_GetLicensesEx ermittelt den Status aller TwinCAT-3-Lizenzen und OEM-Lizenzen.

VAR_INPUT

```
VAR_INPUT
    bExecute : BOOL;
    tTimeout : TIME;
    sNetId   : T_AmsNetId;
END_VAR
```

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Timeout-Zeit, die bei Ausführung des Befehls nicht überschritten werden darf.

sNetId: AmsNetId (AMS-Netzwerkennung) des TwinCAT-Rechners, dessen Lizenzstatus ausgelesen werden soll (Typ: T_AmsNetId). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    nErrorId    : UDINT;
    nValidLicenses : UDINT
    aValidLicenses : ARRAY[1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx
    nPendingLicenses : UDINT
    aPendingLicenses : ARRAY[1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx
    nDemoLicenses : UDINT
    aDemoLicenses : ARRAY[1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx
    nOemLicenses : UDINT
    aOemLicenses : ARRAY[1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx
    nFailedLicenses : UDINT
    aFailedLicenses : ARRAY[1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx
    nInvalidLicenses : UDINT
    aInvalidLicenses : ARRAY[1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx
END_VAR
```

bBusy: TRUE, solange der Baustein aktiv ist.

bError: TRUE, wenn bei der Ausführung des Kommandos ein Fehler auftritt.

nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

nValidLicenses: Anzahl gültiger Lizenzen

aValidLicenses: Informationen zu den gültigen Lizenzen

nPendingLicenses: Anzahl noch offener Lizenzen

aPendingLicenses: Informationen zu den noch offenen Lizenzen

nDemoLicenses: Anzahl gültiger Demolizenzen

aDemoLicenses: Informationen zu den gültigen Demolizenzen

nOemLicenses: Anzahl gültiger OEM-Lizenzen

aOemLicenses: Informationen zu den gültigen OEM-Lizenzen

nFailedLicenses: Anzahl fehlgeschlagener Lizenzen

aFailedLicenses: Informationen zu den gültigen Lizenzen

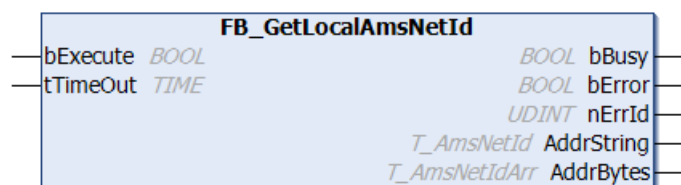
nInvalidLicenses: Anzahl ungültiger Lizenzen

aInvalidLicenses: Informationen zu den ungültigen Lizenzen

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022	PC oder CX (x64, x86)	Tc2_Utilities >= 3.3.24.0

3.33 FB_GetLocalAmsNetId



Mit dem Funktionsbaustein kann die Netzwerkadresse (AmsNetId) des lokalen TwinCAT PCs ausgelesen werden.

VAR_INPUT

```
VAR_INPUT
    bExecute   :BOOL;
    tTimeout   :TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy      :BOOL;
    bError     :BOOL;
    nErrId     :UDINT;
    AddrString :T_AmsNetId;
    AddrBytes  :T_AmsNetIdArr;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [► 360].

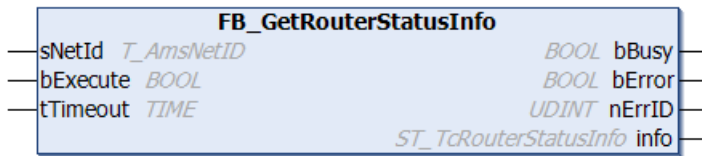
AddrString: Die AmsNetId des lokalen PCs als String (Typ: T_AmsNetID).

AddrBytes: Die AmsNetId des lokalen PCs als Byte-Array (Typ: T_AmsNetIDArr).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.34 FB_GetRouterStatusInfo



Mit dem Funktionsbaustein FB_GetRouterStatusInfo können Statusinformationen des TwinCAT Routers aus der SPS ausgelesen werden (verfügbarer Speicher, Anzahl der angemeldeten Ports usw.).

VAR_INPUT

```
VAR_INPUT
    sNetId    : T_AmsNetID := '';
    bExecute  : BOOL;
    tTimeout  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen TwinCAT Router-Informationen ausgelesen werden sollen (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeOut: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy     : BOOL;
    bError    : BOOL;
    nErrId    : UDINT;
    info      : ST_TcRouterStatusInfo;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

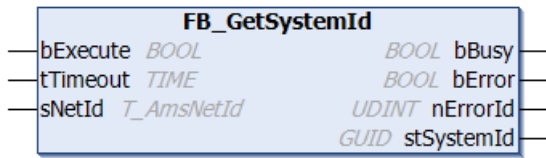
nErrId: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [▶ 360].

info: Strukturvariable mit TwinCAT Router-Statusinformationen (Typ: ST_TcRouterStatusInfo [▶ 321]).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.35 FB_GetSystemId



Der Funktionsbaustein liest die System-ID vom Typ GUID (siehe "Über TwinCAT..." beim TwinCAT Icon in dem System Tray).

VAR_INPUT

```
VAR_INPUT
    bExecute : BOOL;
    tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
    sNetId   : T_AmsNetId;
END_VAR
```

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

sNetId: Hier kann die AmsNetId des TwinCAT-Rechners angegeben werden, dessen System ID gelesen werden sollen (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy      : BOOL;
    bError     : BOOL;
    nErrorId   : UDINT;
    stSystemId : GUID;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

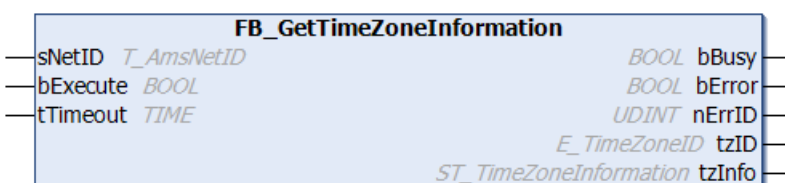
nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer [► 360].

stSystemId: Liefert die System-ID als GUID [► 309].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.36 FB_GetTimeZoneInformation



Mit dem Funktionsbaustein können die Zeitzone-Einstellungen des Betriebssystems ausgelesen werden.

VAR_INPUT

```
VAR_INPUT
  sNetID : T_AmsNetID;
  bExecute : BOOL;
  tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Zeitzone-Einstellungen ausgelesen werden sollen (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy : BOOL;
  bError : BOOL;
  nErrId : UDINT;
  tzID : E_TimeZoneID;
  tzInfo : ST_TimeZoneInformation;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [▶ 360].

tzID: Zusätzliche Sommer-/ Winterzeit-Informationen (nicht immer vorhanden) (Typ: E_TimeZoneID [▶ 308]).

tzInfo: Bei Erfolg liefert diese Strukturvariable die aktuellen Zeitzone-Informationen des Betriebssystems (Typ: ST_TimeZoneInformation [▶ 321]).

Beispiel:

Siehe in der Beschreibung des FB_SetTimeZoneInformation [▶ 111]-Funktionsbausteins.

Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

- [FB_TzSpecificLocalTimeToSystemTime](#) [▶ 117]
- [FB_TzSpecificLocalTimeToFileTime](#) [▶ 29]
- [FB_SystemTimeToTzSpecificLocalTime](#) [▶ 113]
- [FB_FileTimeTimeToTzSpecificLocalTime](#) [▶ 27]
- [FB_SetTimeZoneInformation](#) [▶ 111]
- [NT_SetLocalTime](#) [▶ 124]
- [NT_GetTime](#) [▶ 122]
- [NT_SetTimeToRTCTime](#) [▶ 125]
- [F_TranslateFileTimeBias](#) [▶ 243]
- [FB_LocalSystemTime](#) [▶ 90]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Uilities (System)

3.37 FB_GetVolumeld



Der Funktionsbaustein FB_GetVolumeld liest die System-ID und die Volume-System-ID als GUID aus.

VAR_INPUT

```
VAR_INPUT
    bExecute      :   BOOL;
    tTimeout      :   TIME;
    sNetId        :   T_AmsNetId;
END_VAR
```

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

sNetId: Hier kann die AmsNetId des TwinCAT-Rechners angegeben werden, dessen System-ID ausgelesen werden soll (Typ: T_AmsNetId). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy         :   BOOL;
    bError        :   BOOL;
    nErrorId      :   UDINT;
    stVolumeId    :   GUID;
    stSystemId    :   GUID;
END_VAR
```

bBusy: Dieser Ausgang wird bei der Aktivierung des Funktionsbaustein gesetzt und bleibt gesetzt bis eine Rückmeldung erfolgt.

bError: Wenn bei der Übertragung des Kommandos ein Fehler auftritt, wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer [► 360].

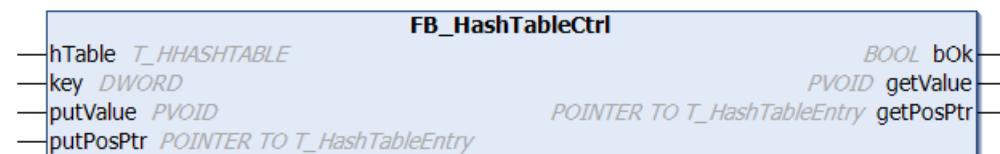
stVolumeld: Liefert die Volume-System-ID als GUID [► 309].

stSystemId: Liefert die System-ID als GUID [► 309].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4018	PC oder CX (x86, x64, ARM)	Tc2_Utilities

3.38 FB_HashTableCtrl



Hashtable kann dazu verwendet werden um ein einzelnes Datenelement in einer größeren Menge von Datenelementen schnell auffinden zu können. Die Datenobjekte müssen mit einem eindeutigen Schlüssel versehen werden. Über diesen Schlüssel können die Datenobjekte dann in der Tabelle eindeutig identifiziert und schnell gefunden werden.

Mit dem Funktionsbaustein FB_HashTableCtrl kann eine einfache Hash-Tabelle im SPS-Projekt realisiert werden. Es wird dabei das Verfahren: Hashing mit Verkettung (Separate-Chaining) angewandt.

Die maximale Anzahl der Datenelemente kann zur Laufzeit nicht verändert werden und muss vorher festgelegt werden. Das Hinzufügen/Entfernen/Suchen der Datenelemente wird durch Aktionsaufrufe gesteuert. Der Funktionsbaustein besitzt folgende Aktionen:

- **A_Add** (Fügt ein neues Datenelement der Tabelle hinzu (Schlüssel/Wert). Ein bereits existierendes Element mit demselben Schlüssel wird überschrieben!)
- **A_GetFirst** (Liest das erste Tabellen-Datenelement. Bei Erfolg liefert *getValue* den dazugehörigen Wert.)
- **A_GetNext** (Liest das nächste Tabellen-Datenelement. Die Adresse: *putPosPtr* muss auf das vorherige Datenelement zeigen!)
- **A_Lookup** (Sucht ein zum Schlüssel (key) passendes Datenelement. Bei Erfolg liefert *getValue* den dazugehörigen Wert.)
- **A_Remove** (Entfernt ein zum Schlüssel passendes Datenelement.)
- **A_RemoveAll** (Entfernt alle Datenelemente)
- **A_RemoveFirst** (Entfernt das erste Datenelement)
- **A_Reset** (Löscht alle Datenelemente und setzt die Tabelle zurück.)
- **A_GetIndexAtPosPtr** (Liefert den Array-Index des Datenelements an der Adresse: *putPosPtr*. Bei Erfolg liefert *getValue* den Null-Basierten Array-Index. Der Wert *putValue* wird nicht benutzt. Bitte beachten Sie, dass der Wert *getValue* einen Datenelement-Index und nicht den Datenelement-Wert zurückliefert!)

VAR_IN_OUT

```
VAR_IN_OUT
  hTable : T_HHASHTABLE;
END_VAR
```

hTable: Hash-Tabellen-Handle (Typ: [T_HHASHTABLE \[► 325\]](#)). Das Handle muss einmalig vor der Benutzung mit der Funktion: [F_CreateHashTableHnd \[► 264\]](#) initialisiert werden. Für jede Tabelle muss eine dazugehörige Instanz der Handle-Variablen angelegt und initialisiert werden.

VAR_INPUT

```
VAR_INPUT
  key      : DWORD := 0;
  putValue : PVOID := 0;
  putPosPtr : POINTER TO T_HashTableEntry := 0;
END_VAR
```

key: Schlüssel (vorzeichenlose 32 Bit Zahl oder auch Pointer). Über diesen Schlüssel kann ein Datenelement in der Tabelle schnell identifiziert/gefunden werden.

putValue: Wert/Datenelement (Eingangsparameter, 32/64 Bit, vorzeichenlose Zahl oder auch Pointer).

putPosPtr: Adresse auf das Datenelement (Eingangsparameter, Typ: [T_HashTableEntry \[► 325\]](#))

VAR_OUTPUT

```
VAR_OUTPUT
  bOk      : BOOL := FALSE;
  getValue : PVOID := 0;
  getPosPtr : POINTER TO T_HashTableEntry := 0;
END_VAR
```

bOk: Liefert TRUE wenn ein neues Datenelement der Tabelle hinzugefügt/entfernt oder in der Tabelle gefunden werden konnte. FALSE wird geliefert wenn das gesuchte Datenelement nicht gefunden werden konnte, die Tabelle leer ist oder einen Überlauf hat (keine freien Datenelemente mehr).

getValue: Der zum Schlüssel passende Wert/Datenelement (Ausgangsparameter, 32/64 Bit, vorzeichenlose Zahl oder auch Pointer).

getPosPtr: Die Adresse auf das Datenelement (Ausgangsparameter, Typ: [T_HashTableEntry](#) [► 325]).

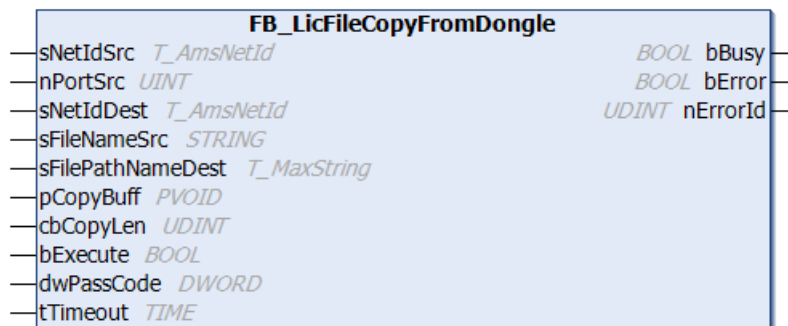
Beispiel:

Siehe: [Beispiel: Hash-Tabelle \(FB_HashTableCtrl\)](#). [► 341]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.39 FB_LicFileCopyFromDongle



Der Funktionsbaustein kopiert eine Datei vom Lizenz-Dongle auf die Festplatte. Ist die Datei größer als der Buffer (cbCopyLen), dann wird die Datei automatisch in mehreren Lese- und Schreibzugriffen kopiert, bis die komplette Datei kopiert ist, erst dann geht das bBusy auf FALSE.

VAR_INPUT

```

VAR_INPUT
  sNetIdSrc      : T_AmsNetId;
  nPortSrc       : UINT;
  sNetIdDest     : T_AmsNetId;
  sFileNameSrc   : STRING;
  sFilePathNameDest : T_MaxString;
  pCopyBuff      : PVOID;
  cbCopyLen      : UDINT;
  bExecute       : BOOL;
  dwPassCode     : DWORD;
  tTimeout       : TIME      := DEFAULT_ADS_TIMEOUT;
END_VAR
  
```

sNetIdSrc: AmsNetId (AMS-Netzwerkennung) des Lizenz-Dongles (Typ: T_AmsNetId)

- USB-Dongle: AmsNetId des TwinCAT-Rechners. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
- EL6070: AmsNetId des EtherCAT-Masters (siehe AdsAddr.netId in den InfoData der EL6070)

nPortSrc: AMS-Port des Lizenz-Dongles

- USB: ADS-Port des ESB-Geräts (siehe ADS Port auf dem ESB Device Tab vom USB-Dongle, Default ist 16#7100)
- EL6070: ADS-Port des EtherCAT-Terminals (siehe AdsAddr.port in den InfoData der EL6070)

sNetIdDest: AmsNetId (AMS-Netzwerkennung) des TwinCAT-Rechners. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

sFileNameSrc: Name der Datei auf dem Lizenz-Dongle

sFilePathNameSrc: Pfadname der Datei auf der Festplatte. (Typ: T_MaxString)

pCopyBuff: Buffer address for write

cbCopyLen: Count of bytes for write

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

dwPassCode: Passcode für den Dateizugriff

tTimeout: Timeout-Zeit, die bei Ausführung des Befehls nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorId   : UDINT;
END_VAR
```

bBusy: TRUE, solange der Baustein aktiv ist.

bError: TRUE, wenn bei der Ausführung des Kommandos ein Fehler auftritt.

nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022	PC oder CX (x64, x86)	Tc2_Utilities >= 3.3.26.0

3.40 FB_LicFileCopyToDongle



Der Funktionsbaustein kopiert eine Datei von der Festplatte auf den Lizenz-Dongle. Ist die Datei größer als der Buffer (cbCopyLen), dann wird die Datei automatisch in mehreren Lese- und Schreibzugriffen kopiert, bis die komplette Datei kopiert ist, erst dann geht das bBusy auf FALSE.

VAR_INPUT

```
VAR_INPUT
  sNetIdSrc      : T_AmsNetId;
  sNetIdDest     : T_AmsNetId;
  nPortDest      : UINT;
  sFilePathNameSrc : T_MaxString;
  sFileNameDest  : STRING;
  pCopyBuff      : PVOID;
  cbCopyLen      : UDINT;
  bExecute       : BOOL;
  dwPassCode     : DWORD;
  tTimeout       : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetIdSrc: AmsNetId (AMS-Netzwerkennung) des TwinCAT-Rechners. Für den lokalen Rechner kann auch ein Leerstring angegeben werden. (Typ: T_AmsNetId)

sNetIdDest: AmsNetId (AMS-Netzwerkennung) des Lizenz-Dongles (Typ: T_AmsNetId)

- USB-Dongle: AmsNetId des TwinCAT-Rechners. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

- EL6070: AmsNetId des EtherCAT-Masters (siehe AdsAddr.netId in den InfoData der EL6070)

nPortDest: AMS-Port des Lizenz-Dongles

- USB: ADS-Port des ESB-Geräts (siehe ADS-Port auf dem ESB Device Tab vom USB-Dongle, Default ist 16#7100)
- EL6070: ADS-Port des EtherCAT-Terminals (siehe AdsAddr.port in den InfoData der EL6070)

sFilePathNameScr: Pfadname der Datei auf der Festplatte (Typ: T_MaxString)

sFileNameDest: Name der Datei auf dem Lizenz-Dongle

pCopyBuff: Buffer address for write

cbCopyLen: Count of bytes for write

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

dwPassCode: Passcode für den Dateizugriff

tTimeout: Timeout-Zeit, die bei Ausführung des Befehls nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorId   : UDINT;
END_VAR
```

bBusy: TRUE, solange der Baustein aktiv ist.

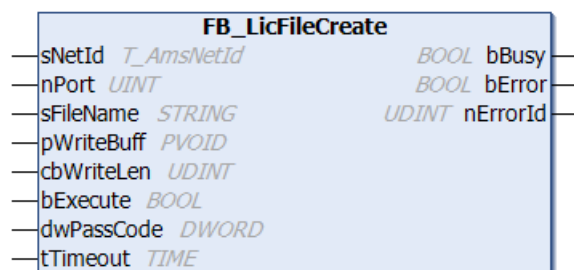
bError: TRUE, wenn bei der Ausführung des Kommandos ein Fehler auftritt.

nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022	PC oder CX (x64, x86)	Tc2_Uilities >= 3.3.26.0

3.41 FB_LicFileCreate



Der Funktionsbaustein erzeugt eine Datei auf dem Lizenz-Dongle. Die Daten aus dem Buffer (pWriteBuff und cbWriteLen) werden bei steigender Flanke an bExecute direkt in eine neue Datei auf dem Dongle geschrieben.

VAR_INPUT

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  nPort       : UINT;
  sFileName   : STRING;
  pWriteBuff  : PVOID;
  cbWriteLen  : UDINT;
  bExecute    : BOOL;
  dwPassCode  : DWORD;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: AmsNetId (AMS-Netzwerkennung) des Lizenz-Dongles (Typ: T_AmsNetId)

- USB-Dongle: AmsNetId des TwinCAT-Rechners. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
- EL6070: AmsNetId des EtherCAT-Masters (siehe AdsAddr.netId in den InfoData der EL6070)

nPort: AMS-Port des Lizenz-Dongles

- USB: ADS-Port des ESB-Geräts (siehe ADS-Port auf dem ESB Device Tab vom USB-Dongle, Default ist 16#7100)
- EL6070: ADS-Port des EtherCAT-Terminals (siehe AdsAddr.port in den InfoData der EL6070)

sFileName: Name der zu erzeugenden Datei

pWriteBuff: Buffer address for write

cbWriteLen: Count of bytes for write

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

dwPassCode: Passcode für den Dateizugriff

tTimeout: Timeout-Zeit, die bei Ausführung des Befehls nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorId   : UDINT;
END_VAR
```

bBusy: TRUE, solange der Baustein aktiv ist.

bError: TRUE, wenn bei der Ausführung des Kommandos ein Fehler auftritt.

nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022	PC oder CX (x64, x86)	Tc2_Utilities >= 3.3.26.0

3.42 FB_LicFileDelete



Der Funktionsbaustein löscht eine Datei auf dem Lizenz-Dongle. Hierfür werden der Dateiname und die Dateilänge genutzt, die Datenbytes der zu löschenden Datei werden auf dem Dongle zwar freigegeben aber nicht überschrieben.

VAR_INPUT

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  nPort       : UINT;
  sFileName   : STRING;
  bExecute    : BOOL;
  dwPassCode  : DWORD;
  tTimeout    : TIME      := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: AmsNetId (AMS-Netzwerkennung) des Lizenz-Dongles (Typ: T_AmsNetId)

- USB-Dongle: AmsNetId des TwinCAT-Rechners. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
- EL6070: AmsNetId des EtherCAT-Masters (siehe AdsAddr.netId in den InfoData der EL6070)

nPort: AMS-Port des Lizenz-Dongles

- USB: ADS-Port des ESB-Geräts (siehe ADS-Port auf dem ESB Device Tab vom USB-Dongle, Default ist 16#7100)
- EL6070: ADS-Port des EtherCAT-Terminals (siehe AdsAddr.port in den InfoData der EL6070)

sFileName: Name der zu löschenden Datei

pWriteBuff: Buffer address for write

cbWriteLen: Count of bytes for write

dwPassCode: Passcode für den Dateizugriff

tTimeout: Timeout-Zeit, die bei Ausführung des Befehls nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorId   : UDINT;
END_VAR
```

bBusy: TRUE, solange der Baustein aktiv ist.

bError: TRUE, wenn bei der Ausführung des Kommandos ein Fehler auftritt.

nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022	PC oder CX (x64, x86)	Tc2_Utilities >= 3.3.26.0

3.43 FB_LicFileGetStorageInfo



Der Funktionsbaustein liest die StorageInfo des Lizenz-Dongles und das Dateiverzeichnis aus.

Die StorageInfo enthält Verwaltungsdaten des Datenträgers (wie Kapazität, Anzahl der freien Bytes, Anzahl der Dateien, ...) und ein Array auf die einzelnen Dateieinträge (Name, Größe, Attribute, ... der Datei).

VAR_INPUT

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  nPort       : UDINT;
  bExecute    : BOOL;
  dwPassCode  : DWORD;
  tTimeout    : TIME      := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: AmsNetId (AMS-Netzwerkennung) des Lizenz-Dongles (Typ: T_AmsNetId)

- USB-Dongle: AmsNetId des TwinCAT-Rechners. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
- EL6070: AmsNetId des EtherCAT-Masters (siehe AdsAddr.netId in den InfoData der EL6070)

nPort: AMS-Port des Lizenz-Dongles

- USB: ADS-Port des ESB-Geräts (siehe ADS-Port auf dem ESB Device Tab vom USB-Dongle, Default ist 16#7100)
- EL6070: ADS-Port des EtherCAT-Terminals (siehe AdsAddr.port in den InfoData der EL6070)

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

dwPassCode: Passcode für den Dateizugriff (nur für speziell geschützte Dateien)

tTimeout: Timeout-Zeit, die bei Ausführung des Befehls nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorId   : UDINT;
  nFileEntries : UDINT;
END_VAR
```

bBusy: TRUE, solange der Baustein aktiv ist.

bError: TRUE, wenn bei der Ausführung des Kommandos ein Fehler auftritt.

nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

nFileEntries: Anzahl der Dateien auf dem Lizenz-Dongle

VAR_IN_OUT

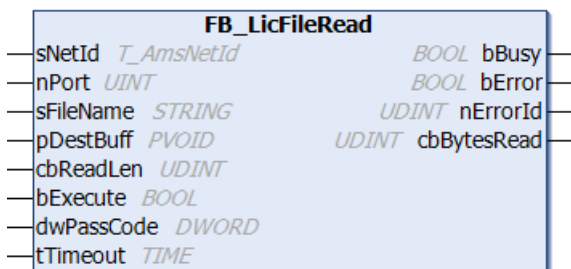
```
VAR_IN_OUT
  stStorageInfo : ST_LicStorageInfo;
END_VAR
```

stStorageInfo: StorageInfo des Lizenz-Dongles (Typ: ST_LicStorageInfo)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022	PC oder CX (x64, x86)	Tc2_Utilities >= 3.3.26.0

3.44 FB_LicFileRead



Der Funktionsbaustein liest über eine steigende Flanke an bExecute eine Datei vom Lizenz-Dongle in einen bereitgestellten Buffer (pDestBuff und cbReadLen). Der Buffer muss groß genug für die Datei sein, sonst wird nur der vordere Teil der Datei gelesen.

VAR_INPUT

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  nPort       : UINT;
  sFileName   : STRING;
```

```
pDestBuff : PVOID;
cbReadLen : UDINT
bExecute  : BOOL;
dwPassCode : DWORD;
tTimeout  : TIME      := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: AmsNetId (AMS-Netzwerkennung) des Lizenz-Dongles (Typ: T_AmsNetId)

- USB-Dongle: AmsNetId des TwinCAT-Rechners. Für den lokalen Rechner kann auch ein Leerstring angegeben werden.
- EL6070: AmsNetId des EtherCAT-Masters (siehe AdsAddr.netId in den InfoData der EL6070)

nPort: AMS-Port des Lizenz-Dongles

- USB: ADS-Port des ESB-Geräts (siehe ADS-Port auf dem ESB Device Tab vom USB-Dongle, Default ist 16#7100)
- EL6070: ADS-Port des EtherCAT-Terminals (siehe AdsAddr.port in den InfoData der EL6070)

sFileName: Name der zu lesenden Datei

pDestBuff: Buffer address for read

cbReadLen: Count of bytes for read

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

dwPassCode: Passcode für den Dateizugriff

tTimeout: Timeout-Zeit, die bei Ausführung des Befehls nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy      : BOOL;
  bError     : BOOL;
  nErrorId   : UDINT;
END_VAR
```

bBusy: TRUE, solange der Baustein aktiv ist.

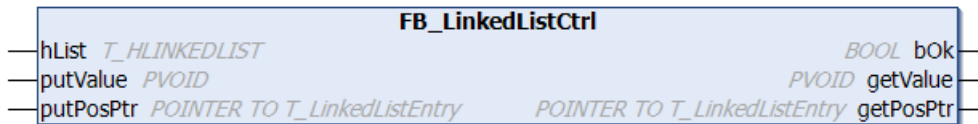
bError: TRUE, wenn bei der Ausführung des Kommandos ein Fehler auftritt.

nErrorId: Liefert bei einem gesetzten bError-Ausgang die ADS-Fehlernummer.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken
TwinCAT v3.1.4022	PC oder CX (x64, x86)	Tc2_Uilities >= 3.3.26.0

3.45 FB_LinkedListCtrl



Mit dem Funktionsbaustein FB_LinkedListCtrl kann eine verkettete Liste im SPS-Projekt realisiert werden. Es wird dabei eine doppelt verkettete Liste erstellt. In einer verketteten Liste können Werte (genannt Knoten) abgespeichert werden. Durch die Liste kann von hinten nach vorne oder umgekehrt iteriert werden. Die Knoten können schnell hinzugefügt oder gelöscht werden.

Die maximale Anzahl der Knoten kann zur Laufzeit nicht verändert werden und muss vor dem Compile-Vorgang festgelegt werden. Als "Knoten-Pool" wird ein Array von Typ: T_LinkedListEntry verwendet. Das Hinzufügen/Entfernen/Suchen der Knoten wird durch Aktionsaufrufe gesteuert. Der Funktionsbaustein besitzt folgende Aktionen:

- **A_AddHeadValue** (Fügt einen neuen Knoten mit dem Wert: *putValue* am Anfang der Liste hinzu. Derselbe Wert kann auch mehrfach hinzugefügt werden. Bei Erfolg liefert *getPosPtr* die Adresse und *getValue* den Wert des neuen Knoten.)
- **A_AddTailValue** (Fügt einen neuen Knoten mit dem Wert: *putValue* am Ende der Liste hinzu. Derselbe Wert kann auch mehrfach hinzugefügt werden. Bei Erfolg liefert *getPosPtr* die Adresse und *getValue* den Wert des neuen Knoten.)
- **A_FindNext** (Sucht nach dem nächsten Knoten (relativ zu *putPosPtr*) mit demselben Wert *putValue*. Bei Erfolg liefert *getPosPtr* die Adresse und *getValue* den Wert des Knoten.)
- **A_FindPrev** (Sucht nach dem vorherigen Knoten (relativ zu *putPosPtr*) mit demselben Wert *putValue*. Bei Erfolg liefert *getPosPtr* die Adresse und *getValue* den Wert des Knoten.)
- **A_GetNext** (Navigiert zum nächsten Knoten (relativ zu *putPosPtr*). Die Adresse: *putPosPtr* muss auf den vorherigen Knoten zeigen! Der Wert *putValue* wird nicht benutzt.)
- **A_GetPrev** (Navigiert zum vorherigen Knoten (relativ zu *putPosPtr*) in die entgegengesetzte Richtung wie *A_GetNext*. Die Adresse: *putPosPtr* muss auf den vorherigen Knoten zeigen! Der Wert *putValue* wird nicht benutzt.)
- **A_GetHead** (Liest den Anfangsknoten. Bei Erfolg liefert *getPosPtr* die Adresse des Knoten und *getValue* den dazugehörigen Wert. Der Wert *putValue* und *putPosPtr* wird nicht benutzt.)
- **A_GetTail** (Liest den Endknoten. Bei Erfolg liefert *getPosPtr* die Adresse des Knoten und *getValue* den dazugehörigen Wert. Der Wert *putValue* und *putPosPtr* wird nicht benutzt.)
- **A_RemoveHeadValue** (Entfernt einen Knoten am Anfang der Liste. Bei Erfolg liefert *getPosPtr* die Adresse und *getValue* den Wert des Knoten. Der Wert *putValue* und *putPosPtr* wird nicht benutzt.)
- **A_RemoveTailValue** (Entfernt einen Knoten am Ende der Liste. Bei Erfolg liefert *getPosPtr* die Adresse und *getValue* den Wert des Knoten. Der Wert *putValue* und *putPosPtr* wird nicht benutzt.)
- **A_RemoveValueAtPosPtr** (Sucht und entfernt einen Knoten mit der Adresse: *putPosPtr*. Bei Erfolg liefert *getPosPtr* die Adresse und *getValue* den Wert des Knoten. Der Wert *putValue* wird nicht benutzt.)
- **A_GetIndexAtPosPtr** (Liefert den Array-Index (vom "Knoten-Pool") des Knoten an der Adresse: *putPosPtr*. Bei Erfolg liefert *getValue* den nullbasierten Array-Index. Der Wert *putValue* wird nicht benutzt. Bitte beachten Sie, dass der Wert *getValue* einen Knoten-Index und nicht den Knoten-Wert zurückliefert!)
- **A_SetValueAtPosPtr** (Aktualisiert/setzt den Wert des Knoten *putValue* an der Adresse *putPosPtr*. Bei Erfolg liefert *getPosPtr* die Adresse und *getValue* den Wert des Knoten.)
- **A_Reset** (Löscht alle Listenelemente und setzt die Liste zurück.)

VAR_IN_OUT

```
VAR_IN_OUT
  hList : T_HLINKEDLIST;
END_VAR
```

hList: Linked-List-Handle (Typ: [T_HLINKEDLIST](#) [▶ 325]). Das Handle muss einmalig vor der Benutzung mit der Funktion: [F_CreateLinkedListHnd](#) [▶ 265] initialisiert werden. Für jede verkettete Liste muss eine dazugehörige Instanz der Handle-Variablen angelegt und initialisiert werden.

VAR_INPUT

```
VAR_INPUT
  putValue : PVOID := 0;
  putPosPtr : POINTER TO T_LinkedListEntry := 0;
END_VAR
```

putValue: Wert/Datenelement (Eingangsparameter, 32/64 Bit, vorzeichenlose Zahl oder auch Pointer).

putPosPtr: Die Adresse des Knotenelements (Eingangsparameter, Typ: [T_LinkedListEntry](#) [▶ 326]).

VAR_OUTPUT

```
VAR_OUTPUT
  bOk : BOOL := FALSE;
  getValue : PVOID := 0;
  getPosPtr : POINTER TO T_LinkedListEntry := 0;
END_VAR
```

bOk: Ergebnis des letzten Aktionsaufrufs. Liefert TRUE wenn ein neues Knotenelement hinzugefügt/entfernt oder in der Liste gefunden werden konnte. FALSE wird geliefert wenn das gesuchte Knotenelement nicht gefunden werden konnte, die Liste leer ist oder einen Überlauf (keine freien Knotenelemente mehr) hat.

getValue: Wert/Datenelement (Ausgangsparameter, 32/64 Bit, vorzeichenlose Zahl oder auch Pointer).

getPosPtr: Die Adresse des Knotenelements (Ausgangsparameter, Typ: [T_LinkedListEntry](#) [► 326]).

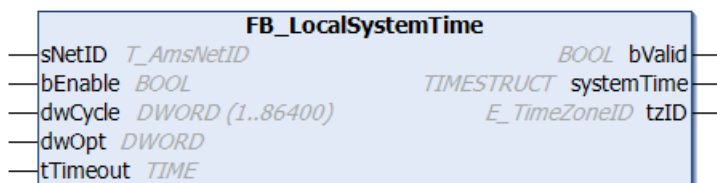
Beispiel:

Siehe: [Beispiel: Verkettete-Liste \(FB_LinkedListCtrl\)](#). [► 345]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.46 FB_LocalSystemTime



In einigen Anwendungen wird die lokale Windows-Systemzeit mit Hilfe des SNTP-Zeitserver oder einer Funkuhr synchronisiert. Oft muss die lokale Windows-Systemzeit (z. B. als Zeitstempel Log-Meldungen an das HMI) in der SPS verwendet werden (die lokale Windows-Systemzeit wird in der Taskleiste eingeblendet). Für solche Anwendungen kann der FB_LocalSystemTime-Funktionsbaustein nützlich sein.

Dieser Funktionsbaustein vereint intern die Funktion folgender Funktionsbausteine: [RTC_EX2](#) [► 140], [NT_GetTime](#) [► 122], [FB_GetTimeZoneInformation](#) [► 78] und [NT_SetTimeToRTCTime](#) [► 125]. Mit Hilfe des RTC_EX2-Bausteins können z. B. Zeitstempel für Log-Ausgaben generiert werden. Dieser Baustein hat aber den Nachteil, dass seine Uhrzeit nicht synchron mit der lokalen Windows Systemzeit läuft und zyklisch mit dem NT_GetTime-Funktionsbaustein nachsynchronisiert werden muss (siehe RTC-Bausteinbeispiele in der Dokumentation). Die zyklische Synchronisierung der eigenen Uhrzeit (*systemTime*-Ausgang) ist bereits in dem Funktionsbaustein implementiert. Die Zykluszeit ist über den *dwCycle*-Eingang konfigurierbar. Außerdem liefert der Funktionsbaustein Sommerzeit-/ Winterzeit-Zeitzoneinformation.

Der FB_LocalSystemTime-Funktionsbaustein muss zyklisch (z. B. jede Sekunde oder jedem Zyklus der SPS) aufgerufen werden. Dies ist notwendig, damit die Uhrzeit zwischen den Synchronisierungen berechnet werden kann.

i Jitter!

Die lokale Windows Systemzeit wird mit Hilfe der azyklischen Dienste (ADS-Funktionsbausteine) gelesen. Systembedingt kann die Laufzeit der ADS-Kommandos nicht festgelegt/geschätzt werden. Durch die unterschiedlichen Kommandolaufzeiten und abhängig vom Betriebssystem, Synchronisations-Intervall und der Zykluszeit der SPS kann die Uhrzeit am *systemTime*-Ausgang jittern. Aus diesem Grund ist die vom Baustein gelieferte Uhrzeit nur bedingt für genauere Messaufgaben geeignet. Die Genauigkeit ist aber z. B. ausreichend für Anwendungen aus dem Bereich der Gebäudeautomatisierung.

Sommerzeit/Winterzeit-Umschaltung

Der Funktionsbaustein kann nicht exakt zu dem Zeitpunkt aufgerufen werden bei dem die Umschaltung von Sommer- auf Winterzeit bzw. umgekehrt stattfindet. Um aufwendige Berechnungen zu vermeiden wurde folgende Implementierung gewählt (am Beispiel erklärt).

In unserem Beispiel synchronisiert der Funktionsbaustein z.B. alle 60 Sekunden die eigene Uhrzeit mit der lokalen Windows Systemzeit (grau).

Die SPS-Applikation benötigt und liest die Uhrzeit am Funktionsbaustein z.B. alle 30 Sekunden (blau). In unserem Beispiel wird die Sommer/Winterzeit-Umschaltung mit einer Verspätung von 15 Sekunden erkannt. Dieses Verhalten dürfte aber für die meisten Applikationen unproblematisch sein.

Umschaltung Winterzeit -> Sommerzeit

- ...
- 30-03-2008-01:58:10, tzID = Winterzeit
- 30-03-2008-01:58:15, nach internen Synchronisation
- 30-03-2008-01:58:40, tzID = Winterzeit
- 30-03-2008-01:59:10, tzID = Winterzeit
- 30-03-2008-01:59:15, nach internen Synchronisation
- 30-03-2008-01:59:40, tzID = Winterzeit
- 30-03-2008-02:00:00, das Betriebssystem stellt die Zeit von 2 Uhr auf 3 Uhr um
- 30-03-2008-02:00:10, tzID = Winterzeit (immer noch!)
- 30-03-2008-03:00:15, nach internen Synchronisation, folgende tzID = Sommerzeit
- 30-03-2008-03:00:40, tzID = Sommerzeit
- 30-03-2008-03:01:10, tzID = Sommerzeit
- 30-03-2008-03:01:15, nach internen Synchronisation
- 30-03-2008-03:01:40, tzID = Sommerzeit
- ...

Umschaltung Sommerzeit -> Winterzeit

- ...
- 26-10-2008-02:58:10, tzID = Sommerzeit
- 26-10-2008-02:58:15, nach internen Synchronisation
- 26-10-2008-02:58:40, tzID = Sommerzeit
- 26-10-2008-02:59:10, tzID = Sommerzeit
- 26-10-2008-02:59:15, nach internen Synchronisation
- 26-10-2008-02:59:40, tzID = Sommerzeit
- 26-10-2008-03:00:00, das Betriebssystem stellt die Zeit von 3 Uhr auf 2 Uhr um
- 26-10-2008-03:00:10, tzID = Sommerzeit (immer noch!)
- 26-10-2008-02:00:15, nach internen Synchronisation, folgende tzID = Winterzeit
- 26-10-2008-02:00:40, tzID = Winterzeit
- 26-10-2008-02:01:10, tzID = Winterzeit
- 26-10-2008-02:01:15, nach internen Synchronisation
- 26-10-2008-02:01:40, tzID = Winterzeit
- ...

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetID := '';
  bEnable     : BOOL;
  dwCycle     : DWORD(1..86400) := 5;
  dwOpt       : DWORD := 1;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Uhrzeit für die Synchronisation benutzt werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

bEnable: Bei einer steigenden Flanke an diesem Eingang wird die sofortige Synchronisation der eigenen Uhrzeit mit der lokalen Windows Systemzeit ausgelöst. Der Ausgang *bValid* wird solange auf FALSE gesetzt bis die Synchronisation abgeschlossen wurde. Durch die erste steigende Flanke wird die zyklische Synchronisation aktiviert. D.h. die danach folgenden zyklischen Synchronisierungen werden automatisch ausgeführt. Die Applikation muss in den meisten Fällen nur ein Mal diesen Eingang auf TRUE setzen.

dwCycle: Zykluszeit (in Sekunden) in der der Funktionsbaustein die eigene Uhrzeit nachsynchronisiert. Die zyklische Synchronisation wird nach der ersten steigenden Flanke am *bEnable*-Eingang aktiviert. Default: Synchronisation alle 5 Sekunden.

dwOpt: Zusätzliche Optionsparameter. Zurzeit stehen folgende Parameter zur Verfügung:

- Bit 0: Wenn gesetzt dann wird zusätzlich die Windows Systemzeit zyklisch zu der Hardware-Uhr (RTC) synchronisiert (entspricht der Funktion: `NT_SetTimeToRTCTime`). Default: Aktiviert. Auf einem Windows CE-System hat diese Option keine Bedeutung.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung der internen ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bValid      : BOOL;
  systemTime  : TIMESTRUCT;
  tzID        : E_TimeZoneID := eTimeZoneID_Invalid;
END_VAR
```

bValid: Die Uhrzeit am *systemTime*-Ausgang ist ungültig wenn dieser Ausgang FALSE ist. Beim TRUE ist die Uhrzeit gültig (wurde mindestens einmal mit der lokalen Windows Zeit synchronisiert).

systemTime: Lokale Windows-Systemzeit (Typ: [TIMESTRUCT](#) [[▶ 327](#)]).

tzID: Zeitzonen-Information (Sommerzeit, Winterzeit) (Typ: [E_TimeZoneID](#) [[▶ 308](#)]).

Beispiel:

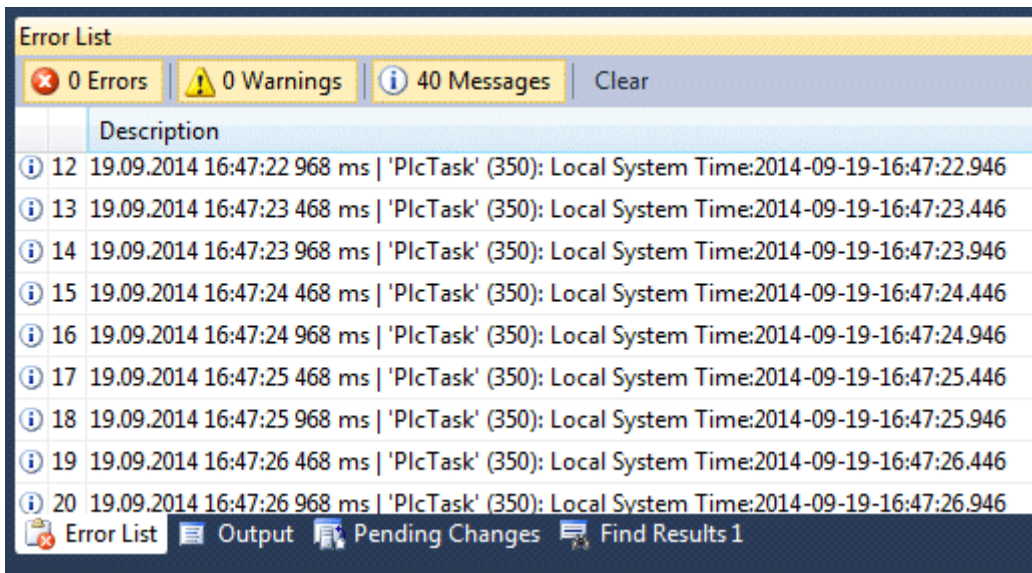
Im Beispiel wird der `FB_LocalSystemTime`-Funktionsbaustein beim Programmstart aktiviert (steigende Flanke am *bEnable*-Eingang). Nachdem die Uhrzeit synchronisiert wurde (*bValid* = TRUE) schreibt die SPS alle 500 ms eine Meldung ins TwinCAT „Error List“-Fenster. Die interne Synchronisierung wird jede Sekunde durchgeführt.

```
PROGRAM MAIN
VAR
  fbTime      : FB_LocalSystemTime := ( bEnable := TRUE, dwCycle := 1 );
  logTimer    : TON := ( IN := TRUE, PT := T#500ms );
END_VAR

fbTime();

logTimer( IN := fbTime.bValid );
IF logTimer.Q THEN
  logTimer( IN := FALSE ); logTimer( IN := fbTime.bValid );
  ADSLOGSTR( ADSLOG_MSGTYPE_HINT OR ADSLOG_MSGTYPE_LOG, 'Local System Time:
%s', SYSTEMTIME_TO_STRING(fbTime.systemTime));
END_IF
```

Die geschriebenen Meldungen können Sie im TwinCAT „Error List“-Fenster sehen.



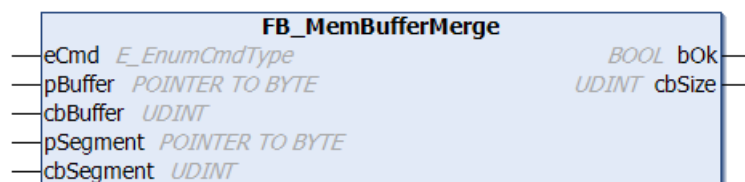
Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

- [FB_TzSpecificLocalTimeToSystemTime \[► 117\]](#)
- [FB_TzSpecificLocalTimeToFileTime \[► 29\]](#)
- [FB_SystemTimeToTzSpecificLocalTime \[► 113\]](#)
- [FB_FileTimeTimeToTzSpecificLocalTime \[► 27\]](#)
- [FB_GetTimeZoneInformation \[► 78\]](#)
- [FB_SetTimeZoneInformation \[► 111\]](#)
- [NT_SetLocalTime \[► 124\]](#)
- [NT_GetTime \[► 122\]](#)
- [NT_SetTimeToRTCTime \[► 125\]](#)
- [F_TranslateFileTimeBias \[► 243\]](#)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Uilities (System)

3.47 FB_MemBufferMerge



Dieser Funktionsbaustein fügt einzelne kleinere Datensegmente zu einem größeren Datensegment zusammen. Der Zielpuffer muss als Eingangsparameter an den Baustein übergeben werden. Es werden keine weiteren Datenbytes hinzugefügt, wenn das Segment, welches hinzugefügt werden soll, die verbliebene freie Puffergröße überschreitet.

VAR_INPUT

```
VAR_INPUT
  eCmd      : E_EnumCmdType := eEnumCmd_First;
  pBuffer   : POINTER TO BYTE;
  cbBuffer  : UDINT;
  pSegment  : POINTER TO BYTE := 0;
  cbSegment : UDINT := 0;
END_VAR
```

eCmd: Steuerparameter für den Aufzählungsbaustein (Typ: `E_EnumCmdType` [► 303]). `eEnumCmd_First` fügt das erste Segment, `eEnumCmd_Next` fügt das nächste Segment hinzu. Andere Parameter werden nicht benutzt.

pBuffer: Adresse (Pointer) auf die Zielpuffervariable. Die Adresse kann mit dem ADR-Operator ermittelt werden.

cbBuffer: Maximal verfügbare Größe (in Byte) der Zielpuffervariablen. Die Größe kann mit dem SIZEOF-Operator ermittelt werden.

pSegment: Adresse (Pointer) auf das nächste Datensegment, welches hinzugefügt werden soll (Optional, kann auch Null sein). Die Adresse kann ebenfalls mit dem ADR-Operator ermittelt werden.

cbSegment: Größe des nächsten Datensegments, welches hinzugefügt werden soll (Optional, kann auch Null sein). Die Größe kann ebenfalls mit dem SIZEOF-Operator ermittelt werden.

VAR_OUTPUT

```
VAR_OUTPUT
  bOk      : BOOL;
  cbSize   : UDINT;
END_VAR
```

bOk: TRUE = Success, FALSE = Pufferüberlauf oder fehlerhafte Eingangsparameter.

cbSize: Der aktuelle Puffer-Füllstatus (Anzahl der Datenbytes im Puffer).

Beispiel:

Im folgenden Beispiel wird zu Testzwecken nach dem Zusammenfügen der kleinen Datensegmente das große Datensegment in einen Hexadezimalstring konvertiert.

```
PROGRAM MAIN
VAR
  bMerge      : BOOL := TRUE;
  fbMerge     : FB_MemBufferMerge;
  buffer      : ARRAY[0..25] OF BYTE;
  seg1        : ARRAY[0..5] OF BYTE := [0,1,2,3,4,5];
  seg2        : ARRAY[0..3] OF BYTE := [6,7,8,9];
  seg3        : ARRAY[0..9] OF BYTE := [10,11,12,13,14,15,16,17,18,19];
  sHex        : T_MaxString;
END_VAR

IF bMerge THEN
  bMerge := FALSE;

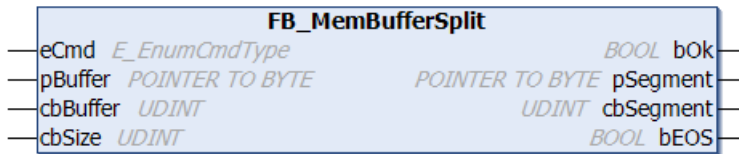
  fbMerge( eCmd := eEnumCmd_First, pBuffer := ADR(buffer), cbBuffer := SIZEOF(buffer), pSegment :=
  ADR(seg1), cbSegment:= SIZEOF(seg1) );
  fbMerge( eCmd := eEnumCmd_Next, pBuffer := ADR(buffer), cbBuffer := SIZEOF(buffer), pSegment :=
  ADR(seg2), cbSegment:= SIZEOF(seg2) );
  fbMerge( pBuffer := ADR(buffer), cbBuffer := SIZEOF(buffer), pSegment := ADR(seg3), cbSegment:=
  SIZEOF(seg3) );
  fbMerge( pBuffer := ADR(buffer), cbBuffer := SIZEOF(buffer), pSegment := 0, cbSegment:= 0 );
  (* merge zero length segment *)
  fbMerge( pBuffer := ADR(buffer), cbBuffer := SIZEOF(buffer), pSegment := ADR(seg3), cbSegment:=
  SIZEOF(seg3) );
  IF NOT fbMerge.bOk THEN
    ;(* TODO: Error handler *)
  END_IF

  sHex := DATA_TO_HEXSTR( pData := ADR(buffer), cbData := fbMerge.cbSize, FALSE );
END_IF
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.48 FB_MemBufferSplit



Dieser Funktionsbaustein teilt einen Speicherbereich (Datenpuffer) in mehrere kleinere Segmente von einer maximal gewünschten Länge auf. Der Funktionsbaustein liefert ein kleineres Teilsegment zurück, falls das letzte Segment eine kleinere Länge als die gewünschte besitzt.

VAR_INPUT

```
VAR_INPUT
    eCmd      : E_EnumCmdType := eEnumCmd_First;
    pBuffer   : POINTER TO BYTE;
    cbBuffer  : UDINT;
    cbSize    : UDINT;
END_VAR
```

eCmd: Steuerparameter für den Funktionsbaustein (Typ: [E_EnumCmdType](#) [[▶ 303](#)]). `eEnumCmd_First` liefert das erste Segment, `eEnumCmd_Next` liefert das nächste Segment. Andere Parameter werden nicht benutzt.

pBuffer: Adresse (Pointer) des Datenpuffers, der geteilt werden soll. Die Adresse kann mit dem `ADR`-Operator ermittelt werden.

cbBuffer: Länge des Datenpuffers, der geteilt werden soll. Die Länge kann mit dem `SIZEOF`-Operator ermittelt werden.

cbSize: Maximale Segmentgröße, in die der Datenpuffer geteilt werden soll.

VAR_OUTPUT

```
VAR_OUTPUT
    bOk       : BOOL;
    pSegment  : POINTER TO BYTE;
    cbSegment : UDINT;
    bEOS      : BOOL;
END_VAR
```

bOk: TRUE = Success, FALSE = Fehler, falscher Parameterwert oder kein weiteres Segment vorhanden.

pSegment: Adresse (Pointer) auf das nächste Datensegment.

cbSegment: Länge (Bytes) von dem nächsten Datensegment.

bEOS: End of segment. TRUE = Letztes Segment. FALSE = Weitere Segmente folgen.

Beispiel:

Im folgenden Beispiel wird die `buffer`-Variable in 5-Byte-Segmente aufgeteilt. Zu Testzwecken werden die zurück gelieferten Segmente in einen Hexadezimalstring konvertiert.

```
PROGRAM MAIN
VAR
    bSplit : BOOL := TRUE;
    buffer : ARRAY[1..30] OF BYTE := [16#A,1,2,3,4,5,6,7,8,9,16#B,1,2,3,4,5,6,7,8,9,16#C,1,2,3,4,5,6,7,8,9];
    fbSplit : FB_MemBufferSplit;
    sHex    : T_MaxString;
END_VAR
```

```

IF bSplit THEN
  bSplit := FALSE;
  fbSplit.eCmd := eEnumCmd_First;

  REPEAT
    fbSplit( pBuffer := ADR(buffer), cbBuffer := SIZEOF(buffer), cbSize := 5 );
    IF fbSplit.bOk THEN
      sHex := DATA_TO_HEXSTR( pData := fbSplit.pSegment, cbData := fbSplit.cbSegment, FALSE );
      fbSplit.eCmd := eEnumCmd_Next;
    END_IF
  UNTIL NOT fbSplit.bOk
  END_REPEAT
END_IF

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.49 FB_MemRingBuffer



Mit dem Funktionsbaustein FB_MemRingBuffer können Datensätze unterschiedlicher Länge in einen Ringpuffer geschrieben oder die vorher geschriebenen Datensätze aus dem Ringpuffer ausgelesen werden. Die geschriebenen Datensätze werden nach dem FIFO-Prinzip in der gleichen Reihenfolge ausgelesen in der sie vorher in den Ringpuffer geschrieben wurden. D.h. beim Lesen werden zuerst die ältesten Einträge ausgelesen. Der Pufferspeicher wird dem Funktionsbaustein über die *pBuffer* / *cbBuffer*-Eingangsvariablen zur Verfügung gestellt. Das Schreiben/Lesen der Datensätze wird durch Aktionsaufrufe gesteuert. Der Funktionsbaustein besitzt folgende Aktionen:

- **A_AddTail** (Schreibt einen neuen Datensatz in den Ringpuffer.)
- **A_GetHead** (Liest den ältesten Datensatz aus dem Ringpuffer, entfernt ihn aber nicht.)
- **A_RemoveHead** (Liest und entfernt den ältesten Datensatz aus dem Ringpuffer.)
- **A_Reset** (Löscht alle Datensätze im Ringpuffer.)

VAR_INPUT

```

VAR_INPUT
  pWrite : POINTER TO BYTE;
  cbWrite : UDINT;
  pRead : POINTER TO BYTE;
  cbRead : UDINT;
  pBuffer : POINTER TO BYTE;
  cbBuffer: UDINT;
END_VAR

```

pWrite: Adresse der SPS-Variablen oder einer Puffervariablen, die die zu schreibenden Value-Daten enthält. Die Adresse kann mit dem ADR-Operator ermittelt werden. Der Programmierer ist selbst dafür verantwortlich die Puffervariable so zu dimensionieren, dass *cbWrite*-Datenbytes daraus entnommen werden können.

cbWrite: Anzahl der zu schreibenden Value-Datenbytes (Bei Stringvariablen inklusive der abschließenden Null).

pRead: Adresse der SPS-Variablen oder einer Puffervariablen in welche die gelesenen Value-Daten hineinkopiert werden sollen. Die Adresse kann mit dem ADR-Operator ermittelt werden. Der Programmierer ist selbst dafür verantwortlich die Puffervariable so zu dimensionieren, dass diese cbRead-Datenbytes aufnehmen kann. Die Bytegröße der Puffervariablen muss größer oder gleich der Größe des zu lesenden Datensatzes sein.

cbRead: Anzahl der zu lesenden Value-Datenbytes. Bei einer zu kleinen Puffergröße werden keine Daten kopiert, der Funktionsbaustein meldet einen Puffer-Underflow-Fehler (bOk = FALSE) und die benötigte Puffergröße für den nächsten zu lesenden Datensatz wird am *cbReturn*-Ausgang zurückgeliefert.

pBuffer: Adresse einer SPS-Variablen (z.B. ARRAY[...] OF BYTES) die vom Funktionsbaustein als Pufferspeicher benutzt werden soll. Die Adresse kann mit dem ADR-Operator ermittelt werden.

cbBuffer: Maximale Bytegröße der SPS-Variablen die als Pufferspeicher benutzt werden soll. Die Größe kann mit dem SIZEOF-Operator ermittelt werden.

VAR_OUTPUT

```
VAR_OUTPUT
  bOk      : BOOL;
  nCount   : UDINT;
  cbSize   : UDINT;
  cbReturn : UDINT;
END_VAR
```

bOk: Liefert TRUE wenn ein neuer Datensatz erfolgreich hinzugefügt oder entfernt werden konnte und FALSE beim Puffer-Überlauf oder wenn keine Einträge im Puffer mehr vorhanden sind.

nCount: Liefert die aktuelle Anzahl der gepufferten Datensätze.

cbSize: Liefert die aktuelle Anzahl der belegten Datenbytes im Puffer. Die Anzahl der belegten Datenbytes ist immer größer als die tatsächliche Anzahl der geschriebenen Value-Daten. Jeder Datensatz wird um zusätzliche Informationen ergänzt um ihn später lokalisieren zu können.

cbReturn: Anzahl der erfolgreich gelesenen Value-Datenbytes. Beim Lesebuffer-Underflow-Fehler liefert dieser Ausgang die benötigte Lesebuffer-Bytegröße. In diesem Fall ist die *cbRead*-Länge zu klein dimensioniert.

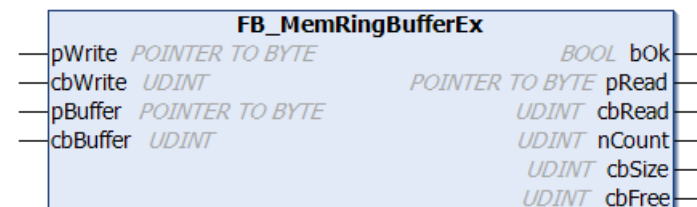
Beispiel:

Siehe: [Beispiel: Memory-Ring-FiFo \(FB MemRingBuffer\)](#). [▶ 339]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.50 FB_MemRingBufferEx



Mit dem Funktionsbaustein FB_MemRingBufferEx können Datensätze unterschiedlicher Länge in einen Ringpuffer geschrieben oder die vorher geschriebenen Datensätze aus dem Ringpuffer ausgelesen werden. Die geschriebenen Datensätze werden nach dem FIFO-Prinzip in der gleichen Reihenfolge ausgelesen in der sie vorher in den Ringpuffer geschrieben wurden. D.h. beim Lesen werden zuerst die ältesten Einträge ausgelesen. Der Pufferspeicher wird dem Funktionsbaustein über die *pBuffer* / *cbBuffer*-Eingangsvariablen zur Verfügung gestellt. Das Schreiben/Lesen der Datensätze wird durch Aktionsaufrufe gesteuert.

Dieser Funktionsbaustein ähnelt in der Funktion dem [FB_MemRingBuffer \[► 96\]](#)-Funktionsbaustein. Beim Lesen der Datensätze kopiert der `FB_MemRingBuffer` die Daten in eine externe Puffervariable. Der `FB_MemRingBufferEx` liefert aber nur eine Referenz auf den Datensatz (Adresspointer/Länge). Die Applikation muss dann die Daten für die Weiterverarbeitung selber umkopieren.

Der Funktionsbaustein besitzt folgende Aktionen:

- **A_AddTail** (Schreibt einen neuen Datensatz in den Ringpuffer.)
- **A_GetHead** (Liefert eine Referenz: Adresspointer/Länge auf den ältesten Datensatz aus dem Ringpuffer, entfernt ihn aber nicht.)
- **A_FreeHead** (Liest und entfernt den ältesten Datensatz aus dem Ringpuffer. Der zurück gelieferte Adresspointer/Länge ist Null! Das freie Speichersegment wird für einen neuen Datensatz freigegeben.)
- **A_Reset** (Löscht alle Datensätze im Ringpuffer.)
- **A_GetFreeSize** (Liefert die Bytegröße des größten freien Speichersegments im Puffer)

VAR_INPUT

```
VAR_INPUT
  pWrite   : POINTER TO BYTE;
  cbWrite  : UDINT;
  pBuffer  : POINTER TO BYTE;
  cbBuffer : UDINT;
END_VAR
```

pWrite: Adresse der SPS-Variablen oder einer Puffervariablen, die die zu schreibenden Value-Daten enthält. Die Adresse kann mit dem ADR-Operator ermittelt werden. Der Programmierer ist selbst dafür verantwortlich die Puffervariable so zu dimensionieren, dass `cbWrite`-Datenbytes daraus entnommen werden können.

cbWrite: Anzahl der zu schreibenden Value-Datenbytes (bei Stringvariablen inklusive der abschließenden Null). Die Größe kann mit dem SIZEOF-Operator ermittelt werden.

pBuffer: Adresse einer SPS-Variablen (z.B. ARRAY[...] OF BYTES) die vom Funktionsbaustein als Pufferspeicher benutzt werden soll. Die Adresse kann mit dem ADR-Operator ermittelt werden.

cbBuffer: Max. Bytegröße der SPS-Variablen die als Pufferspeicher benutzt werden soll. Die Größe kann mit dem SIZEOF-Operator ermittelt werden.

VAR_OUTPUT

```
VAR_OUTPUT
  bOk      : BOOL;
  pRead    : POINTER TO BYTE;
  cbRead   : UDINT;
  nCount   : UDINT;
  cbSize   : UDINT;
  cbFree   : UDINT;
END_VAR
```

bOk: Liefert TRUE wenn ein neuer Datensatz erfolgreich hinzugefügt oder entfernt werden konnte und FALSE beim Puffer-Überlauf oder wenn keine Einträge im Puffer mehr vorhanden sind.

pRead: Diese Variable liefert nach dem Aufruf der Aktion: `A_GetHead` beim Erfolg (`bOk=TRUE`) eine Referenz (Adresspointer) auf den ältesten Datensatz im Ringpuffer. Es wird Null zurückgeliefert wenn keine Datensätze mehr im Ringpuffer vorhanden sind.

cbRead: Diese Variable liefert nach dem Aufruf der Aktion: `A_GetHead` beim Erfolg (`bOk=TRUE`) die Länge vom ältesten Datensatz im Ringpuffer. Es wird Null zurückgeliefert wenn keine Datensätze mehr im Ringpuffer vorhanden sind.

nCount: Liefert die aktuelle Anzahl der gepufferten Datensätze.

cbSize: Liefert die aktuelle Anzahl der belegten Datenbytes im Puffer. Die Anzahl der belegten Datenbytes ist immer größer als die tatsächliche Anzahl der geschriebenen Value-Daten. Jeder Datensatz wird um zusätzliche Informationen ergänzt um ihn später lokalisieren zu können.

cbFree: Liefert nach dem Aufruf der Aktion: *A_GetFreeSize* die Bytegröße des größten freien Speichersegments im Puffer. Die Datensätze müssen auf kontinuierlichen Adressen im Pufferspeicher vorhanden sein da der Funktionsbaustein Referenz auf die Datensätze zurückliefert. Dies führt automatisch zu Segmentierung am Pufferende. Dieser Speicher kann nicht verwendet werden wenn der neue Datensatz größer ist als das freie Segment am Pufferende.

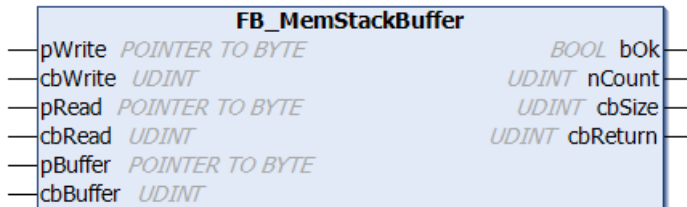
Beispiel:

Siehe: [Beispiel: Memory-Ring FiFo \(FB_MemRingBufferEx\)](#) [► 340].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.51 FB_MemStackBuffer



Mit dem Funktionsbaustein *FB_MemStackBuffer* können Datensätze unterschiedlicher Länge in einen Puffer geschrieben oder die vorher geschriebenen Datensätze aus dem Puffer ausgelesen werden. Die geschriebenen Datensätze werden nach dem LIFO-Prinzip (Last In - First Out) in der umgekehrten Reihenfolge ausgelesen in der sie vorher in den Puffer geschrieben wurden. D.h. beim Lesen werden zuerst die neuesten Einträge ausgelesen. Der Pufferspeicher wird dem Funktionsbaustein über die *pBuffer* und *cbBuffer*-EingangsvARIABLEN zur Verfügung gestellt. Das Schreiben/Lesen der Datensätze wird durch Aktionsaufrufe gesteuert. Der Funktionsbaustein besitzt folgende Aktionen:

- **A_Push():** Schreibt einen neuen Datensatz in den Puffer;
- **A_Top():** Liest den zuletzt hinzugefügten/neuesten Datensatz aus dem Puffer, entfernt ihn aber nicht;
- **A_Pop():** Liest und entfernt den zuletzt hinzugefügten/neuesten Datensatz aus dem Puffer;
- **A_Reset():** Löscht alle Datensätze im Puffer;

VAR_INPUT

```

VAR_INPUT
  pWrite      : POINTER TO BYTE;
  cbWrite     : UDINT;
  pRead      : POINTER TO BYTE;
  cbRead     : UDINT;
  pBuffer    : POINTER TO BYTE;
  cbBuffer   : UDINT;
END_VAR
    
```

pWrite: Adresse der SPS-VARIABLEN oder einer PuffervARIABLEN, die die zu schreibenden Value-Daten enthält. Die Adresse kann mit dem ADR-Operator ermittelt werden. Der Programmierer ist selbst dafür verantwortlich die PuffervARIABLE so zu dimensionieren, dass *cbWrite*-Datenbytes daraus entnommen werden können.

cbWrite: Anzahl der zu schreibenden Value-Datenbytes (Bei Stringvariablen inklusive der abschließenden Null).

pRead: Adresse der SPS-Variablen oder einer Puffervariablen in welche die gelesenen Value-Daten hineinkopiert werden sollen. Die Adresse kann mit dem ADR-Operator ermittelt werden. Der Programmierer ist selbst dafür verantwortlich die Puffervariable so zu dimensionieren, dass diese *cbRead*-Datenbytes aufnehmen kann. Die Bytegröße der Puffervariablen muss größer oder gleich der Größe des zu lesenden Datensatzes sein.

cbRead: Anzahl der zu lesenden Value-Datenbytes. Bei einer zu kleinen Puffergröße werden keine Daten kopiert, der Funktionsbaustein meldet einen Puffer-Underflow-Fehler (*bOk* = FALSE) und die benötigte Puffergröße für den nächsten zu lesenden Datensatz wird am *cbReturn*-Ausgang zurückgeliefert.

pBuffer: Adresse einer SPS-Variablen (z.B. ARRAY[...] OF BYTES) die vom Funktionsbaustein als Pufferspeicher benutzt werden soll. Die Adresse kann mit dem ADR-Operator ermittelt werden.

cbBuffer: Maximale Bytegröße der SPS-Variablen die als Pufferspeicher benutzt werden soll. Die Größe kann mit dem SIZEOF-Operator ermittelt werden.

VAR_OUTPUT

```
VAR_OUTPUT
  bOk      : BOOL;
  nCount   : UDINT;
  cbSize   : UDINT;
  cbReturn : UDINT;
END_VAR
```

bOk: Liefert TRUE wenn ein neuer Datensatz erfolgreich hinzugefügt oder entfernt werden konnte und FALSE beim Puffer-Überlauf oder wenn keine Einträge im Puffer mehr vorhanden sind.

nCount: Liefert die aktuelle Anzahl der gepufferten Datensätze.

cbSize: Liefert die aktuelle Anzahl der belegten Datenbytes im Puffer. Die Anzahl der belegten Datenbytes ist immer größer als die tatsächliche Anzahl der geschriebenen Value-Daten. Jeder Datensatz wird um zusätzliche Informationen ergänzt um ihn später lokalisieren zu können.

cbReturn: Anzahl der erfolgreich gelesenen Value-Datenbytes. Beim Lesepuffer-Underflow-Fehler liefert dieser Ausgang die benötigte Lesepuffer-Bytegröße. In diesem Fall ist die *cbRead*-Länge zu klein dimensioniert.

Beispiel:

Folgendes Beispiel zeigt eine einfache Verwendung des Funktionsbausteins. Es sollen Strings unterschiedlicher Länge gepuffert werden. Die steigende Flanke am *bReset* löscht den Puffer. Wenn Sie *bAdd* = TRUE setzen werden 10 neue Strings in den Puffer geschrieben und beim *bRemove*=TRUE wird der zuletzt geschriebene String aus dem Puffer entfernt. Bei einer steigenden Flanke am *bGet* wird der zuletzt geschriebene String gelesen aber nicht entfernt.

Deklarationsteil:

```
PROGRAM MAIN
VAR
  buffer      : ARRAY[0..1000] OF BYTE;
  fbStack     : FB_MemStackBuffer;
  bReset      : BOOL := TRUE;
  bAdd        : BOOL := TRUE;
  bGet        : BOOL := TRUE;
  bRemove     : BOOL := TRUE;
  putEntry    : ARRAY[0..9] OF STRING(20) := ['Str_1', 'Str_2', 'Str_3', 'Str_4', 'Str_5', 'Str_6', 'Str_7', 'Str_8', 'Str_9', 'Str_10'];
  getEntry    : STRING;
  i           : UDINT;
END_VAR
```

Programmcode:

```
IF bReset THEN(* Clear buffer *)
  bReset := FALSE;
  fbStack.A_Reset( pBuffer := ADR( buffer ), cbBuffer := SIZEOF( buffer ) );
END_IF

IF bAdd THEN(* Add entries *)
  bAdd := FALSE;
  FOR i:= 0 TO 9 BY 1 DO
    fbStack.A_Push( pBuffer := ADR(buffer), cbBuffer := SIZEOF(buffer), pWrite := ADR(putEntry[i
```

```

]), cbWrite := LEN(putEntry[i]) + 1 );
    IF fbStack.bOk THEN(* Success *)
        ;
    ELSE(* Buffer overflow *)
        ;
    END_IF
END_FOR
END_IF

IF bGet THEN(* Peek newest entry *)
    bGet := FALSE;
    fbStack.A_Top( pBuffer := ADR(buffer), cbBuffer := SIZEOF(buffer), pRead := ADR(getEntry), cbRead := SIZEOF(getEntry) );
    IF fbStack.bOk THEN(* Success *)
        ;
    ELSE(* Buffer is empty *)
        ;
    END_IF
END_IF

IF bRemove THEN(* Remove newest entry *)
    bRemove := FALSE;
    fbStack.A_Pop( pBuffer := ADR(buffer), cbBuffer := SIZEOF(buffer), pRead := ADR(getEntry), cbRead := SIZEOF(getEntry) );
    IF fbStack.bOk THEN(* Success *)
        ;
    ELSE(* Buffer is empty *)
        ;
    END_IF
END_IF

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.52 FB_RegQueryValue



Die Systemregistrierung ist ein hierarchisch strukturierter Baum. Ein Knoten im Baum wird als Schlüssel (Key) bezeichnet. Jeder Schlüssel kann wiederum Unterschlüssel (SubKeys) und Datenwerte (Values) enthalten. Mit dem Funktionsbaustein "FB_RegQueryValue" können einzelne Datenwerte (Values) aus der Systemregistrierung aus dem Zweig mit dem vordefinierten Handle **HKEY_LOCAL_MACHINE** ausgelesen werden. Beim Erfolg werden *cbData*-Datenbytes in den Puffer mit der Adresse *pData* hineinkopiert. Mit dem Funktionsbaustein können beliebige Value-Typen (z. B. REG_DWORD, REG_SZ) oder Binärdaten mit einer unbegrenzten Bytelänge (REG_BINARY) ausgelesen werden.

Bemerkung:

Die *sSubKey* und *sValueName*-Strings dürfen keine Leerstrings sein!

i HKEY_LOCAL_MACHINE\SOFTWARE\ bei 64 Bit Betriebssystemen

In einem 64 Bit Windows Betriebssystem werden alle Registry Einträge von und für 32 Bit Anwendungen nicht unter HKEY_LOCAL_MACHINE\SOFTWARE\ sondern unter HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\ abgelegt.

Die Funktionsbausteine FB_RegQueryValue und FB_RegSetValue arbeiten wie jede 32 Bit Anwendung automatisch unterhalb des WOW6432Node-Ordners, wenn ein Registry Eintrag unterhalb vom SOFTWARE-Ordner ausgewählt wird. Die Umleitung wird automatisch vom Betriebssystem ausgeführt.

VAR_INPUT

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  sSubKey     : T_MaxString;
  sValName    : T_MaxString;
  cbData      : UDINT;
  pData      : POINTER TO BYTE;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Systemregistrierung gelesen werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

sSubKey: String mit dem SubKey-Namen (Typ: T_MaxString).

sValName: String mit dem Value-Namen (Typ: T_MaxString).

cbData: Anzahl der zu lesenden Value-Datenbytes.

pData: Adresse eines Datenpuffers/Variablen in den/die Value-Daten hineinkopiert werden sollen. Die Adresse kann dem ADR-Operator ermittelt werden. Der Programmierer ist selbst dafür verantwortlich den Datenpuffer so zu dimensionieren, dass dieser cbData-Datenbytes aufnehmen kann.

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrId      : UDINT;
  cbRead      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

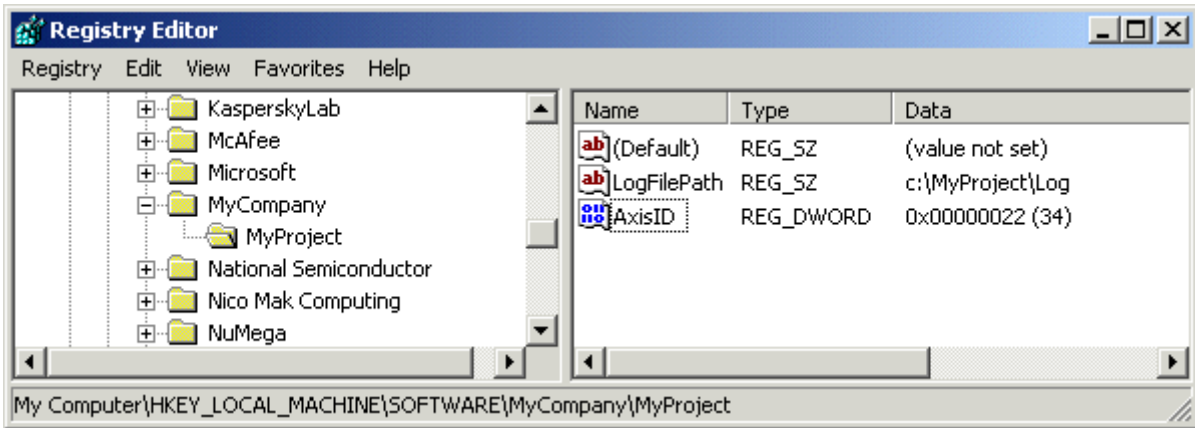
nErrId: Liefert bei einem gesetzten *bError*-Ausgang die [ADS-Fehlernummer \[► 360\]](#) oder einen Befehlsspezifischen Fehlercode zurück (Tabelle).

cbRead: Anzahl der erfolgreich gelesenen Value-Datenbytes.

Fehlercodes	Fehlerbeschreibung
0x00	Kein Fehler
0x01	Der Key mit dem Namen sSubKey konnte nicht geöffnet/gefunden werden.
0x02	Der Schlüsselwert mit dem Namen sValName konnte nicht geöffnet/gefunden werden.

Beispiele:

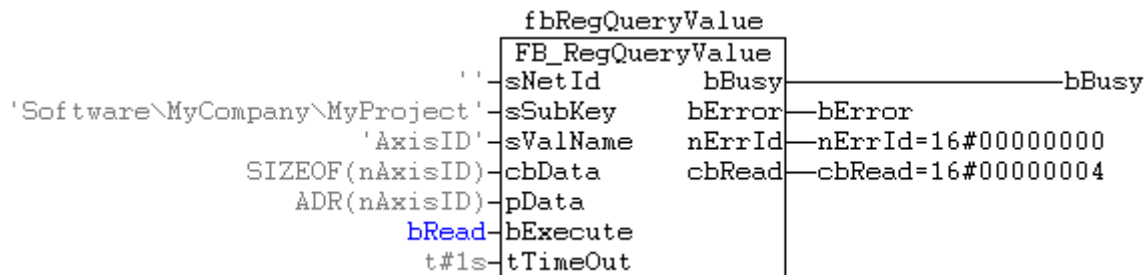
Es sollen aus der Systemregistrierung die Werte *AxisID* und *LogFilePath* gelesen werden.



```
PROGRAM MAIN
VAR
  fbRegQueryValue : FB_RegQueryValue;
  bRead           : BOOL;
  bBusy          : BOOL;
  bError         : BOOL;
  nErrId         : UDINT;
  cbRead         : UDINT;
  sValData       : STRING;
  nAxisID        : DWORD;
END_VAR
```

REG_DWORD-Value lesen:

```
fbRegQueryValue
bRead = TRUE
bBusy = FALSE
bError = FALSE
nErrId = 16#00000000
cbRead = 16#00000004
nAxisID = 16#00000022
```

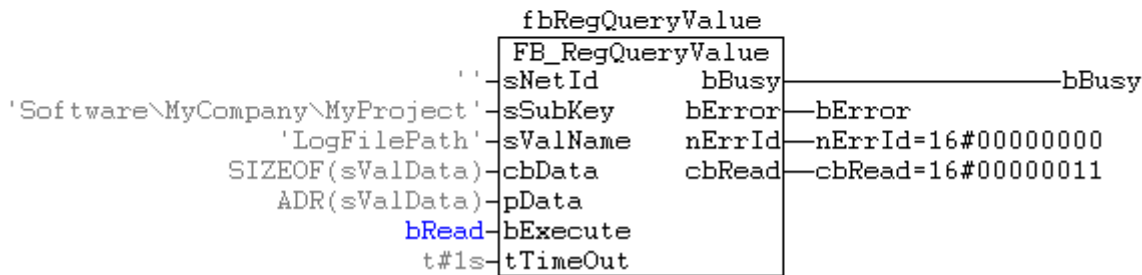


Hier wurde der Wert 0x22 aus der Registrierung in die SPS-Variablen *nAxisID* eingelesen.

REG_SZ-Value lesen:

```

fbRegQueryValue
-----
bRead = TRUE
bBusy = FALSE
bError = FALSE
nErrId = 16#00000000
cbRead = 16#00000011
sValData = 'c:\MyProject\Log'
    
```



Hier wurde der String 'c:\MyProject\Log' aus der Registrierung in die SPS-Variable *sValData* eingelesen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Uilities (System)

3.53 FB_RegSetValue



Die Systemregistrierung ist ein hierarchisch strukturierter Baum. Ein Knoten im Baum wird als Schlüssel (Key) bezeichnet. Jeder Schlüssel kann wiederum Unterschlüssel (SubKeys) und Datenwerte (Values) enthalten.

Mit dem Funktionsbaustein "FB_RegSetValue" können einzelne Schlüsselwerte (Values) oder neue Schlüsselnamen und Werte (SubKeys+Values) in dem Zweig mit dem vordefinierten Handle **HKEY_LOCAL_MACHINE** geschrieben bzw. generiert werden. Es können beliebige Value-Typen (z. B. REG_DWORD, REG_SZ) oder maximal 500 Byte Binärdaten (REG_BINARY) in die Systemregistrierung geschrieben werden. Ist ein Schlüsselwert noch nicht vorhanden, dann wird dieser automatisch neu erzeugt.

Bemerkung:

Die *sSubKey* und *sValueName*-Strings dürfen keine Leerstrings sein!

i HKEY_LOCAL_MACHINE\SOFTWARE\ bei 64 Bit Betriebssystemen

In einem 64 Bit Windows Betriebssystem werden alle Registry Einträge von und für 32 Bit Anwendungen nicht unter HKEY_LOCAL_MACHINE\SOFTWARE\ sondern unter HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\ abgelegt.

Die Funktionsbausteine FB_RegQueryValue und FB_RegSetValue arbeiten wie jede 32 Bit Anwendung automatisch unterhalb des WOW6432Node-Ordners, wenn ein Registry Eintrag unterhalb vom SOFTWARE-Ordner ausgewählt wird. Die Umleitung wird automatisch vom Betriebssystem ausgeführt.

VAR_INPUT

```
VAR_INPUT
  sNetId      : T_AmsNetId;
  sSubKey     : T_MaxString;
  sValName    : T_MaxString;
  eValType    : E_RegValueType;
  cbData      : UDINT;
  pData      : POINTER TO BYTE;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetId: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Systemregistrierung geschrieben werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

sSubKey: String mit dem SubKey-Namen (Typ: T_MaxString).

sValName: String mit dem Value-Namen (Typ: T_MaxString).

eValType: Das Datentypformat der zu schreibenden Registrierungsdaten z. B: REG_DWORD oder REG_SZ (Typ: E_RegValueType [▶ 306]).

cbData: Anzahl der zu schreibenden Value-Datenbytes (Bei Stringvariablen inklusiv der abschließenden Null).

pData: Adresse von einem Datenpuffer/SPS-Variablen, der/die Value-Daten enthält. Die Adresse kann dem ADR-Operator ermittelt werden. Der Programmierer ist selbst dafür verantwortlich den Datenpuffer so zu dimensionieren, dass *cbData*-Datenbytes daraus entnommen werden können.

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrId      : UDINT;
  cbWrite     : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [▶ 360] oder den Befehlsspezifischen Fehlercode (Tabelle).

cbWrite: Anzahl der erfolgreich geschriebenen Value-Datenbytes.

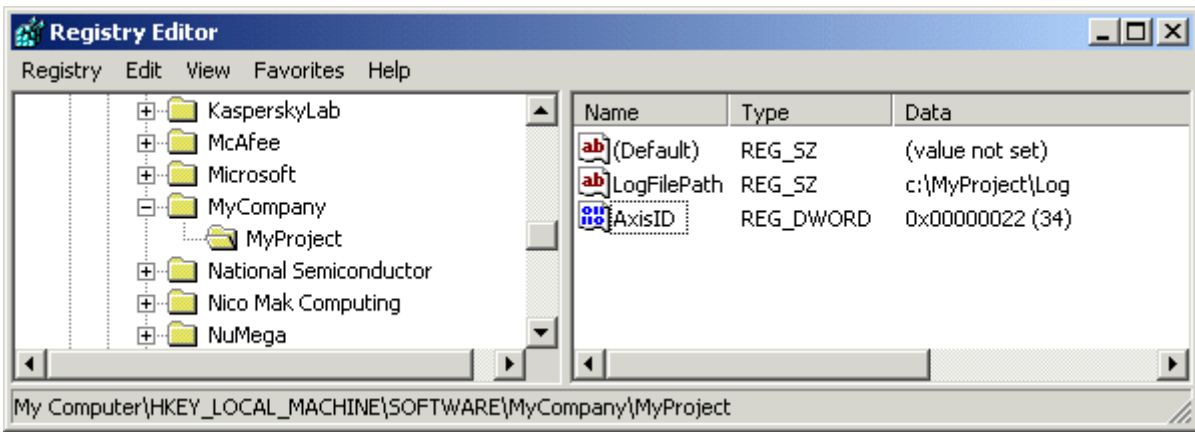
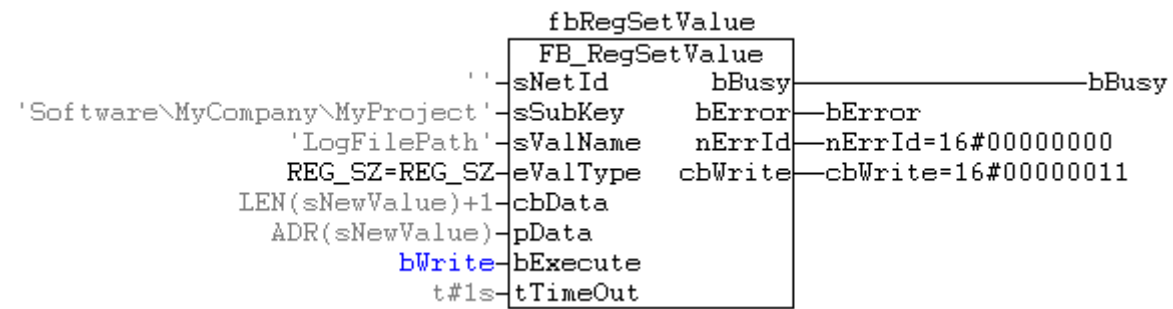
Fehlercodes	Fehlerbeschreibung
0x00	Kein Fehler

Fehlercodes	Fehlerbeschreibung
0x01	Der Key mit dem Namen sSubKey konnte nicht geöffnet/gefunden werden.
0x02	Der Schlüsselwert mit dem Namen sValName konnte nicht geöffnet/gefunden werden.

Beispiel:

In dem Zweig mit dem vordefinierten Handle *HKEY_LOCAL_MACHINE* soll ein SubKey 'SOFTWARE\MyCompany\MyProject' mit dem Schlüsselnamen 'LogFilePath', dem Typ *REG_SZ* und Wert 'c:\MyProject\Log' neu angelegt und gesetzt werden.

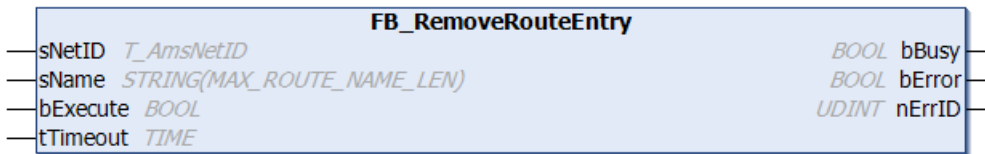
```
PROGRAM MAIN
VAR
    fbRegSetValue : FB_RegSetValue;
    bBusy         : BOOL;
    bError        : BOOL;
    nErrId        : UDINT;
    cbWrite       : UDINT;
    bWrite        : BOOL;
    sNewValue     : STRING := 'c:\MyProject\Log';
END_VAR
```



Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Uilities (System)

3.54 FB_RemoveRouteEntry



Mit dem Funktionsbaustein kann eine vorhandene Verbindung zu einem TwinCAT System aus der Liste der AMS Router Verbindungen (Remote Routes) gelöscht werden.

VAR_INPUT

```
VAR_INPUT
  sNetID   : T_AmsNetID;
  sName    : STRING(MAX_ROUTE_NAME_LEN);
  bExecute : BOOL;
  tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden auf dem die AMS Router Verbindung gelöscht werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

sName: Verbindungsname der Verbindung die gelöscht werden soll. Die max. Stringlänge ist durch eine Konstante begrenzt (default: 31 Zeichen).

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy   : BOOL;
  bError  : BOOL;
  nErrId  : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

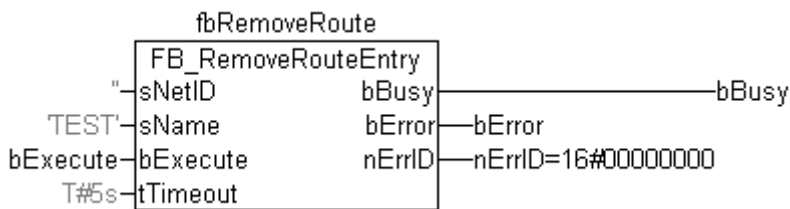
bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten *bError*-Ausgang die ADS-Fehlernummer [► 360].

Beispiel:

Die Verbindung mit dem Verbindungsnamen: "TEST" soll aus der Liste der AMS Router Verbindungen auf dem lokalen TwinCAT System gelöscht werden. Bei einer steigenden Flanke an der bExecute-Variablen wird die Verbindung gelöscht.

```
PROGRAM P_TEST2
VAR
  fbRemoveRoute : FB_RemoveRouteEntry;
  bExecute      : BOOL;
  bBusy        : BOOL;
  bError       : BOOL;
  nErrID      : UDINT;
END_VAR
```



Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.55 FB_ScopeServerControl



Der Funktionsbaustein "FB_ScopeServerControl" ermöglicht der SPS Daten zu sammeln, die dann später mit TwinCAT Scope 2 angezeigt werden sollen.

VAR_INPUT

```

VAR_INPUT
  sNetId      : T_AmsNetId;
  eReqState   : E_ScopeServerState := SCOPE_SERVER_IDLE;
  sConfigFile : STRING;
  sSaveFile   : STRING;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
    
```

sNetId: Hier kann ein String angegeben werden der die Netzwerkadresse des TwinCAT Zielsystems beinhaltet (Typ: T_AmsNetId). Für den lokalen Computer kann dieser String auch leer sein.

eReqState: Angeforderter Scope Server Status (Typ: E_ScopeServerState [[▶ 308](#)]).

sConfigFile: Vollständiger Pfad mit dem Namen der Konfigurations-Datei (z.B.: 'C:\TwinCAT\TwinCATScope2\First.sv2').

sSaveFile: Vollständiger Pfad mit dem Namen der Daten-Datei (z.B.: 'C:\TwinCAT\TwinCATScope2\First.svd').

tTimeout: Maximal zulässige Zeit für die internen ADS Kommandos

VAR_OUTPUT

```

VAR_OUTPUT
  bBusy      : BOOL;
  bDone      : BOOL;
  bError     : BOOL;
  nErrorId   : UDINT;
END_VAR
    
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bDone: Wird gesetzt wenn der angeforderte Status aktiviert wurde.

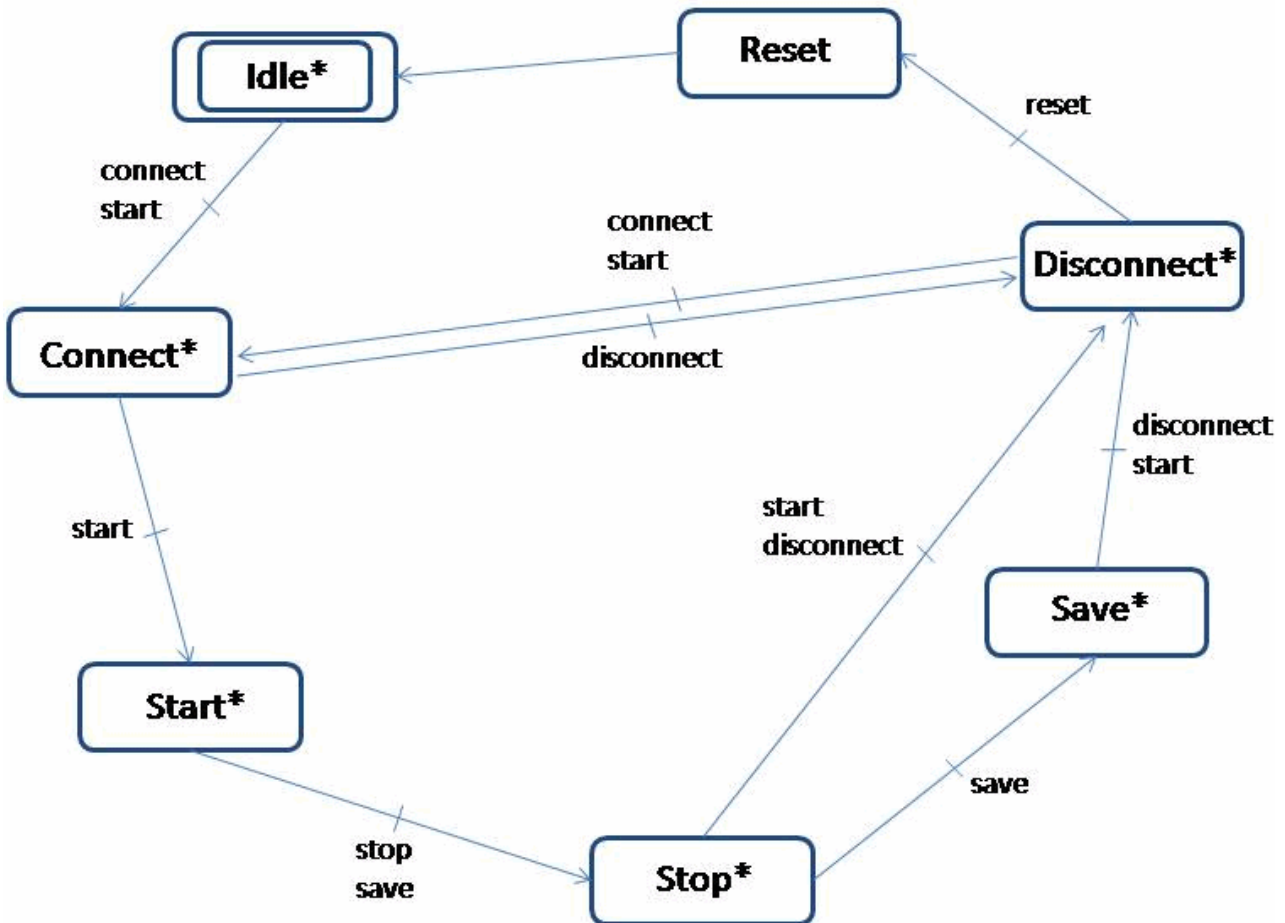
bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der bBusy-Ausgang zurückgesetzt wurde.

nErrorId: Zeigt die Fehlernummer, wenn der Fehlerausgang bError gesetzt ist. Grundsätzlich kann dies eine ADS-Fehlernummer [► 360] oder ein spezifischer Fehler-Code [► 360] dieser Bibliothek sein.



Es ist nur ein Zielsystem für die Konfigurationsdatei (*.sv2) zugelassen.

Status-Diagramm:



*) eine Statusänderung zurückzusetzen ist jederzeit möglich

Dieses Status-Diagramm zeigt die möglichen Übergänge für eReqState. Falls ein anderer Statusübergang angefordert wird, wird bError gesetzt.

TYPE E_ScopeServerState

```

TYPE E_ScopeServerState
(
  SCOPE_SERVER_IDLE,
  SCOPE_SERVER_CONNECT,
  SCOPE_SERVER_START,
  SCOPE_SERVER_STOP,
  SCOPE_SERVER_SAVE,
  SCOPE_SERVER_DISCONNECT,
  SCOPE_SERVER_RESET
);
  
```

Beispiel:

Deklarationsteil:

```

FUNCTION_BLOCK FB_ScopeServerSample
VAR_INPUT
    bExternalTriggerEvent: BOOL := FALSE;
END_VAR
VAR_OUTPUT
END_VAR
VAR
    fbScopeServerControl: FB_ScopeServerControl;
    eReqState: E_ScopeServerState := SCOPE_SERVER_IDLE;
    bBusy: BOOL := FALSE;
    bDone: BOOL := FALSE;
    bError: BOOL := FALSE;
    nErrorId: UDINT := 0;
    fbTimer: TON;
    bTriggerTimer: BOOL := FALSE;
    nState: UDINT := 0;
END_VAR
    
```

Implementierung des FB_ScopeServerSample

```

CASE nState OF
0:
    eReqState := SCOPE_SERVER_START;
    nState := 10;
10:
    IF fbScopeServerControl.bDone AND bExternalTriggerEvent
    THEN
        bTriggerTimer := TRUE;
        nState := 20;
    END_IF
20:
    IF fbTimer.Q THEN
        eReqState := SCOPE_SERVER_SAVE;
        bTriggerTimer := FALSE;
        nState := 30;
    END_IF
30:
    IF fbScopeServerControl.bDone THEN
        eReqState := SCOPE_SERVER_DISCONNECT;
    END_IF
END_CASE
fbTimer(IN:=bTriggerTimer, PT:=t#10s);
fbScopeServerControl( sNetId:= '',
    eReqState:= eReqState,
    sConfigFile:= 'C:\twinCat\scope\test.sv2',
    sSaveFile:= 'C:\twinCat\scope\test.svd',
    tTimeout:= t#5s,
    bBusy=>bBusy,
    bDone=>bDone,
    bError=>bError,
    nErrorId=>nErrorId);
    
```

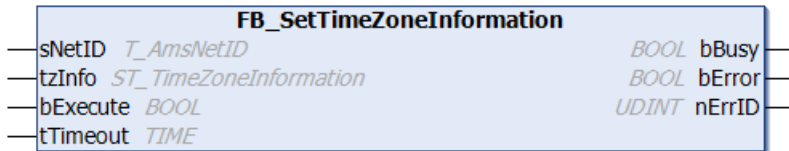
Dieses Beispiel soll zeigen, wie eine Langzeitaufzeichnung mit dem Scope Server durchgeführt werden kann.

Dafür wird eine existierende Konfiguration (Test.sv2) geladen. In diesem Beispiel wurde Test.sv2 gespeichert um im Ring-Buffer zu laufen. So wird die Datenaufzeichnung nicht enden bis sie von FB_ScopeServerControl angesteuert wird. Falls ein internes Triggerereignis (dies kann ein Fehlerereignis sein) auftritt, wird ein Timer gestartet und 10 Sekunden später werden die Daten in Test.svd gespeichert. Auf diese Weise beinhaltet die Daten-Datei Informationen vor und nach dem Trigger-Ereignis.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uutilities (System)

3.56 FB_SetTimeZoneInformation



Mit dem Funktionsbaustein können die Zeitzone-Einstellungen des Betriebssystems geändert bzw. gesetzt werden.

Uhrzeiteinstellungen

i Das Betriebssystem ändert teilweise die Uhrzeiteinstellungen nachdem die neuen Zeitzone-Einstellungen gesetzt wurden.

Die Uhrzeit kann mit dem Funktionsbaustein [NT_SetLocalTime](#) [► 124] neu eingestellt werden.

VAR_INPUT

```
VAR_INPUT
  sNetID      : T_AmsNetID;
  tzInfo      : ST_TimeZoneInformation;
  bExecute    : BOOL;
  tTimeout    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Zeitzone-Konfiguration geändert werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

tzInfo: Struktur mit den neuen Zeitzone-Einstellungen, die gesetzt werden sollen (Typ: [ST_TimeZoneInformation](#) [► 321]).

bExecute: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

tTimeout: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  bBusy       : BOOL;
  bError      : BOOL;
  nErrId      : UDINT;
END_VAR
```

bBusy: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

bError: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der *bBusy*-Ausgang zurückgesetzt wurde.

nErrId: Liefert bei einem gesetzten *bError*-Ausgang die [ADS-Fehlernummer](#) [► 360].

Beispiel:

Auf dem lokalen TwinCAT System soll die Zeitzone: "West Europa Standard Time" eingestellt werden. Als Beispiel wurde in der SPS-Bibliothek bereits eine Konstante: **WEST_EUROPE_TZI** mit den passenden Parameterwerten deklariert. Um andere Zeitzone konfigurieren zu können muss der *tzInfo*-Eingang des Funktionsbausteins mit entsprechenden Werten belegt werden (siehe in der Beschreibung der [ST_TimeZoneInformation](#) [► 321]-Struktur).

```
VAR_GLOBAL CONSTANT
...
(* West Europa Standard Time Zone settings *)
WEST_EUROPE_TZI : ST_TimeZoneInformation := (bias:=-60,
  standardName:='W. Europe Standard Time',
  standardDate:=(wYear:=0,wMonth:=10,wDayOfWeek:=0,wDay:=5,wHour:=3),
  standardBias:=0,
  daylightName:='W. Europe Daylight Time',
```

```

        daylightDate:=(wYear:=0,wMonth:=3,wDayOfWeek:=0,wDay:=5,wHour:=2),
        daylightBias:=-60);
...
END_VAR
    
```

Der Deklarationsteil:

```

PROGRAM MAIN
VAR
    fbGet    : FB_GetTimeZoneInformation;
    fbSet    : FB_SetTimeZoneInformation;
    tzi_get  : ST_TimeZoneInformation;
    tzID     : E_TimeZoneID;
    bGet     : BOOL := TRUE;
    bSet     : BOOL := FALSE;
END_VAR
    
```

Bei einer steigenden Flanke an der *bSet* -Variablen wird die gewünschte Zeitzone-Einstellung gesetzt. Zur Kontrolle können die aktuellen Einstellungen mit einer steigenden Flanke an der *bGet*-Variablen ausgelesen werden.

```

IF bGet THEN
    bGet := FALSE;
    fbGet(bExecute := TRUE);
ELSE
    fbGet(bExecute := FALSE, tzInfo => tzi_get, tzID => tzID );
END_IF

IF bSet THEN
    bSet := FALSE;
    fbSet( bExecute := TRUE, tzInfo := WEST_EUROPE_TZI );
ELSE
    fbSet( bExecute := FALSE );
END_IF
    
```

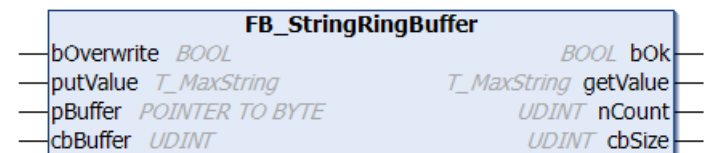
Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

- [FB_TzSpecificLocalTimeToSystemTime \[► 117\]](#)
- [FB_TzSpecificLocalTimeToFileTime \[► 29\]](#)
- [FB_SystemTimeToTzSpecificLocalTime \[► 113\]](#)
- [FB_FileTimeTimeToTzSpecificLocalTime \[► 27\]](#)
- [FB_GetTimeZoneInformation \[► 78\]](#)
- [NT_SetLocalTime \[► 124\]](#)
- [NT_GetTime \[► 122\]](#)
- [NT_SetTimeToRTCTime \[► 125\]](#)
- [F_TranslateFileTimeBias \[► 243\]](#)
- [FB_LocalSystemTime \[► 90\]](#)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Uilities (System)

3.57 FB_StringRingBuffer



Mit dem Funktionsbaustein FB_StringRingBuffer können String-Variablen in einen Ringpuffer geschrieben oder die vorher geschriebenen String-Variablen aus dem Ringpuffer ausgelesen werden. Die geschriebenen Strings werden nach dem FIFO-Prinzip in der gleichen Reihenfolge ausgelesen in der sie vorher in den Ringpuffer geschrieben wurden. D.h. beim Lesen werden zuerst die ältesten Einträge ausgelesen. Der Pufferspeicher wird dem Funktionsbaustein über die *pBuffer* / *cbBuffer*-Eingangsvariablen zur Verfügung gestellt. Das Schreiben/Lesen der Strings wird durch Aktionsaufrufe gesteuert. Der Funktionsbaustein besitzt folgende Aktionen:

- **A_AddTail** (Schreibt einen neuen String in den Ringpuffer.)
- **A_GetHead** (Liest den ältesten String aus dem Ringpuffer, entfernt ihn aber nicht.)
- **A_RemoveHead** (Liest und entfernt den ältesten String aus dem Ringpuffer.)
- **A_Reset** (Löscht alle Strings im Ringpuffer.)

VAR_INPUT

```
VAR_INPUT
  bOverwrite : BOOL;
  putValue   : T_MaxString := '';
  pBuffer    : POINTER TO BYTE;
  cbBuffer   : UDINT;
END_VAR
```

bOverwrite : Beim TRUE und Pufferüberlauf werden die ältesten Einträge überschrieben. Beim FALSE wird beim Pufferüberlauf ein Fehler gemeldet (bOk = FALSE).

putValue : String der in den Ringpuffer geschrieben werden soll (Typ: T_MaxString).

pBuffer: Adresse einer SPS-Variablen (z.B. ARRAY[...] OF BYTES) die vom Funktionsbaustein als Pufferspeicher benutzt werden soll. Die Adresse kann mit dem ADR-Operator ermittelt werden.

cbBuffer: Max. Bytegröße der SPS-Variablen die als Pufferspeicher benutzt werden soll. Die Größe kann mit dem SIZEOF-Operator ermittelt werden.

VAR_OUTPUT

```
VAR_OUTPUT
  bOk       : BOOL;
  getValue  : T_MaxString := '';
  nCount    : UDINT;
  cbSize    : UDINT;
END_VAR
```

bOk: Liefert TRUE wenn ein neuer String erfolgreich hinzugefügt oder entfernt werden konnte und FALSE beim Puffer-Überlauf oder wenn keine Einträge im Puffer mehr vorhanden sind.

getValue: Dieser Ausgang liefert den String, der zuletzt aus dem Ringpuffer gelesen wurde (Typ: T_MaxString).

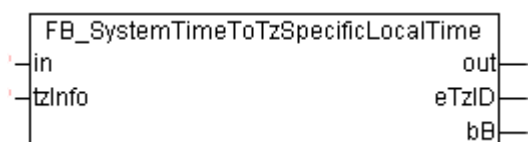
nCount: Liefert die aktuelle Anzahl der gepufferten Strings.

cbSize: Liefert die aktuelle Anzahl der belegten Datenbytes im Puffer. Die Anzahl der belegten Datenbytes ist immer größer als die tatsächliche Anzahl der geschriebenen Value-Daten. Jeder String wird um zusätzliche Informationen ergänzt um ihn später lokalisieren zu können.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.58 FB_SystemTimeToTzSpecificLocalTime



Der Funktionsbaustein konvertiert die UTC-Zeit (structured system time format) in Lokalzeit (structured system time format) unter der Berücksichtigung der angegebenen Zeitzoneinformation. Der Funktionsbaustein: [FB_FileTimeToTzSpecificLocalTime \[► 27\]](#) besitzt eine ähnliche Funktionalität, mit dem Unterschied dass er ein anderes Zeitformat (file time format) umrechnet.

Der Baustein eignet sich nur zur Konvertierung von **kontinuierlichen** UTC-Zeitstempelinformation. Anhand der Zeitzoneinformation errechnet der Funktionsbaustein die erforderlichen Zeitsprünge in der Lokalzeit (Sommer-/Winterzeit Umstellung). Zeitsprünge in der UTC-Eingangszeit sind nicht zulässig führen zur fehlerhaften Konvertierung. Der Grund: Die zuletzt konvertierte Zeit wird im Baustein intern gespeichert um aus dem Verlauf der UTC-Eingangszeit und dem gespeicherten Wert die B-Zeiten (siehe unten) beim Zurückstellen der Lokalzeit erkennen zu können.

Der Baustein besitzt eine Aktion: `A_Reset()`. Ein Aufruf dieser Aktion setzt die Ausgänge des Bausteins und die intern gespeicherte (zuletzt konvertierte Zeit) auf Null zurück.

VAR_INPUT

```
VAR_INPUT
  in      : TIMESTRUCT;
  tzInfo  : ST_TimeZoneInformation;
END_VAR
```

in: UTC-Zeit (structured system time format) die konvertiert werden soll (Typ: [TIMESTRUCT \[► 327\]](#)).

tzInfo: Strukturvariable mit der aktuellen Zeitzone-Information des Betriebssystems (Typ: [ST_TimeZoneInformation \[► 321\]](#)).

VAR_OUTPUT

```
VAR_OUTPUT
  out      : TIMESTRUCT;
  eTzID    : E_TimeZoneID := eTimeZoneID_Unknown;
  bB       : BOOL;
END_VAR
```

out: Konvertierte Lokalzeit (structured system time format) (Typ: [TIMESTRUCT \[► 327\]](#)).

eTzID: Zusätzliche Sommer-/ Winterzeit-Information (Typ: [E_TimeZoneID \[► 308\]](#)).

bB: TRUE => B-Zeit (Z.B.: **02:05:00 MEZ B**), FALSE => Übrige Zeit (Z.B.: **02:05:00 MESZ A**). Bei einem Sprung der Lokalzeit zurück wird dieser Ausgang gesetzt und beim Überschreiten der doppelten Lokalzeit wieder zurückgesetzt.

Beispiel:

```
PROGRAM MAIN
VAR
  in      : TIMESTRUCT := ( wYear := 2011, wMonth := 4, wDay := 29, wHour := 14, wMinute := 46,
wSecond := 31, wMilliseconds := 99 ); (* UTC time *)
  out     : TIMESTRUCT; (* Local time result is:= ( wYear := 2011, wMonth := 4, wDay := 29, wHour := 16, wMinute := 46, wSecond := 31, wMilliseconds := 99 ) *)
  fbToLocal : FB_SystemTimeToTzSpecificLocalTime;
END_VAR

fbToLocal( in := in, tzInfo := WEST_EUROPE_TZI, out => out );
```

Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

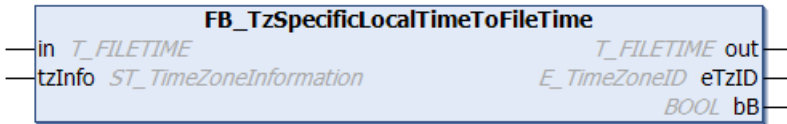
- [FB_TzSpecificLocalTimeToSystemTime \[► 117\]](#)
- [FB_TzSpecificLocalTimeToFileTime \[► 29\]](#)
- [FB_FileTimeTimeToTzSpecificLocalTime \[► 27\]](#)
- [FB_GetTimeZoneInformation \[► 78\]](#)
- [FB_SetTimeZoneInformation \[► 111\]](#)
- [NT_SetLocalTime \[► 124\]](#)
- [NT_GetTime \[► 122\]](#)
- [NT_SetTimeToRTCTime \[► 125\]](#)

- [F_TranslateFileTimeBias](#) [▶ 243]
- [FB_LocalSystemTime](#) [▶ 90]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.59 FB_TzSpecificLocalTimeToFileTime64



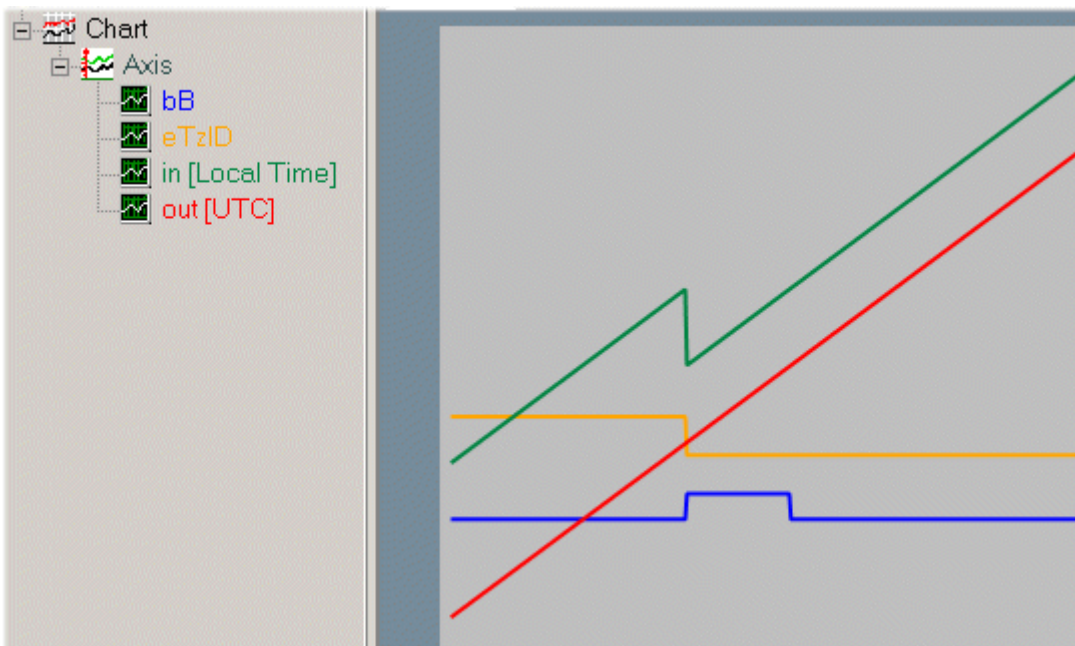
Der Funktionsbaustein konvertiert die Lokalzeit (file time format) in UTC-Zeit (file time format) unter der Berücksichtigung der angegebenen Zeitzoneinformation. Der Funktionsbaustein:

[FB_TzSpecificLocalTimeToSystemTime](#) [▶ 117] besitzt eine ähnliche Funktionalität, mit dem Unterschied dass er ein anderes Zeitformat (structured system time format) umrechnet.

Der Baustein eignet sich nur zur Konvertierung von **kontinuierlichen** Lokalzeit-Zeitstempelinformation. Die Lokalzeitsprünge durch die Sommer-/Winterzeit Umstellung sind zulässig und werden vom Baustein erkannt. Lokalzeiten, die sich beliebig ändern führen zur fehlerhaften Konvertierung. Der Grund: Die zuletzt konvertierte Zeit wird im Baustein intern gespeichert um die Sommerzeit-/Winterzeit-Information und die B-Zeiten (siehe unten) beim Zurückstellen der Lokalzeit erkennen zu können. Der Baustein besitzt eine Aktion: A_Reset(). Ein Aufruf dieser Aktion setzt die Ausgänge des Bausteins und die intern gespeicherte (zuletzt konvertierte Zeit) auf Null zurück.

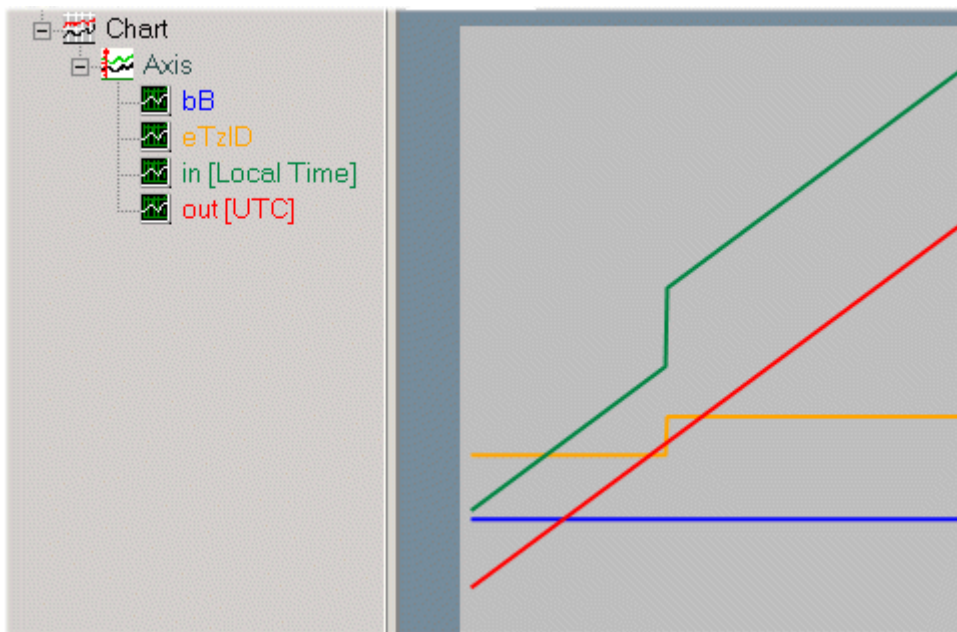
Die Sprünge in der Lokalzeit stellen hier ein Problem dar da sie in eine lineare UTC-Zeit konvertiert werden müssen. Es empfiehlt sich daher für Zeitstempelungsaufgaben die (kontinuierliche) UTC-Zeit zu verwenden und diese erst zur optischen Anzeige der Werte (z.B. in einer Visualisierung) in die jeweilige Lokalzeit zu konvertieren.

1. Grafische Darstellung des zeitlichen Verhaltens beim Übergang von Sommerzeit zur Winterzeit (hier tzInfo = WEST_EUROPE_TZI):



Die Lokalzeit (grün) macht einen Sprung zurück. Die UTC-Ausgangszeit (rot) verläuft kontinuierlich weiter. Der Lokalzeit: **02h:59m:59s:999ms..** folgt unmittelbar die Zeit: **02h:00m:00s:000ms..** Die Zeiten zwischen 2h und 3h sind zweifach vorhanden. Die doppelte Zeit vor der Zeitumstellung wird z.B. als **02:05:00 MESZ A** bezeichnet und die Zeit nach der Umstellung als **02:05:00 MEZ B**. Die Ausgangsvariable *bB* gibt Information darüber ob es sich um den ersten oder zweiten *Durchgang* handelt. Beim zweiten *Durchgang* ist die *bB*-Ausgangsvariable (blau) auf TRUE gesetzt. Die *bB*-Ausgangsvariable wird automatisch zurückgesetzt nach dem die doppelte Zeit überschritten wurde. Die Zeitzonen-ID (orange) wechselt von *eTimeZoneID_Daylight* (Sommerzeit) zu *eTimeZoneID_Standard* (Winterzeit).

2. Grafische Darstellung des zeitlichen Verhaltens beim Übergang von Winterzeit zur Sommerzeit (hier tzInfo = WEST_EUROPE_TZI):



Die Lokalzeit (grün) macht einen Sprung nach Vorne. Die UTC-Ausgangszeit (rot) verläuft kontinuierlich weiter. Der Lokalzeit: **2h:59m:59s:999ms..** folgt unmittelbar die Zeit: **3h:00m:00s:000ms..** Die Zeitzonen-ID (orange) wechselt von *eTimeZoneID_Standard* (Winterzeit) zu *eTimeZoneID_Daylight* (Sommerzeit).

VAR_INPUT

```
VAR_INPUT
  in      : T_FILETIME64;
  tzInfo  : ST_TimeZoneInformation;
END_VAR
```

in: Lokalzeit (file time format), die konvertiert werden soll (Typ: [T_FILETIME](#) [► 323]).

tzInfo: Strukturvariable mit der aktuellen Zeitzonen-Information des Betriebssystems (Typ: [ST_TimeZoneInformation](#) [► 321]).

VAR_OUTPUT

```
VAR_OUTPUT
  out     : T_FILETIME64;
  eTzID   : E_TimeZoneID := eTimeZoneID_Unknown;
  bB      : BOOL;
END_VAR
```

out: Konvertierte UTC-Zeit (file time format) (Typ: [T_FILETIME](#) [► 323]).

eTzID: Zusätzliche Sommer-/ Winterzeit-Information (Typ: [E_TimeZoneID](#) [► 308]).

bB: TRUE => B-Zeit (Z.B.: **02:05:00 MEZ B**), FALSE => Übrige Zeit (Z.B.: **02:05:00 MESZ A**). Bei einem Sprung der Lokalzeit zurück wird dieser Ausgang gesetzt und beim Überschreiten der doppelten Lokalzeit wieder zurückgesetzt.

Beispiel:

Die Lokalzeit: DT#2011-09-02-11:01:31 wird in UTC-Zeit: DT#2011-09-02-09:01:31 konvertiert.

```
PROGRAM MAIN
VAR
  in      : DT := DT#2011-09-02-11:01:31; (* Local time *)
  out     : DT; (* UTC time *)
  fbToUTC : FB_TzSpecificLocalTimeToFileTime64;
END_VAR

fbToUTC( in := DT_TO_FILETIME64( in ), tzInfo := WEST_EUROPE_TZI );
out := FILETIME64_TO_DT( fbToUTC.out );
```

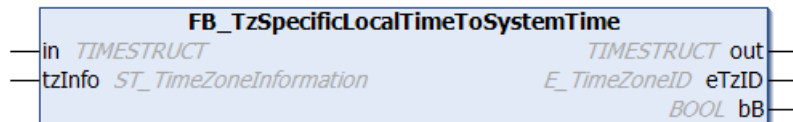
Weitere Zeit-, Zeitzonen-Funktionen und -Funktionsbausteine:

- [FB_TzSpecificLocalTimeToSystemTime \[► 117\]](#)
- [FB_SystemTimeToTzSpecificLocalTime \[► 113\]](#)
- [FB_FileTime64ToTzSpecificLocalTime \[► 62\]](#)
- [FB_GetTimeZoneInformation \[► 78\]](#)
- [FB_SetTimeZoneInformation \[► 111\]](#)
- [NT_SetLocalTime \[► 124\]](#)
- [NT_GetTime \[► 122\]](#)
- [NT_SetTimeToRTCTime \[► 125\]](#)
- [F_TranslateFileTime64Bias \[► 154\]](#)
- [FB_LocalSystemTime \[► 90\]](#)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.44.0

3.60 FB_TzSpecificLocalTimeToSystemTime



Der Funktionsbaustein konvertiert die Lokalzeit (structured system time format) in UTC-Zeit (structured system time format) unter der Berücksichtigung der angegebenen Zeitzoneinformation. Der Funktionsbaustein: [FB_TzSpecificLocalTimeToFileTime \[► 29\]](#) besitzt eine ähnliche Funktionalität, mit dem Unterschied dass er ein anderes Zeitformat (file time format) umrechnet.

Der Baustein eignet sich nur zur Konvertierung von **kontinuierlichen** Lokalzeit-Zeitstempelinformationen. Die Lokalzeitsprünge durch die Sommer-/Winterzeit Umstellung sind zulässig und werden vom Baustein erkannt. Lokalzeiten, die sich beliebig ändern führen zur fehlerhaften Konvertierung. Der Grund: Die zuletzt konvertierte Zeit wird im Baustein intern gespeichert um die Sommerzeit-/Winterzeit-Information und die B-Zeiten (siehe unten) beim Zurückstellen der Lokalzeit erkennen zu können. Der Baustein besitzt eine Aktion: `A_Reset()`. Ein Aufruf dieser Aktion setzt die Ausgänge des Bausteins und die intern gespeicherte (zuletzt konvertierte Zeit) auf Null zurück.

Die Sprünge in der Lokalzeit stellen hier ein Problem dar da sie in eine lineare UTC-Zeit konvertiert werden müssen. Es empfiehlt sich daher für Zeitstempelungsaufgaben die (kontinuierliche) UTC-Zeit zu verwenden und diese erst zur optischen Anzeige der Werte (z.B. in einer Visualisierung) in die jeweilige Lokalzeit zu konvertieren.

Weitere Informationen finden Sie in der Dokumentation des [FB_TzSpecificLocalTimeToFileTime \[► 29\]](#) Funktionsbausteins.

VAR_INPUT

```
VAR_INPUT
  in      : Timestruct;
  tzInfo  : ST_TimeZoneInformation;
END_VAR
```

in: Lokalzeit (structured system time format) die konvertiert werden soll (Typ: [Timestruct](#) [[▶ 327](#)]).

tzInfo: Strukturvariable mit der aktuellen Zeitzone-Information des Betriebssystems (Typ: [ST_TimeZoneInformation](#) [[▶ 321](#)]).

VAR_OUTPUT

```
VAR_OUTPUT
  out     : Timestruct;
  eTzID   : E_TimeZoneID := eTimeZoneID_Unknown;
  bB      : BOOL;
END_VAR
```

out: Konvertierte UTC-Zeit (structured system time format) (Typ: [Timestruct](#) [[▶ 327](#)]).

eTzID: Zusätzliche Sommer-/ Winterzeit-Information (Typ: [E_TimeZoneID](#) [[▶ 308](#)]).

bB: TRUE => B-Zeit (Z.B.: **02:05:00 MEZ B**), FALSE => Übrige Zeit (Z.B.: **02:05:00 MESZ A**). Bei einem Sprung der Lokalzeit zurück wird dieser Ausgang gesetzt und beim Überschreiten der doppelten Lokalzeit wieder zurückgesetzt.

Beispiel:

```
PROGRAM MAIN
VAR
  in      : Timestruct := ( wYear := 2011, wMonth := 4, wDay := 29, wHour := 16, wMinute := 46, wSecond := 31, wMilliseconds := 99 ); (* Local time *)
  out     : Timestruct;
  (* UTC time result is:= ( wYear := 2011, wMonth := 4, wDay := 29, wHour := 14, wMinute := 46, wSecond := 31, wMilliseconds := 99 ) *)
  fbToUTC : FB_TzSpecificLocalTimeToSystemTime;
END_VAR

fbToUTC( in := in, tzInfo := WEST_EUROPE_TZI, out => out );
```

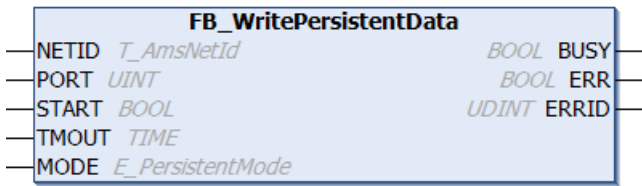
Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

- [FB_TzSpecificLocalTimeToFileTime](#) [[▶ 29](#)]
- [FB_SystemTimeToTzSpecificLocalTime](#) [[▶ 113](#)]
- [FB_FileTimeTimeToTzSpecificLocalTime](#) [[▶ 27](#)]
- [FB_GetTimeZoneInformation](#) [[▶ 78](#)]
- [FB_SetTimeZoneInformation](#) [[▶ 111](#)]
- [NT_SetLocalTime](#) [[▶ 124](#)]
- [NT_GetTime](#) [[▶ 122](#)]
- [NT_SetTimeToRTCTime](#) [[▶ 125](#)]
- [F_TranslateFileTimeBias](#) [[▶ 243](#)]
- [FB_LocalSystemTime](#) [[▶ 90](#)]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.61 FB_WritePersistentData



Bei dem Funktionsbaustein FB_WritePersistentData handelt es sich um eine erweiterte Version des WritePersistentData [► 146]-Funktionsbausteins. Über den MODE-Parameter kann aber das Systemverhalten beim Schreiben der pers. Daten [► 356] beeinflusst werden (Datenkonsistenz/Taskzykluszeitüberschreitung).

VAR_INPUT

```
VAR_INPUT
    NETID    : T_AmsNetId;
    PORT     : UINT;
    START    : BOOL;
    TMOUT    : TIME := DEFAULT_ADS_TIMEOUT;
    MODE     : E_PersistentMode;
END_VAR
```

NETID : Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen persistente Daten gespeichert werden sollen (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

PORT : Der PORT-Parameter bestimmt das Laufzeitsystem dessen persistente Daten gespeichert werden sollen.

START : Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT : Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

MODE : Modus in dem die pers. Daten geschrieben werden sollen (Typ: E_PersistentMode [► 306]).

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY     : BOOL;
    ERR      : BOOL;
    ERRID    : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [► 360].

Beispiel:

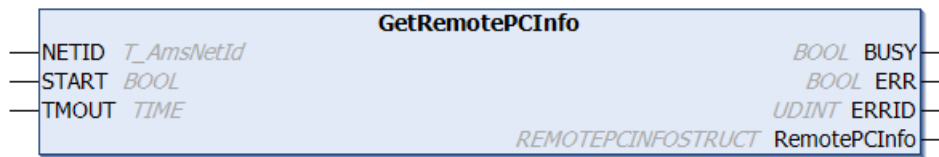
Siehe: Schreiben der pers. Daten: Systemverhalten [► 356].

Siehe: Beispiel in der Dokumentation des WritePersistentData-Funktionsbausteins [► 146].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.62 GetRemotePCInfo



Mit dem Funktionsbaustein "GetRemotePCInfo" können Informationen über konfigurierte Remote-PCs im TwinCAT-Router ausgelesen werden. Nach einer erfolgreichen Ausführung sind in der Struktur "RemotePCInfo" die NetIds und Namen der Remote-PCs als Strings abgelegt, in der Reihenfolge, wie sie im TwinCAT - Router abgelegt wurden. Mit dem Funktionsbaustein können Router-Informationen des lokalen oder eines Remote TwinCAT-Systems ausgelesen werden.

VAR_INPUT

```
VAR_INPUT
    NETID    : T_AmsNetId;
    START    : BOOL;
    TMOUT    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Router-Informationen über konfigurierte Remote-PCs ausgelesen werden soll (Typ: T_AmsNetID). Sollen die Remote-PCs des lokalen TwinCAT-Systems ermittelt werden, kann auch ein Leerstring angegeben werden.

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY      : BOOL;
    ERR       : BOOL;
    ERRID     : UDINT;
    RemotePCInfo : REMOTEPCINFOSTRUCT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

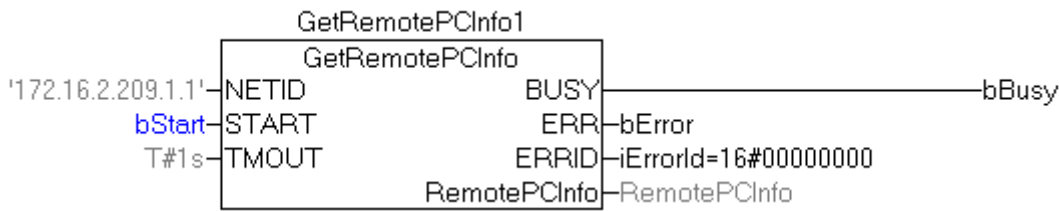
ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die [ADS-Fehlernummer](#) [[▶ 360](#)].

RemotePCInfoStruct: Struktur mit Informationen zu den konfigurierten Remote-PCs (Typ: REMOTEPCINFOSTRUCT [[▶ 311](#)]).

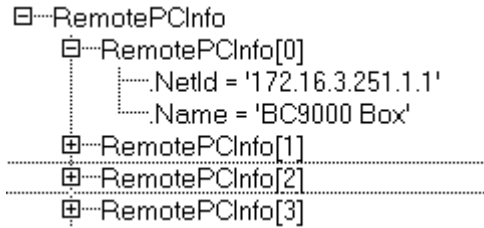
Beispiel:

```
PROGRAM MAIN
VAR
    GetRemotePCInfo1 : GetRemotePCInfo;
    RemotePCInfo     : REMOTEPCINFOSTRUCT;
    bBusy            : BOOL;
    bError           : BOOL;
    iErrorId        : UDINT;
    bStart           : BOOL;
END_VAR
```



Online-Ansicht:

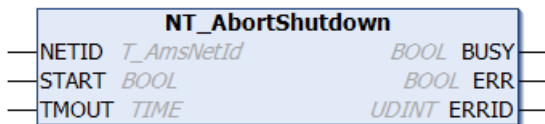
Die NetIds und Namen der konfigurierten Remote-PCs.



Voraussetzungen

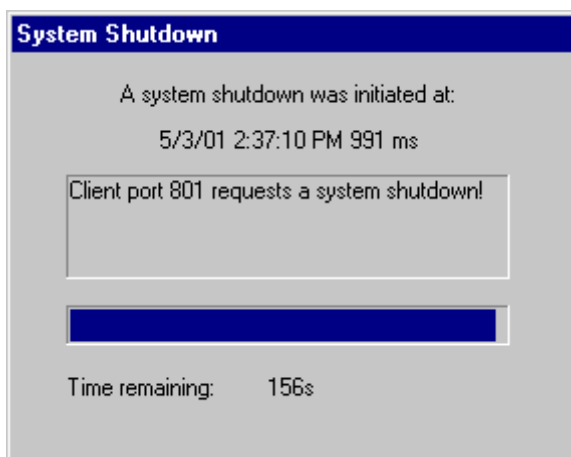
Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.63 NT_AbortShutdown



Mit dem Funktionsbaustein "NT_AbortShutdown" kann ein zuvor mit dem [NT_Shutdown](#) [127]-Funktionsbaustein aufgerufenes Shutdown-Kommando abgebrochen werden.

Beim Aufruf des [NT_Shutdown](#) [127]-Funktionsbausteines kann eine Verzögerungszeit als Parameter angegeben werden. Die verbleibende Zeit wird in einem Meldungsfenster angezeigt:



Erst nachdem die Verzögerungszeit abgelaufen ist, wird ein Shutdown des Betriebssystems durchgeführt. In dieser Zeit kann der Shutdown-Prozess mit dem Funktionsbaustein "NT_AbortShutdown" aus der SPS unterbrochen werden.

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  START   : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem der Shutdown-Prozess abgebrochen werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY    : BOOL;
  ERR     : BOOL;
  ERRID   : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

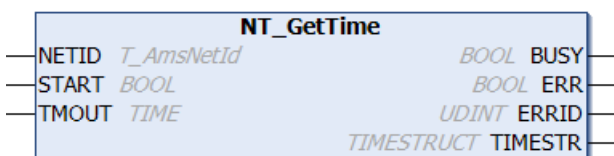
ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [► 360].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Uilities (System)

3.64 NT_GetTime



Mit dem Funktionsbaustein "NT_GetTime" kann die lokale Windows-Systemzeit eines TwinCAT-Systems ermittelt werden (die lokale Windows-Systemzeit wird in der Taskleiste eingeblendet). Das Jahr, Monat, Tag, Wochentag, Stunde, Minute, Sekunde und Millisekunde werden in den Variablen der Struktur TIMESTRUCT [► 327] abgelegt.

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  START   : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen lokale Windows-Systemzeit ermittelt werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY      : BOOL;
  ERR       : BOOL;
  ERRID     : UDINT;
  TIMESTR   : TIMESTRUCT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [▶ 360].

TIMESTR: Struktur mit der lokalen Windows-Systemzeit (Typ: TIMESTRUCT [▶ 327]).

Beispiel:

Siehe: Beispiel: Software-Uhren (RTC, RTC_EX, RTC_EX2) [▶ 354].

Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

- [FB_TzSpecificLocalTimeToSystemTime](#) [▶ 117]
- [FB_TzSpecificLocalTimeToFileTime](#) [▶ 29]
- [FB_SystemTimeToTzSpecificLocalTime](#) [▶ 113]
- [FB_FileTimeTimeToTzSpecificLocalTime](#) [▶ 27]
- [FB_GetTimeZoneInformation](#) [▶ 78]
- [FB_SetTimeZoneInformation](#) [▶ 111]
- [NT_SetLocalTime](#) [▶ 124]
- [NT_SetTimeToRTCTime](#) [▶ 125]
- [F_TranslateFileTimeBias](#) [▶ 243]
- [FB_LocalSystemTime](#) [▶ 90]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Uilities (System)

3.65 NT_Reboot



Mit dem Funktionsbaustein "NT_Reboot" kann ein Restart des Windows NT Betriebssystems durchgeführt werden. Die Funktion entspricht in etwa dem Restart-Befehl in der Windows-Startleiste. Über den Parameter DELAY kann eine Verzögerungszeit für die Ausführung des Restart-Befehls angegeben werden.

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  DELAY   : DWORD;
  START   : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Restart durchgeführt werden soll (Typ: T_AmsNetID). Für einen Restart des lokalen Rechners, kann auch ein Leerstring angegeben werden.

DELAY: Verzögerungszeit für die Ausführung des Restart-Befehls in Sekunden.

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY    : BOOL;
  ERR     : BOOL;
  ERRID   : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [► 360].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Uilities (System)

3.66 NT_SetLocalTime



Mit dem Funktionsbaustein "NT_SetLocalTime" kann die lokale Windows-Systemzeit eines TwinCAT-Systems gesetzt werden (die lokale Windows-Systemzeit wird in der Taskleiste eingeblendet).

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  START   : BOOL;
  TIMESTR : TIMESTRUCT;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann die Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen lokale Windows-Systemzeit gesetzt werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TIMESTR: Struktur mit der neuen lokalen Windows-Systemzeit (Typ: [TIMESTRUCT](#) [[▶ 327](#)]).

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY : BOOL;
  ERR  : BOOL;
  ERRID : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die [ADS-Fehlernummer](#) [[▶ 360](#)].

Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

- [FB_TzSpecificLocalTimeToSystemTime](#) [[▶ 117](#)]
- [FB_TzSpecificLocalTimeToFileTime](#) [[▶ 29](#)]
- [FB_SystemTimeToTzSpecificLocalTime](#) [[▶ 113](#)]
- [FB_FileTimeTimeToTzSpecificLocalTime](#) [[▶ 27](#)]
- [FB_GetTimeZoneInformation](#) [[▶ 78](#)]
- [FB_SetTimeZoneInformation](#) [[▶ 111](#)]
- [NT_GetTime](#) [[▶ 122](#)]
- [NT_SetTimeToRTCTime](#) [[▶ 125](#)]
- [F_TranslateFileTimeBias](#) [[▶ 243](#)]
- [FB_LocalSystemTime](#) [[▶ 90](#)]

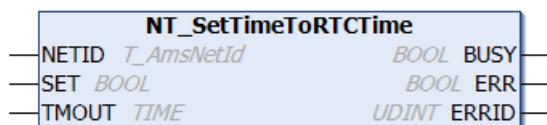
Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Uilities (System)

3.67 NT_SetTimeToRTCTime



Diese Funktionalität ist in dem SPS-Laufzeitsystem unter Windows CE nicht verfügbar!



Mit dem Funktionsbaustein "NT_SetTimeToRTCTime" kann die lokale Windows-Systemzeit (wird in der Taskleiste eingeblendet) mit der Echtzeituhr des PCs (RTC-Zeit in BIOS) synchronisiert werden.

Bemerkungen

Beim Aufruf des Funktionsbausteins wird die Echtzeituhr des TwinCAT PCs mit der lokalen Windows-Systemzeit verglichen und die lokale Windows-Systemzeit um die ermittelte Differenz korrigiert. Zeitzonen und Sommerzeit werden dabei berücksichtigt. Zu beachten ist, dass es durch die Korrektur zu Zeitsprüngen während Messungen oder Logbuchaufzeichnungen führen kann.

Beim Setzen der lokalen Windows-Systemzeit wird vom Betriebssystem auch automatisch die RTC-Zeit auf die neue lokale Windows-Systemzeit gesetzt. Die neue RTC-Zeit wird dabei zwangsläufig durch die Umrechnung und Verzögerung mit einem kleinen Fehler behaftet. Der Fehler liegt im Millisekundenbereich. D.h. mit jedem Aufruf von NT_SetTimeToRTCTime wird die Echtzeituhr ein wenig verfälscht. Um möglichst kleine Abweichungen über einen längeren Zeitraum zu erreichen, sollte der Abgleich z.B. alle 24 Stunden und nicht in jedem SPS-Zyklus durchgeführt werden.

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  SET     : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem die lokale Windows-Systemzeit synchronisiert werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

SET: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY    : BOOL;
  ERR     : BOOL;
  ERRID   : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die [ADS-Fehlernummer](#) [► 360].

Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

- [FB_TzSpecificLocalTimeToSystemTime](#) [► 117]
- [FB_TzSpecificLocalTimeToFileTime](#) [► 29]
- [FB_SystemTimeToTzSpecificLocalTime](#) [► 113]
- [FB_FileTimeTimeToTzSpecificLocalTime](#) [► 27]
- [FB_GetTimeZoneInformation](#) [► 78]
- [FB_SetTimeZoneInformation](#) [► 111]
- [NT_SetLocalTime](#) [► 124]
- [NT_GetTime](#) [► 122]
- [F_TranslateFileTimeBias](#) [► 243]
- [FB_LocalSystemTime](#) [► 90]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64)	Tc2_Utilities (System)

3.68 NT_Shutdown



Diese Funktionalität ist unter Windows CE nicht verfügbar!



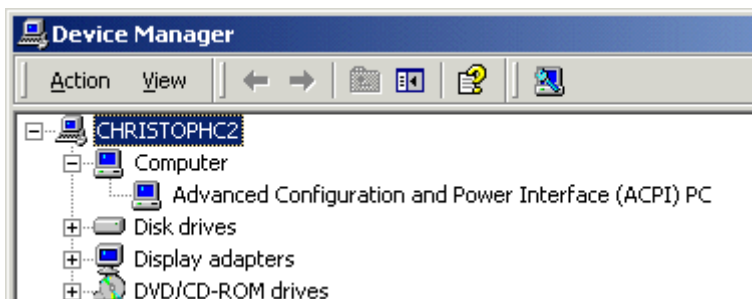
Mit dem Funktionsbaustein "NT_Shutdown" kann ein Shutdown des Windows NT Betriebssystems durchgeführt werden. Die Funktion entspricht in etwa dem Shutdown-Befehl in der Windows-Startleiste. Über den Parameter DELAY kann eine Verzögerungszeit für die Ausführung des Shutdown-Befehls angegeben werden.

Bemerkungen:

Bei den neueren Betriebssystemen (z.B. Windows 2000) wird beim Aufruf des dem NT_Shutdown-Funktionsbausteins ein Shutdown mit Power OFF ausgeführt. D.h. der PC schaltet sich nach dem Shutdown Vorgang selbsttätig ab. Diese Funktion steht aber nur dann zur Verfügung wenn von dem Betriebssystem ein ACPI-fähiger PC erkannt wurde (Advanced Configuration and Power Interface). Die ACPI-Funktionen müssen auch von dem Motherboard und dem Netzteil des PCs unterstützt werden. In den meisten Fällen muss die ACPI-Funktionsfähigkeit im BIOS vor der Installation des Betriebssystems aktiviert werden. Eine nachträgliche Änderung wird in den meisten Fällen von dem Betriebssystem nicht erkannt. Ob ein ACPI-fähiger PC erkannt wurde, kann z.B. bei Windows 2000 auf folgende Weise geprüft werden:

1. In der "Systemsteuerung" den Ordner "System" öffnen.
2. Auf dem Karteireiter "Hardware" den "Geräte Manager" auswählen.

In dem Baum mit den Geräten muss unter Computer folgendes stehen: "Advanced Configuration and Power Interface (ACPI) PC".



Beim Aufruf von NT_SHUTDOWN führt TwinCAT standardmäßig ein Shutdown mit POWER-OFF. Unterstützt ein PC diese Funktionalität nicht, dann kann diese durch einen Eintrag in der Windows-Registry deaktiviert werden. Folgender Wert muss in die Registry eingetragen werden

"DisableACPIPowerOff"REG_DWORD = 0x00000001 in Registry unter:
"HKEY_LOCAL_MACHINE\SOFTWARE\BeckhoffTwinCAT\System"

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  DELAY   : DWORD;
  START   : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem ein Shutdown des Betriebssystems durchgeführt werden soll (Typ: T_AmsNetId). Für einen Shutdown auf dem lokalen Rechner kann auch ein Leerstring angegeben werden.

DELAY: Verzögerungszeit für die Ausführung des Shutdown-Befehls in Sekunden.

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY    : BOOL;
  ERR     : BOOL;
  ERRID   : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

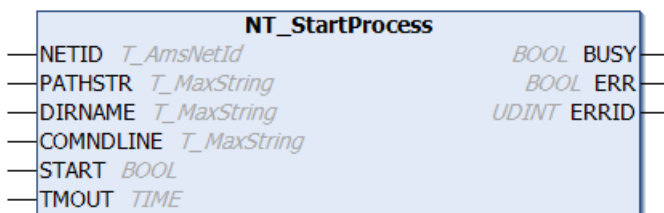
ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [► 360].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Uilities (System)

3.69 NT_StartProcess



Mit dem Funktionsbaustein "NT_StartProcess" kann aus der SPS heraus eine Windows-Anwendung gestartet werden. Mit dem Funktionsbaustein können auch Anwendungen auf einem Remote-PC gestartet werden.

VAR_INPUT

```
VAR_INPUT
  NETID       : T_AmsNetId;
  PATHSTR     : T_MaxString;
  DIRNAME     : T_MaxString;
  COMNDLINE   : T_MaxString;
  START       : BOOL;
  TMOUT       : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem die Anwendung gestartet werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

PATHSTR: Der gesamte Pfad der auszuführenden Anwendung als String (Typ: T_MaxString, z.B. "C:\WINNT\notepad.exe").

DIRNAME: Arbeitsverzeichnis der auszuführenden Anwendung als String (Typ: T_MaxString, z.B. "C:\WINNT").

COMNDLINE: Kommandozeilen-Parameter (Typ: T_MaxString, z.B.: "win.ini").

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY   : BOOL;
  ERR    : BOOL;
  ERRID  : UDINT;
END_VAR
```

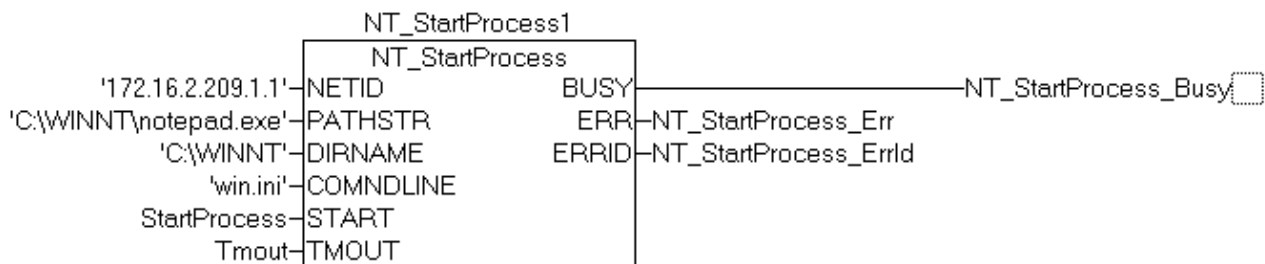
BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die [ADS-Fehlernummer \[▶ 360\]](#) oder den [Win32-Fehlercode \[▶ 364\]](#) (Plattform SDK: Win32 API).

Beispiel:

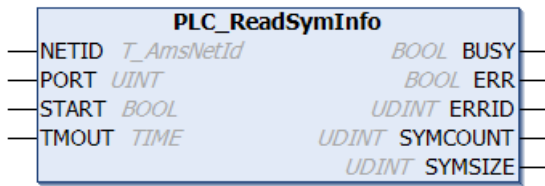
```
NT_StartProcess1 : NT_StartProcess;
NT_StartProcess_Busy : BOOL;
NT_StartProcess_Err : BOOL;
NT_StartProcess_ErrId : UDINT;
StartProcess : BOOL;
Tmout : TIME;
```



Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64)	Tc2_Utilities (System)

3.70 PLC_ReadSymInfo



Mit dem Funktionsbaustein "PLC_ReadSymInfo" können Informationen über die Symbole (Variablen) eines SPS-Laufzeitsystems ermittelt werden.

VAR_INPUT

```
VAR_INPUT
  NETID : T_AmsNetId;
  PORT  : T_AmsPort;
  START : BOOL;
  TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Symbolinformationen ermittelt werden sollen (Typ: T_AmsNetId). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

PORT: Die Portnummer eines SPS-Laufzeitsystems (Typ: T_AmsPort).

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY      : BOOL;
  ERR       : BOOL;
  ERRID     : UDINT;
  SYMCOUNT  : UDINT;
  SYMSIZE   : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [► 360].

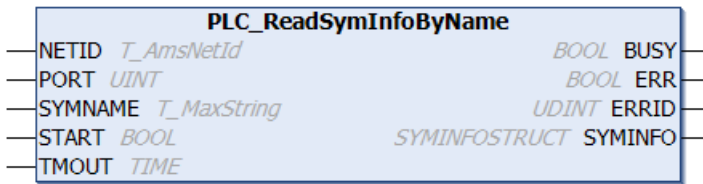
SYMCOUNT: Anzahl der Symbole in dem SPS-Laufzeitsystem.

SYMSIZE: Länge der Daten in Byte, in denen die Symbolinformationen gespeichert werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.71 PLC_ReadSymInfoByName



Mit dem Funktionsbaustein "PLC_ReadSymInfoByName" können zusätzliche Informationen über eine SPS-Symbolvariable (z.B. Datentypbezeichnung, Index-Group, Index-Offset, Kommentar...) anhand des Symbolnamens ausgelesen werden. Nach einer erfolgreichen Ausführung liegen die Daten in der Datenstruktur **SymInfo** vom Typ: SYMINFOSTRUCT zur Verfügung. Die maximale Länge von dem Symbolnamen, der an den Funktionsbaustein als Parameter übergeben wird ist auf 255 Zeichen begrenzt.

VAR_INPUT

```
VAR_INPUT
    NETID      : T_AmsNetId;
    PORT       : T_AmsPort;
    SYMNAME    : T_MaxString;
    START      : BOOL;
    TMOUT      : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem Funktion ausgeführt werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

PORT: Die Portnummer des SPS-Laufzeitsystems zu dem die Symbolvariable gehört (Typ: T_AmsPort).

SYMNAME: Der Symbolname der SPS-Variablen deren Informationen gelesen werden sollen (Typ: T_MaxString, max. 255 Zeichen, inklusive des gesamten Pfades z.B. 'MAIN.INIT_TASK.VARINT').

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY       : BOOL;
    ERR        : BOOL;
    ERRID      : UDINT;
    SymInfo    : SYMINFOSTRUCT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

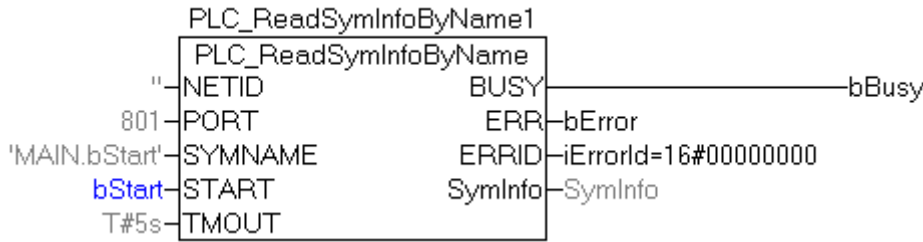
ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [► 360].

SymInfo: Struktur mit zusätzlichen Informationen zu der Symbolvariablen (Typ: SYMINFOSTRUCT [► 322]).

Beispiel:

```
PROGRAM MAIN
VAR
    PLC_ReadSymInfoByName1 : PLC_ReadSymInfoByName;
    bStart AT%X0.5         : BOOL; (*Starts FB execution*)
    bBusy                  : BOOL;
    bError                  : BOOL;
    iErrorId                : UDINT;
    SymInfo                 : SYMINFOSTRUCT;(*Structure with sombol information*)
END_VAR
```



Online-Ansicht:

```

E--SymInfo
.....
.....:symEntryLen = 16#00000043
.....:idxGroup = 16#0000F031
.....:idxOffset = 16#00000005
.....:byteSize = 16#00000001
.....:adsDataType = ADST_BIT
.....:symDataType = 'BOOL'
.....:symComment = 'STARTS FB EXECUTION'
    
```

Die erhaltenen Daten haben folgende Bedeutung:

symEntryLen = 16#43: Der Eintrag in der Symboltabelle hat eine tatsächliche Größe von 67 Byte;

idxGroup = 16#F031: Es handelt es sich um eine Variable aus dem SPS-Prozessabbild der physikalischen Ausgänge;

idxOffset = 16#5: Die Variable liegt auf dem Byteoffset Null und dem Bitoffset 5;

byteSize = 16#1: Der Variablenwert belegt einen Byte im Speicher;

adsDataType = ADST_BIT: Die Ads-Datentyp-Id;

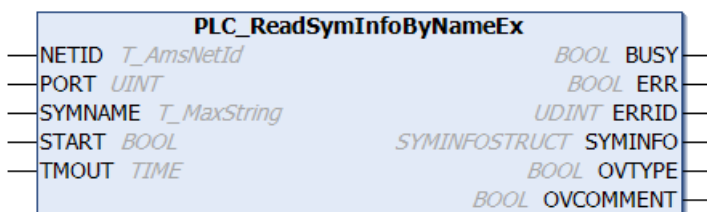
symDataType = BOOL: Die Datentypbezeichnung in der SPS;

symComment = 'STARTS FB EXECUTION': Kommentar, den der Benutzer in der Zeile der Variablendefinition hinzugefügt hat.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.72 PLC_ReadSymInfoByNameEx



Der Funktionsbaustein "PLC_ReadSymInfoByNameEx" hat eine ähnliche Funktionalität wie der Funktionsbaustein [PLC_ReadSymInfoByName](#) [131]. Beide Funktionsbausteine können Symbolinformationen anhand des Symbolnamens auslesen. Der Unterschied bei diesen zwei Bausteinen besteht darin, dass der hier beschriebene Baustein bei einer Überschreitung der verfügbaren Puffergröße keinen Fehler meldet und die nicht ganz kompletten Informationen am Ausgang trotzdem ausgibt. In diesem

Fall kann der Kommentar und/oder die Datentypbezeichnung abgeschnitten worden sein. Zwei zusätzliche Ausgangsvariablen zeigen dies an: *OVTYPE* und *OVCOMMENT*, so dass die Applikation darauf reagieren kann.

VAR_INPUT

```
VAR_INPUT
  NETID      : T_AmsNetId;
  PORT       : T_AmsPort;
  SYMNAME    : T_MaxString;
  START      : BOOL;
  TMOUT      : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT Rechners angegeben werden, auf dem Funktion ausgeführt werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

PORT: Die Portnummer des SPS-Laufzeitsystems zu dem die Symbolvariable gehört (Typ: T_AmsPort).

SYMNAME: Der Symbolname der SPS-Variablen deren Informationen gelesen werden soll (Typ: T_MaxString, max. 255 Zeichen, inklusive des gesamten Pfades z.B. 'MAIN.INIT_TASK.VARINT').

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY       : BOOL;
  ERR        : BOOL;
  ERRID      : UDINT;
  SymInfo    : SYMINFOSTRUCT;
  OVTYPE     : BOOL;
  OVCOMMENT  : BOOL;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [[▶ 360](#)].

SymInfo: Struktur mit zusätzlichen Informationen zu der Symbolvariablen (Typ: SYMINFOSTRUCT [[▶ 322](#)]).

OVTYPE: Zeigt an ob der String mit der Datentypbezeichnung einen Überlauf verursacht hat (TRUE). Der String mit der Datentypbezeichnung wurde möglicherweise abgeschnitten.

OVCOMMENT: Zeigt an ob der String mit dem Symbolkommentar einen Überlauf verursacht hat(TRUE). Der String mit dem Kommentar wurde möglicherweise abgeschnitten.

Beispiel:

Siehe in der Dokumentation des Funktionsbausteins: PLC_ReadSymInfoByName [[▶ 131](#)]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.73 PLC_Reset



Mit dem Funktionsbaustein "PLC_Reset" kann ein Reset eines SPS-Laufzeitsystems durchgeführt werden. Bei einem Reset der SPS werden die SPS-Variablen mit den Initialwerten belegt und die Ausführung des SPS-Programms gestoppt.

VAR_INPUT

```
VAR_INPUT
    NETID : T_AmsNetId;
    PORT : T_AmsPort;
    RESET : BOOL;
    TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem ein Reset eines SPS-Laufzeitsystems durchgeführt werden soll (Typ: T_AmsNetId). Für einen SPS-Reset auf dem lokalen Rechner kann auch ein Leerstring angegeben werden.

PORT: Beinhaltet die ADS-Portnummer des SPS-Laufzeitsystems dessen Reset durchgeführt werden soll (Typ: T_AmsPort).

RESET: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY : BOOL;
    ERR : BOOL;
    ERRID : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

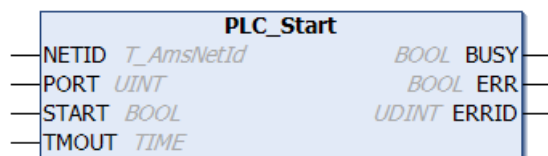
ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [[▶ 360](#)].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.74 PLC_Start



Mit dem Funktionsbaustein "PLC_Start" kann ein SPS-Laufzeitsystem eines TwinCAT-Rechners gestartet werden. Der Funktionsbaustein kann z.B. dazu benutzt werden, um die SPS auf einem Remote-PC zu starten.

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  PORT    : T_AmsPort;
  START   : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem die SPS gestartet werden soll (Typ: T_AmsNetID). Für einen SPS-Start auf dem lokalen Rechner kann auch ein Leerstring angegeben werden.

PORT: Beinhaltet die ADS-Portnummer des SPS-Laufzeitsystems das gestartet werden soll (Typ: T_AmsPort).

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY    : BOOL;
  ERR     : BOOL;
  ERRID   : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

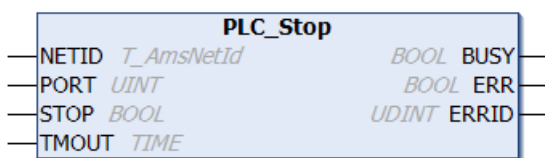
ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [► 360].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.75 PLC_Stop



Mit dem Funktionsbaustein "PLC_Stop" kann ein SPS-Laufzeitsystem eines TwinCAT-Rechners gestoppt werden. Der Funktionsbaustein kann z.B. dazu benutzt werden, um die SPS auf einem Remote/Lokalen-PC zu stoppen.

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  PORT    : T_AmsPort;
  STOP    : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem die SPS gestoppt werden soll (Typ: T_AmsNetID). Für einen SPS-Stop auf dem lokalen Rechner, kann auch ein Leerstring angegeben werden.

PORT: Beinhaltet die ADS-Portnummer des SPS-Laufzeitsystems das gestoppt werden soll (Typ: T_AmsPort).

STOP: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOU: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY   : BOOL;
    ERR    : BOOL;
    ERRID  : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [▶ 360].

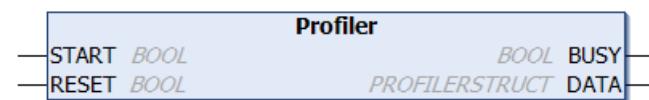
Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.76 Profiler



Diese Funktionalität ist in der SPS unter Windows CE nicht verfügbar!



Mit dem Funktionsbaustein "Profiler" können Messungen der Ausführungszeit vom SPS-Code durchgeführt werden. Intern wird eine Instanz des GETCPUACCOUNT-Funktionsbausteines aufgerufen. Über eine steigende Flanke an dem START-Eingang wird der Messvorgang gestartet und bei einer fallenden Flanke gestoppt. Die Messungen werden intern ausgewertet und stehen am DATA-Ausgang in einer Struktur vom Typ PROFILERSTRUCT [▶ 310] für weitere Verarbeitung zur Verfügung. Neben der aktuellen, minimalen und maximalen Ausführungszeit wird von dem Funktionsbaustein die mittlere Ausführungszeit der letzten 10 Messungen berechnet. Die Anzahl der gemittelten Messwerte kann über die globale Variable MAX_AVERAGE_MEASURES [▶ 330] auf 2 bis 100 Messwerte konfiguriert werden. Die gemessenen Zeiten werden in Mikrosekunden ausgegeben. Die Ausgangsvariable DATA.MeasureCycle [▶ 310] gibt Auskunft über die gesamte Anzahl der bereits durchgeführten Messungen. Um die Ausführungszeit von einem bestimmten Programmabschnitt der SPS zu messen, muss der Messvorgang über eine steigende START-Flanke am Anfang des zu messenden Programmabschnitts gestartet und am Ende über eine fallende START-Flanke gestoppt werden. Alle Werte am DATA-Ausgang können zurückgesetzt werden wenn gleichzeitig mit der steigenden Flanke am START auch eine steigende Flanke am RESET-Eingang erzeugt wird. Die alten Messwerte werden so zu Beginn einer neuen Messung zurückgesetzt und aus den folgenden Aufrufen des Funktionsbausteins neu berechnet.

Bemerkung:

Die ermittelten Zeiten können von den tatsächlichen Werten abweichen, da bereits für die Aufrufe des GETCPUACCOUNT-Funktionsbausteins einige Zeit benötigt wird. Diese Zeit ist rechnerabhängig und ist in den ermittelten Zeiten enthalten.

VAR_INPUT

```
VAR_INPUT
  START : BOOL;
  RESET : BOOL;
END_VAR
```

START: Über eine positive Flanke an diesem Eingang wird die Messung der Ausführungszeit gestartet. Über eine negative Flanke an diesem Eingang wird die Messung gestoppt und die aktuelle, minimale, maximale und mittlere Ausführungszeit neu berechnet. Die Variable DATA.MeasureCycle [► 310] wird dabei inkrementiert.

RESET: Wenn gleichzeitig mit einer steigenden Flanke am START-Eingang auch eine steigende Flanke an diesem Eingang erzeugt wurde dann werden alle Variablen am DATA-Ausgang zurückgesetzt. Die alten Werte für die aktuelle, minimale, maximale und mittlere Ausführungszeit werden dabei zurückgesetzt und für die nächsten Messungen neu berechnet.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY : BOOL;
  DATA : PROFILERSTRUCT;
END_VAR
```

BUSY: Beim Starten des Messvorgangs wird dieser Ausgang gesetzt und bleibt gesetzt, bis die Zeitmessung abgeschlossen wurde. Nachdem der BUSY-Ausgang zurückgesetzt wurde, stehen die aktuellen Zeiten am DATA-Ausgang zur Verfügung.

DATA: Struktur vom Typ PROFILERSTRUCT [► 310] mit den gemessenen Zeiten in [µs].

Beispiel 1:

```
PROGRAM ProfilerTest_ST
VAR
  Profiler1 : PROFILER;
  ProfilerData : PROFILERSTRUCT;
  a : LREAL;
END_VAR
```

Online Ansicht der gemessenen Zeiten:

▣ Profiler1 ▣ ProfilerData ----- .LastExecTime = 16#00000002 .MinExecTime = 16#00000002 .MaxExecTime = 16#00000004 .AverageExecTime = 16#00000002 .MeasureCycle = 16#00000E2B a = 0.370490944866529	
Profiler1 (Start= TRUE , Reset= TRUE): a := SIN(COS(TAN(12*0.4))); Profiler1 (Start= FALSE): ProfilerData:=Profiler1.Data;	a = 0.370490944866529

Beispiel 2:

```
PROGRAM ProfilerTest_FUP
VAR
  Profiler2 : PROFILER;
  Profiler2_Busy : BOOL;
  Profiler2_Data : PROFILERSTRUCT;
END_VAR
```

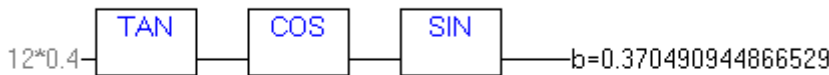


```
b :LREAL;
END_VAR
```

Online Ansicht der gemessenen Zeiten:

```

+ Profiler2
  Profiler2_Busy = FALSE
- Profiler2_Data
  .LastExecTime = 16#00000002
  .MinExecTime = 16#00000002
  .MaxExecTime = 16#00000002
  .AverageExecTime = 16#00000002
  .MeasureCycle = 16#00001FBB
  b = 0.370490944866529
    
```



Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64)	Tc2_Utilities (System)

3.77 RTC



Mit dem Funktionsbaustein "RTC" (Real Time Clock) kann eine interne Software-Uhr in der TwinCAT SPS realisiert werden. Die Uhr muss mit einem Anfangsdatum und einer Uhrzeit initialisiert werden. Nach der Initialisierung wird die Uhrzeit und das Datum mit jedem Aufruf des Funktionsbausteins aktualisiert. Um die aktuelle Uhrzeit und das Datum zu berechnen, wird ein Systemtakt der CPU benutzt. Damit die aktuelle Zeit berechnet werden kann, sollte der Funktionsbaustein in jedem Zyklus der SPS einmal aufgerufen werden. Am Ausgang des Funktionsbausteines steht das aktuelle Datum und Uhrzeit in dem gängigen DATE_AND_TIME (DT) Format zur Verfügung. In einem SPS-Programm können mehrere Instanzen von dem RTC-Funktionsbaustein erzeugt werden.

i Abweichung der RTC-Zeit von einer Referenzzeit

Bedingt durch die Systemeigenschaften weicht die RTC-Zeit von einer Referenzzeit ab. Die Abweichung hängt von der Zykluszeit der SPS, dem Wert des System-Basis-Ticks und der verwendeten Hardware.

Um größere Abweichungen zu vermeiden, muss die RTC-Instanz zyklisch (z. B. mit einer Funkuhr oder mit der lokalen Windows-Systemzeit) synchronisiert werden. Die lokale Windows-Systemzeit können Sie wiederum mit der Hilfe des SNTP-Protokolls mit einer Referenzzeit synchronisieren.

VAR_INPUT

```
VAR_INPUT
  EN : BOOL;
  PDT : DATE_AND_TIME;
END_VAR
```

EN: Bei einer steigenden Flanke an diesem Eingang wird der Funktionsbaustein mit einer vorgegebenen Uhrzeit und Datum neu initialisiert.

PDT: (Preset Date and Time) Die Initialisierungswerte für das Datum und Uhrzeit des Funktionsbausteins. Bei einer steigenden Flanke an dem EN-Eingang wird dieser Wert von dem Funktionsbaustein übernommen.

VAR_OUTPUT

```
VAR_OUTPUT
  Q : BOOL;
  CDT : DATE_AND_TIME;
END_VAR
```

Q: Wurde der Funktionsbaustein mindestens ein Mal initialisiert, wird dieser Ausgang gesetzt. Ist dieser Ausgang gesetzt, dann sind die Werte für das Datum und Uhrzeit am PDT-Ausgang gültig.

CDT: (Current Date and Time) Aktuelles Datum und Uhrzeit von der RTC-Instanz. Der CDT-Ausgang wird nur dann aktualisiert, wenn der Funktionsbaustein aufgerufen wurde. Daher sollten die Instanzen des Funktionsbausteines ein Mal in jedem Zyklus der SPS aufgerufen werden.

Beispiel:

Siehe: [Beispiel: Software-Uhren \(RTC, RTC_EX, RTC_EX2\)](#). [► 354]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.78 RTC_EX



Mit dem Funktionsbaustein "RTC_EX" (Extended Real Time Clock) kann eine interne Software-Uhr in der TwinCAT SPS realisiert werden. Die Uhr muss mit einem Anfangsdatum und einer Uhrzeit initialisiert werden. Nach der Initialisierung wird die Uhrzeit und das Datum mit jedem Aufruf des Funktionsbausteins aktualisiert. Um die aktuelle Uhrzeit und das Datum zu berechnen, wird ein Systemtakt der CPU benutzt. Damit die aktuelle Zeit berechnet werden kann, sollte der Funktionsbaustein in jedem Zyklus der SPS ein Mal aufgerufen werden. Am Ausgang des Funktionsbausteines steht das aktuelle Datum und die Uhrzeit in dem gängigen DATE_AND_TIME (DT) Format zur Verfügung. Im Gegensatz zu dem RTC [► 138]-Funktionsbaustein hat RTC_EX eine Millisekunden-Genauigkeit. In einem SPS-Programm können mehrere Instanzen von dem RTC_EX-Funktionsbaustein erzeugt werden.

i Abweichung der RTC_EX-Zeit von einer Referenzzeit

Bedingt durch die Systemeigenschaften weicht die RTC_EX-Zeit von einer Referenzzeit ab. Die Abweichung hängt ab von der Zykluszeit der SPS, dem Wert des System-Basis-Ticks und der verwendeten Hardware.

Um größere Abweichungen zu vermeiden, muss die RTC_EX-Instanz zyklisch (z. B. mit einer Funkuhr oder mit der lokalen Windows-Systemzeit) synchronisiert werden. Die lokale Windows-Systemzeit können Sie wiederum mit der Hilfe des SNTP-Protokolls mit einer Referenzzeit synchronisieren.

VAR_INPUT

```
VAR_INPUT
  EN      : BOOL;
  PDT     : DATE_AND_TIME;
  PMSEK   : DWORD;
END_VAR
```

EN: Bei einer steigenden Flanke an diesem Eingang wird der RTC_EX-Funktionsbaustein mit vorgegebener Uhrzeit, Datum und Millisekunden neu initialisiert.

PDT: (Preset Date and Time) Die Initialisierungswerte für Datum und Uhrzeit des Funktionsbausteins. Bei einer steigenden Flanke an dem EN-Eingang wird dieser Wert von dem Funktionsbaustein übernommen.

PMSEK: (Preset Milliseconds) Der Initialisierungswert für die Millisekunden. Bei einer steigenden Flanke an dem EN-Eingang wird dieser Wert von dem Funktionsbaustein übernommen.

VAR_OUTPUT

```
VAR_OUTPUT
  Q       : BOOL;
  CDT     : DATE_AND_TIME;
  CMSEK   : DWORD;
END_VAR
```

Q: Wurde der Funktionsbaustein mindestens ein Mal initialisiert, wird dieser Ausgang gesetzt. Ist dieser Ausgang gesetzt, dann sind die Werte für Datum, Uhrzeit und Millisekunden am PDT-Ausgang und CMSEK-Ausgang gültig.

CDT: (Current Date and Time) Aktuelles Datum und Uhrzeit von der RTC_EX-Instanz. Der CDT-Ausgang wird nur dann aktualisiert, wenn der Funktionsbaustein aufgerufen wurde. Daher sollten die Instanzen des Funktionsbausteines ein Mal in jedem Zyklus der SPS aufgerufen werden.

CMSEK: (Current Milliseconds) Der Millisekunden-Ausgang.

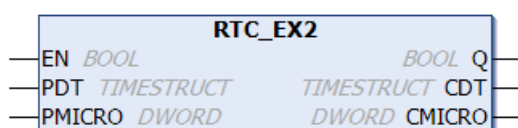
Beispiel:

Siehe: [Beispiel: Software-Uhren \(RTC, RTC_EX, RTC_EX2\)](#). [▶ 354]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.79 RTC_EX2



Mit dem Funktionsbaustein "RTC_EX2" (Extended Real Time Clock) kann eine interne Software-Uhr in der TwinCAT SPS realisiert werden. Die Uhr muss mit einem Anfangsdatum und einer Uhrzeit initialisiert werden. Nach der Initialisierung wird die Uhrzeit und das Datum mit jedem Aufruf des Funktionsbausteins

aktualisiert. Um die aktuelle Uhrzeit und das Datum zu berechnen, wird ein Systemtakt der CPU benutzt. Damit die aktuelle Zeit berechnet werden kann, sollte der Funktionsbaustein in jedem Zyklus der SPS einmal aufgerufen werden. Am Ausgang des Funktionsbausteines steht das aktuelle Datum und Uhrzeit in dem Windows-Systemzeitformat zur Verfügung. Im Gegensatz zu dem [RTC \[► 138\]](#)-Funktionsbaustein hat RTC_EX2 eine Mikrosekunden-Genauigkeit. In einem SPS-Programm können mehrere Instanzen von dem RTC_EX2-Funktionsbaustein erzeugt werden.

● **Abweichung der RTC_EX2-Zeit von einer Referenzzeit**

I Bedingt durch die Systemeigenschaften weicht die RTC_EX2-Zeit von einer Referenzzeit ab. Die Abweichung ist abhängig von der Zykluszeit der SPS, dem Wert des System-Basis-Ticks und der verwendeten Hardware.

Um größere Abweichungen zu vermeiden, muss die RTC_EX2-Instanz zyklisch (z. B. mit einer Funkuhr oder mit der lokalen Windows-Systemzeit) synchronisiert werden. Die lokale Windows-Systemzeit können Sie wiederum mit der Hilfe des SNTP-Protokolls mit einer Referenzzeit synchronisieren.

VAR_INPUT

```
VAR_INPUT
    EN      : BOOL;
    PDT     : Timestruct;
    PMICRO  : DWORD;
END_VAR
```

EN: Bei einer steigenden Flanke an diesem Eingang wird der RTC_EX2-Funktionsbaustein mit vorgegebener Uhrzeit, Datum und Millisekunden neu initialisiert.

PDT: (Preset Date and Time) Die Initialisierungswerte für Datum und Uhrzeit des Funktionsbausteines (Typ: [Timestruct \[► 327\]](#)). Bei einer steigenden Flanke an dem EN-Eingang wird dieser Wert von dem Funktionsbaustein übernommen.

PMICRO: (Preset Microseconds) Der Initialisierungswert für die Mikrosekunden. Bei einer steigenden Flanke an dem EN-Eingang wird dieser Wert von dem Funktionsbaustein übernommen.

VAR_OUTPUT

```
VAR_OUTPUT
    Q      : BOOL;
    CDT    : Timestruct;
    CMICRO : DWORD;
END_VAR
```

Q: Wurde der Funktionsbaustein mindestens einmal initialisiert, wird dieser Ausgang gesetzt. Ist dieser Ausgang gesetzt, dann sind die Werte für Datum, Uhrzeit und Millisekunden am PDT-Ausgang und CMICRO-Ausgang gültig.

CDT: (Current Date and Time) Aktuelles Datum und Uhrzeit von der RTC_EX2-Instanz (Typ: [Timestruct \[► 327\]](#)). Der CDT-Ausgang wird nur dann aktualisiert, wenn der Funktionsbaustein aufgerufen wurde. Daher sollten die Instanzen des Funktionsbausteines einmal in jedem Zyklus der SPS aufgerufen werden.

CMICRO: (Current Microseconds) Der Mikrosekunden-Ausgang.

Beispiel:

Siehe: [Beispiel: Software-Uhren \(RTC, RTC_EX, RTC_EX2\). \[► 354\]](#)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.80 TC_Config



Ein TwinCAT System, welches sich im RUN-Modus befindet (grünes TwinCAT System Icon) kann mit dem Funktionsbaustein "TC_Config" in den CONFIG-Modus (blaues TwinCAT System Icon) versetzt werden. Wenn sich das System bereits in dem CONFIG-Modus befindet, dann wird das System zuerst in den STOP-Modus (rotes TwinCAT System Icon) und dann in den CONFIG-Modus versetzt.

VAR_INPUT

```

VAR_INPUT
  NETID : T_AmsNetId;
  SET : BOOL;
  TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
  
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, welcher in den CONFIG-Modus versetzt werden soll (Typ: T_AmsNetID). Für den lokalen TwinCAT-Rechner kann auch ein Leerstring angegeben werden.

SET: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```

VAR_OUTPUT
  BUSY : BOOL;
  ERR : BOOL;
  ERRID : UDINT;
END_VAR
  
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

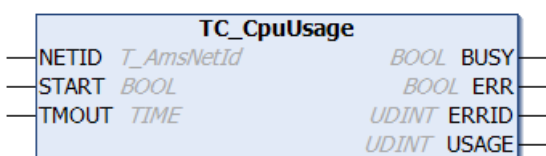
ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [► 360].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.81 TC_CpuUsage



Mit dem Funktionsbaustein "TC_CpuUsage" kann die aktuelle CPU-Auslastung eines TwinCAT-Systems ermittelt werden. Die Funktion entspricht der Anzeige der CPU-Auslastung im TwinCAT-System Menü unter den Echtzeiteinstellungen.

VAR_INPUT

```
VAR_INPUT
  NETID : T_AmsNetId;
  START : BOOL;
  TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen CPU-Auslastung ermittelt werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY : BOOL;
  ERR : BOOL;
  ERRID : UDINT;
  USAGE : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

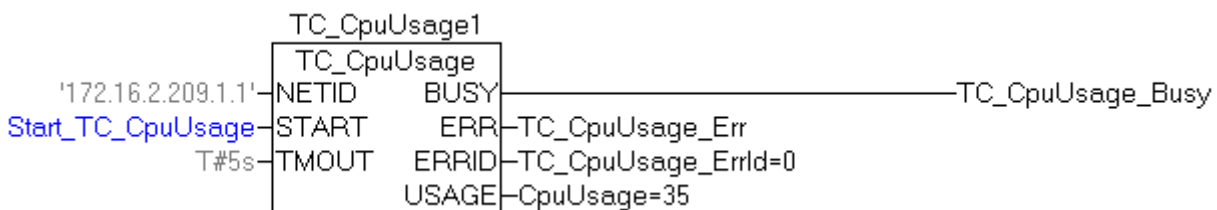
ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [[▶ 360](#)].

USAGE: Die aktuelle CPU-Auslastung eines TwinCAT-Systems in %.

Beispiel:

```
PROGRAM MAIN
VAR
  TC_CpuUsage1 : TC_CpuUsage;
  Start_TC_CpuUsage : BOOL;
  TC_CpuUsage_Busy : BOOL;
  TC_CpuUsage_Err : BOOL;
  TC_CpuUsage_ErrId : UDINT;
  CpuUsage : UDINT;
END_VAR
```

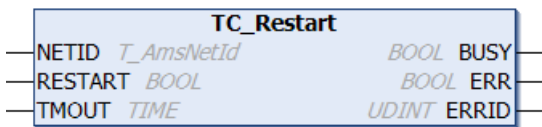


In dem Beispiel benutzt das TwinCAT-System 35% von der gesamten zur Verfügung stehenden CPU-Rechenzeit.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.82 TC_Restart



Mit dem Funktionsbaustein "TC_Restart" kann ein Restart des TwinCAT-Systems durchgeführt werden. Die Funktion entspricht dem Restart-Befehl aus dem TwinCAT-System Menü (rechts in der Windows-Startleiste). Bei einem Restart des TwinCAT-Systems wird zuerst das TwinCAT-System gestoppt und sofort erneut gestartet.

VAR_INPUT

```

VAR_INPUT
  NETID    : T_AmsNetId;
  RESTART  : BOOL;
  TMOUT    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
  
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem ein Restart des TwinCAT-Systems durchgeführt werden soll (Typ: T_AmsNetID). Für einen Restart auf dem lokalen Rechner, kann auch ein Leerstring angegeben werden.

RESTART: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```

VAR_OUTPUT
  BUSY    : BOOL;
  ERR     : BOOL;
  ERRID   : UDINT;
END_VAR
  
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [► 360].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

3.83 TC_Stop



Mit dem Funktionsbaustein "TC_Stop" kann ein Stopp des TwinCAT-Systems durchgeführt werden. Die Funktion entspricht dem Stopp-Befehl aus dem TwinCAT-System Menü (rechts in der Windows-Startleiste).

VAR_INPUT

```
VAR_INPUT
  NETID : T_AmsNetId;
  STOP  : BOOL;
  TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, auf dem ein TwinCAT-Systemstopp durchgeführt werden soll (Typ: T_AmsNetID). Für ein TwinCAT-Systemstopp auf dem lokalen Rechner, kann auch ein Leerstring angegeben werden.

STOP: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY : BOOL;
  ERR   : BOOL;
  ERRID : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [► 360].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.84 TC_SysLatency



Mit dem Funktionsbaustein "TC_SysLatency" kann die aktuelle und maximale Latenzzeit eines TwinCAT-Systems ermittelt werden. Die Funktion entspricht der Anzeige der TwinCAT-Latenzzeit im TwinCAT-System Menü unter den Echtzeiteinstellungen.

VAR_INPUT

```
VAR_INPUT
  NETID : T_AmsNetId;
  START  : BOOL;
  TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Hier kann ein String mit der Netzwerkadresse des TwinCAT-Rechners angegeben werden, dessen Latenzzeit ermittelt werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann auch ein Leerstring angegeben werden.

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Gibt die Timeout-Zeit an, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY      : BOOL;
  ERR       : BOOL;
  ERRID     : UDINT;
  ACTUAL    : UDINT;
  MAXIMUM   : UDINT;
END_VAR
```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die ADS-Fehlernummer [► 360].

ACTUAL: Die aktuelle Latenzzeit eines TwinCAT-Systems in µs.

MAXIMUM: Die maximale Latenzzeit eines TwinCAT-Systems in µs (Maximale Latenzzeit, die seit dem letzten TwinCAT System-Start aufgetreten ist).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

3.85 WritePersistentData



Wenn in einem SPS-Laufzeitsystem persistente Variablen definiert sind, werden deren aktuelle Werte beim TwinCAT-System-Stopp/Shutdown (nach dem letztem Zyklus der SPS) regulär in einer .bootdata-Datei im TwinCAT\Boot-Ordner gespeichert. Vor dem Schreiben der Datei mit den aktuellen persistenten Daten wird ein Backup der alten persistenten Daten erstellt, indem die alte .bootdata-Datei vom System zu .bootdata-old umbenannt wird.

Für jedes konfigurierte Laufzeitsystem wird eine Datei angelegt.

Beim nächsten Systemstart wird die .bootdata-Datei eingelesen und die persistenten Variablen werden im Laufzeitsystem mit den Werten aus der Datei initialisiert.

Die Backup-Datei (.bootdata-old) der persistenten Daten wird beim Systemstart eingelesen, wenn die Datei (.bootdata) der persistenten Daten nicht vorhanden ist. Dieser Fall ist eine Ausnahme, kann aber z. B. dann eintreten, wenn ein IPC ohne USV einen Stromausfall erfährt und TwinCAT somit nicht ordnungsgemäß herunterfahren konnte.

Mit dem Funktionsbaustein WritePersistentData können Sie das Speichern der persistenten Daten aus dem SPS-Programm heraus auslösen und sicherstellen, dass eine aktuelle .bootdata-Datei der persistenten Daten verfügbar ist. Der Eingangsparameter PORT bestimmt das Laufzeitsystem, dessen persistente Daten gespeichert werden sollen.

VAR_INPUT

```
VAR_INPUT
  NETID : T_AmsNetId;
  PORT  : T_AmsPort;
```

```

START   : BOOL;
TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

```

NETID: Netzwerkadresse des TwinCAT-Rechners, auf dem das ADS-Kommando ausgeführt werden soll (Typ: T_AmsNetID). Für den lokalen Rechner kann ein Leerstring angegeben werden.

PORT: ADS-Portnummer des SPS-Laufzeitsystems, dessen persistente Daten gespeichert werden sollen (Typ: T_AmsPort).

START: Über eine positive Flanke an diesem Eingang wird der Baustein aktiviert.

TMOUT: Timeout-Zeit, die bei der Ausführung des ADS-Kommandos nicht überschritten werden darf.

VAR_OUTPUT

```

VAR_OUTPUT
  BUSY   : BOOL;
  ERR    : BOOL;
  ERRID  : UDINT;
END_VAR

```

BUSY: Bei der Aktivierung des Funktionsbausteins wird dieser Ausgang gesetzt und bleibt gesetzt, bis eine Rückmeldung erfolgt.

ERR: Sollte ein ADS-Fehler bei der Übertragung des Kommandos erfolgen, dann wird dieser Ausgang gesetzt, nachdem der BUSY-Ausgang zurückgesetzt wurde.

ERRID: Liefert bei einem gesetzten ERR-Ausgang die [ADS-Fehlernummer](#) [[▶ 360](#)].

Beispiel:

```

PROGRAM MAIN
VAR
  bStart           : BOOL;
  bError           : BOOL;
  bBusy            : BOOL;
  nErrorId        : UDINT;
  fbWritePersistentData : WritePersistentData;
  fbR_Trig        : R_TRIG;
END_VAR

VAR PERSISTENT
  perA : INT;
  perB : BOOL;
  perC : BYTE;
  perD : STRING;
  perE : ARRAY[0..10] OF INT;
  perF : ARRAY[0..10] OF UDINT;
END_VAR

fbR_Trig( CLK:=bStart );
IF fbR_Trig.Q THEN
  perA := 24443;
  perB := TRUE;
  perC := 7;
  perD := 'Switch ON/OFF';
  perE[ 0 ] := 1;
  perE[ 10 ] := 11;
  perF[ 0 ] := 263;
  perF[ 10 ] := 23323;
  fbWritePersistentData( NETID='', PORT:=851, START:=bStart, TMOUT:=T#1s );
ELSE
  fbWritePersistentData( START:=FALSE);
END_IF;

bBusy := fbWritePersistentData.BUSY;
bError := fbWritePersistentData.ERR;
nErrorId := fbWritePersistentData.ERRID;

```

Siehe auch: [Anhang > Systemverhalten beim Schreiben von persistenten Daten](#) [[▶ 356](#)]

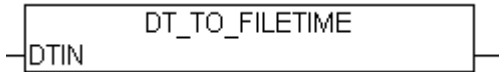
Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4 Funktionen

4.1 Zeitfunktionen

4.1.1 DT_TO_FILETIME64



Mit der Funktion "DT_TO_FILETIME64" kann eine im DATE_AND_TIME-Format (DT) SPS-Variable in das FILETIME-Format (64Bit) konvertiert werden.

FUNCTION DT_TO_FILETIME64 : T FILETIME64 [► 323]

VAR_INPUT

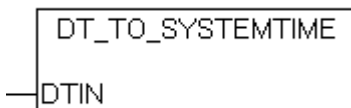
```
VAR_INPUT
    DTIN : DT;
END_VAR
```

DTIN: Das zu konvertierende Datum und Uhrzeit im DATE_AND_TIME-Format.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.44.0

4.1.2 DT_TO_SYSTEMTIME



Mit der Funktion "DT_TO_SYSTEMTIME" kann eine im DATE_AND_TIME-Format (DT) SPS-Variable in eine Windows Systemzeit-Struktur konvertiert werden. Die Systemzeit hat eine Auflösung von 1ms, und das DATE_AND_TIME eine Auflösung von 1s. Die Variable "wMilliseconds" in der Systemzeit-Struktur liefert daher immer den Wert Null zurück.

FUNCTION DT_TO_SYSTEMTIME : Timestruct [► 327]

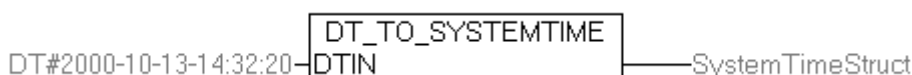
VAR_INPUT

```
VAR_INPUT
    DTIN : DT;
END_VAR
```

DTIN: Das zu konvertierende Datum und Uhrzeit in DATE_AND_TIME-Format.

Beispiel:

```
PROGRAM SystemTimeTest
VAR
    SystemTimeStruct : Timestruct;
END_VAR
```



Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.1.3 F_GetDayOfMonthEx



Die Funktion berechnet das Datum des ersten, zweiten usw. Wochentages in einem bestimmten Monat und Jahr (z.B. das Datum des zweiten Montags in Januar 2011).

FUNCTION F_GetDayOfMonthEx: WORD

VAR_INPUT

```

VAR_INPUT
  wYear  : WORD(1601..30827);
  wMonth : WORD(1..12);
  wWOM   : WORD(1..5);
  wDOW   : WORD(0..6);
END_VAR
    
```

wYear: Jahr (1601 bis 30827).

wMonth: Monat (1 bis 12).

wWOM: Woche im Monat (1 bis 5). Der Wert 1 entspricht der ersten Woche, 2 der zweiten Woche und 5 der letzten Woche (auch wenn der Monat keine 5 Wochen besitzt).

wDOW: Wochentag (0 bis 6). 0 = Sonntag, 1 = Montag... 6 = Samstag.

Rückgabeparameter	Beschreibung
0	Fehler, falscher oder unzulässiger Funktionsparameter
> 0	Kein Fehler. Monatstag

Beispiel:

Das Beispiel ermittelt das Datum des zweiten Montags im August 2011. Das Ergebnis ist: 8.

```

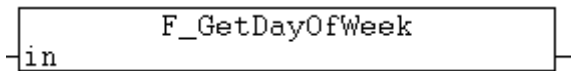
PROGRAM P_Dok_F_GetDayOfMonthEx
VAR
  wYear  : WORD := 2011;
  wMonth : WORD := 8;
  wWOM   : WORD(1..5) := 2; (* Week of month: 2 = Second week *)
  wDOW   : WORD(0..6) := 1; (* Day of week 1 = Monday *)
  wDay   : WORD; (* Day of month *)
END_VAR

wDay := F_GetDayOfMonthEx( wYear, wMonth, wWOM, wDOW );
    
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.1.4 F_GetDayOfWeek



Die Funktion liefert die Nummer des Wochentags nach DIN 1355 / ISO 8601. Nach dieser Norm sind die Wochentage wie folgt Nummeriert: Montag = 1, Dienstag = 2, ... Sonntag = 7.

FUNCTION F_GetDayOfWeek : WORD

VAR_INPUT

```
VAR_INPUT
    in : DT;
END_VAR
```

in: Das Datum, dessen Wochentagsnummer ermittelt werden soll.

Beispiel:

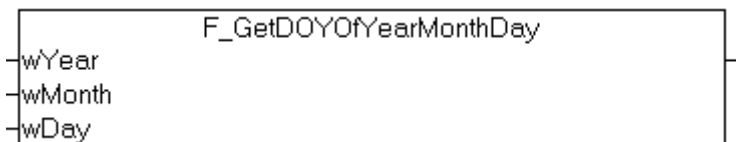
```
PROGRAM MAIN
VAR
    dtFirst : DT := DT#2008-01-01-00:00;
    dayOfWeek : WORD;
END_VAR
dayOfWeek := F_GetDayOfWeek(dtFirst);
```

Das Ergebnis ist 2 (Dienstag)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.5 F_GetDOYOfYearMonthDay



Die Funktion berechnet die Nummer des Tages im Jahr.

FUNCTION F_GetDOYOfYearMonthDay: WORD

VAR_INPUT

```
VAR_INPUT
    wYear : WORD;
    wMonth : WORD;
    wDay : WORD;
END_VAR
```

wYear: Jahr (0 ~ 2999).

wMonth: Monat (1 ~ 12).

wDay: Tag (1 ~ 31).

Rückgabeparameter	Beschreibung
0	Fehler, falscher wYear, wMonth oder wDay Parameterwert

Rückgabeparameter	Beschreibung
> 0	Kein Fehler. Nummer des Tages im Jahr (1 ~ 366)

Beispiel:

```
PROGRAM P_TEST_DOY
VAR
  wYear   : WORD;
  wDOY    : WORD;
  wMonth  : WORD;
  wDay    : WORD;
END_VAR

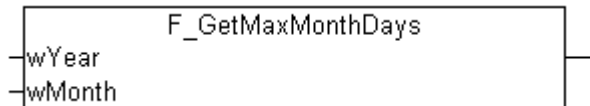
wYear := 2009;
wMonth := 1;
wDay := 31;
wDOY := F_GetDOYOfYearMonthDay( wYear, wMonth, wDay ); (* wDOY = 31 *)

wYear := 2009;
wMonth := 2;
wDay := 1;
wDOY := F_GetDOYOfYearMonthDay( wYear, wMonth, wDay ); (* wDOY = 32 *)

wYear := 2009;
wMonth := 3;
wDay := 1;
wDOY := F_GetDOYOfYearMonthDay( wYear, wMonth, wDay ); (* wDOY = 60 *)
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.1.6 F_GetMaxMonthDays

Die Funktion liefert die maximale Anzahl der Monatstage in einem bestimmten Monat und Jahr.

FUNCTION F_GetMaxMonthDays : WORD**VAR_INPUT**

```
VAR_INPUT
  wYear : WORD;
  wMonth : WORD;
END_VAR
```

wYear: Jahr.

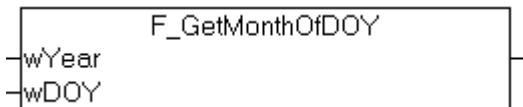
wMonth: Monat (1 bis 12).

Rückgabeparameter	Beschreibung
0	Fehler, falscher wMonth Parameterwert
> 0	Kein Fehler. Maximale Anzahl der Monatstage.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.1.7 F_GetMonthOfDOY



Die Funktion berechnet den Monat anhand der Nummer des Tages im Jahr.

FUNCTION F_GetMonthOfDOY: WORD

VAR_INPUT

```
VAR_INPUT
    wYear   : WORD;
    wDOY    : WORD;
END_VAR
```

wYear: Jahr (0 ~ 2999).

wDOY: Nummer des Tages im vorgegebenen Jahr dessen Monat ermittelt werden soll (1 ~ 366).

Rückgabeparameter	Beschreibung
0	Fehler, falscher wYear oder wDOY Parameterwert.
> 0	Kein Fehler. Monat (1 ~ 12).

Beispiel:

```
PROGRAM P_TEST_DOY
VAR
    wYear   : WORD;
    wDOY    : WORD;
    wMonth  : WORD;
END_VAR

wYear := 2009;
wDOY  := 31;
wMonth := F_GetMonthOfDOY( wYear, wDOY ); (* wMonth = 1 *)

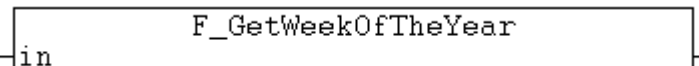
wYear := 2009;
wDOY  := 32;
wMonth := F_GetMonthOfDOY( wYear, wDOY ); (* wMonth = 2 *)

wYear := 2009;
wDOY  := 60;
wMonth := F_GetMonthOfDOY( wYear, wDOY ); (* wMonth = 3 *)
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.1.8 F_GetWeekOfTheYear



Die Funktion liefert die Nummer der Kalenderwoche zu einem vorgegebenen Datum nach der DIN 1355 / ISO 8601 Norm.

- Die erste Kalenderwoche ist definiert als die erste Woche, in die **mindestens vier Tage des neuen Jahres fallen (DIN 1355 / ISO 8601)**;
- Die Kalenderwochen beginnen Montags und jede Kalenderwoche enthält 7 Tage;
- Der zurückgelieferte Wert in der ersten Kalenderwoche hat die Nummer 1;
- Der 29, 30 und 31 Dezember kann auch schon zur ersten Kalenderwoche des Folgejahres gehören;

- Der 1, 2 und 3 Januar kann auch noch zur letzten Kalenderwoche des Vorjahres gehören;

FUNCTION F_GetWeekOfTheYear : WORD

VAR_INPUT

```
VAR_INPUT
  in : DT;
END_VAR
```

in: Das Datum, dessen Kalenderwoche ermittelt werden soll.

Beispiel:

```
PROGRAM MAIN
VAR
  dtNow      : DT := DT#2008-03-17-12:00;
  weekOfYear : WORD;
END_VAR
```

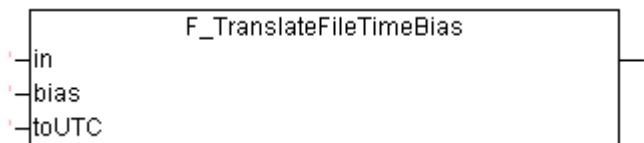
```
weekOfYear := F_GetWeekOfTheYear(dtNow);
```

Das Ergebnis ist 12.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.9 F_TranslateFileTime64Bias



Diese Funktion rechnet die Eingangszeit mit Hilfe der angegebenen Bias-Zeitverschiebung in die Zeit einer anderen Zeitzone um. Mit dieser Funktion kann z.B. die Lokalzeit in UTC-Zeit (Universal Time Coordinates) und umgekehrt umgerechnet werden.

FUNCTION F_TranslateFileTime64Bias: T_FILETIME64 |▶ 323|

VAR_INPUT

```
VAR_INPUT
  in      : T_FILETIME64;
  bias   : DINT;
  toUTC  : BOOL;
END_VAR
```

in: Eingangszeit die umgerechnet werden soll (Typ: [T_FILETIME64 |▶ 323|](#)).

bias: Differenz zwischen der UTC-Zeit und der Lokalzeit in Minuten (positive oder negative Werte sind zulässig).

toUTC: Über diesen Parameter kann die Richtung angegeben werden, in die die Eingangszeit konvertiert werden soll.

toUTC	Direction	Internal formula
FALSE	UTC -> Lokalzeit	Lokalzeit := UTC - Bias
TRUE	Lokalzeit -> UTC	UTC := Lokalzeit + Bias

Beispiel:

Die *in*-Variable enthält die zu konvertierende Zeit. Die *bToUTC*-Variable bestimmt die Konvertierungsrichtung. Bei *bToUTC* = TRUE wird die Lokalzeit in UTC-Zeit konvertiert und bei *bToUTC* = FALSE die UTC-Zeit in Lokalzeit. Die *WEST_EUROPE_TZI*-Konstante enthält die Zeitzoneninformationen für Westeuropa. Der benötigte Bias-Wert wird aus der Zeitzoneninformation in der Konstanten und der aktuellen bDST-Einstellung (Daylight Saving Time) errechnet. Die aktuelle Zeitzoneninformation eines TwinCAT-Systems kann alternativ mit dem Funktionsbaustein: [FB_GetTimeZoneInformation \[▶ 78\]](#) ermittelt werden.

Wichtiger Hinweis: Der Datentyp DT als Eingangszeit wurde nur wegen der Möglichkeit der optischen Kontrolle im Online-Mode gewählt. Konvertierungen in einem anderen Zeitformat sind nur bedingt zu empfehlen da die Konvertierungsfunktionen sehr rechenintensiv sein können.

```
PROGRAM MAIN
VAR
    bDST      : BOOL := TRUE; (* TRUE => Daylight saving time, FALSE => Standard time *)
    bToUTC    : BOOL := FALSE;
    (* TRUE => Convert local time to UTC time, FALSE => Convert UTC time to local time *)
    in        : DT := DT#2011-08-29-15:15:31;
    out       : DT;
    bias      : DINT;
END_VAR

IF bDST THEN
    bias := WEST_EUROPE_TZI.bias + WEST_EUROPE_TZI.daylightBias;
ELSE
    bias := WEST_EUROPE_TZI.bias + WEST_EUROPE_TZI.standardBias;
END_IF

out := FILETIME64_TO_DT( F_TranslateFileTime64Bias( DT_TO_FILETIME64( in ), bias, bToUTC ) );
```

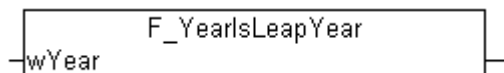
Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

- [FB_TzSpecificLocalTimeToSystemTime \[▶ 117\]](#)
- [FB_TzSpecificLocalTimeToFileTime64 \[▶ 115\]](#)
- [FB_SystemTimeToTzSpecificLocalTime \[▶ 113\]](#)
- [FB_FileTime46ToTzSpecificLocalTime \[▶ 62\]](#)
- [FB_GetTimeZoneInformation \[▶ 78\]](#)
- [FB_SetTimeZoneInformation \[▶ 111\]](#)
- [NT_SetLocalTime \[▶ 124\]](#)
- [NT_GetTime \[▶ 122\]](#)
- [NT_SetTimeToRTCTime \[▶ 125\]](#)
- [FB_LocalSystemTime \[▶ 90\]](#)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.44.0

4.1.10 F_YearIsLeapYear



Die Funktion ermittelt ob ein Jahr ein Schaltjahr ist.

FUNCTION F_YearIsLeapYear: BOOL

VAR_INPUT

```
VAR_INPUT
    wYear : WORD;
END_VAR
```

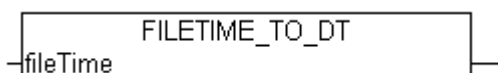
wYear: Jahr.

Rückgabeparameter	Beschreibung
TRUE	Schaltjahr
FALSE	Kein Schaltjahr

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.11 FILETIME64_TO_DT



Die Funktion "FILETIME64_TO_DT" konvertiert die Zeit im FILETIME-Format in das DATE_AND_TIME-Format (DT). Das DT-Format hat einen kleineren Wertebereich als das FILETIME-Format und nur eine Sekundengenauigkeit. Aus diesem Grund wird der zu konvertierende FILETIME-Wert limitiert. Das zulässige Minimum entspricht dem Wert *DT#1970-01-01-00:00:00* und das Maximum dem Wert *DT#2106-02-06-06:28:15*. Millisekunden werden bei der Konvertierung nicht berücksichtigt und auf den DATE_AND_TIME-Rückgabewert entsprechend abgerundet.

FUNCTION FILETIME64_TO_DT : DT

VAR_INPUT

```

VAR_INPUT
    fileTime : T_FILETIME64;
END_VAR
  
```

fileTime: Die zu konvertierende Zeit im FILETIME-Format (Typ: [T_FILETIME64](#) [► 323]).

Beispiel:

```

PROGRAM MAIN
VAR
    timeAsFileTime : T_FILETIME64;
    timeAsDT       : DT;
END_VAR

timeAsFileTime := F_GetSystemTime();
timeAsDT := FILETIME64_TO_DT( timeAsFileTime );
  
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.44.0

4.1.12 FILETIME64_TO_ISO8601

Die Funktion konvertiert die Windows-Systemzeit im T_FILETIME64-Format in einen String mit dem Format der Norm ISO 8601.

Das Resultat entspricht folgendem Schema: **YYYY-MM-DDThh:mm:ss.xxxTZD**

FUNCTION FILETIME64_TO_ISO8601 : STRING(39)

VAR_INPUT

```
VAR_INPUT
    fileTime : T_FILETIME64; (* Time to be converted (file time format), 64-bit value representing the number of 100-nanosecond intervals since January 1, 1601 *)
    nBias : INT; (* Specifies the current bias, in minutes, for local time translation on this computer.

    The bias is the difference, in minutes, between Coordinated Universal Time (UTC) and local time.

    UTC = local time + bias *)
    bUTC : BOOL; (* Specifies whether the fileTime is UTC or local time. *)
    nPrecision : USINT(0..9); (* Precision. Number of decimal places of seconds. (0..9) *)
END_VAR
```

fileTime: Gibt die zu konvertierende Zeit an (Typ: [T_FILETIME64](#) [[▶ 323](#)]).

nBias: Gibt den aktuellen Zeitversatz in Minuten zwischen der Coordinated Universal Time (UTC) und der lokalen Zeit an. Es gilt: UTC = lokale Zeit + Zeitversatz

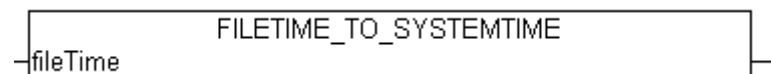
bUTC: Gibt an, ob die am Eingang angegebene Zeit der UTC oder der lokalen Zeit entspricht.

nPrecision: Gibt die Genauigkeit der Sekundendarstellung an als Anzahl der Nachkommastellen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.46.0

4.1.13 FILETIME64_TO_SYSTEMTIME



Die Funktion "FILETIME64_TO_SYSTEMTIME" konvertiert die Zeit im FILETIME-Format in das "lesbare" SYSTEMTIME-Format. Die Konvertierung schlägt fehl, wenn das höchstwertige Bit der 64 Bit fileTime-Variablen gesetzt ist. Die TIMESTRUCT-Membervariablen haben in diesem Fall den Wert Null.

FUNCTION FILETIME64_TO_SYSTEMTIME: TIMESTRUCT [[▶ 327](#)]

VAR_INPUT

```
VAR_INPUT
    fileTime : T_FILETIME64;
END_VAR
```

fileTime: Die zu konvertierende Zeit im FILETIME-Format (Typ: [T_FILETIME64](#) [[▶ 323](#)]).

Beispiel:

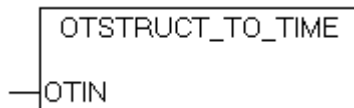
```
PROGRAM MAIN
VAR
    timeAsFileTime : T_FILETIME64;
    timeAsSystemTime : TIMESTRUCT;
END_VAR

timeAsFileTime := F_GetSystemTime();
timeAsSystemTime := FILETIME64_TO_SYSTEMTIME( timeAsFileTime );
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.44.0

4.1.14 OTSTRUCT_TO_TIME



Mit der Funktion "OTSTRUCT_TO_TIME" kann eine Struktur mit den aufgelösten Millisekunden, Sekunden, Minuten, Stunden, Tagen und Wochen in eine TIME-Variable konvertiert werden.

FUNCTION OTSTRUCT_TO_TIME : TIME

VAR_INPUT

```

VAR_INPUT
  OTIN : OTSTRUCT;
END_VAR

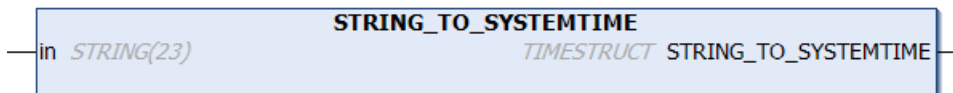
```

OTIN: Die zu konvertierende Struktur (Typ: [OTSTRUCT](#) [[▶ 310](#)]).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.15 STRING_TO_SYSTEMTIME



Die Funktion konvertiert einen String in den Windows SYSTEMTIME -Zeitformat.

FUNCTION STRING_TO_SYSTEMTIME: Timestruct [[▶ 327](#)]

VAR_INPUT

```

VAR_INPUT
  in : STRING(23);
END_VAR

```

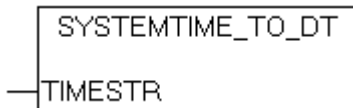
in: Der zu konvertierende String. Der String muss folgenden Format haben: 'YYYY-MM-DD-hh:mm:ss.xxx'

- YYYY: Jahr (1601..9999);
- MM: Monat (01..12);
- DD: Tag (01..31);
- hh: Stunde (00..23);
- mm: Minute (00..59);
- ss: Sekunde (00..59);
- xxx: Millisekunde (000..999);

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.16 SYSTEMTIME_TO_DT



Mit der Funktion "SYSTEMTIME_TO_DT" kann die Windows Systemzeit-Struktur in das in der SPS gängige DATE_AND_TIME-Format (DT) konvertiert werden. Die Systemzeit hat eine Auflösung von 1ms, und das DATE_AND_TIME eine Auflösung von 1s. Die Millisekunden aus der Systemzeit werden bei der Konvertierung berücksichtigt und auf den DATE_AND_TIME-Rückgabewert entsprechend aufgerundet. Setzen Sie das wMilliseconds-Element in der Windows Systemzeit-Struktur auf Null um das Aufrunden zu deaktivieren.

FUNCTION SYSTEMTIME_TO_DT : DT

VAR_INPUT

```
VAR_INPUT
    Timestr : TIMESTRUCT;
END_VAR
```

Timestr: Struktur mit der zu konvertierenden Windows Systemzeit (Typ: [TIMESTRUCT](#) [▶ 327]).

Beispiel:

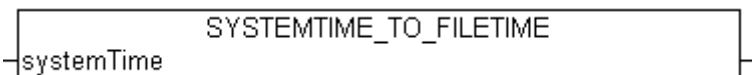
```
PROGRAM SystemTimeTest
VAR
    SystemTimeStruct : TIMESTRUCT;
    DTFromSystemTime : DT;
END_VAR
```



Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.17 SYSTEMTIME_TO_FILETIME64



Mit der Funktion kann die Windows Systemzeit-Struktur in das Filetime-Format konvertiert werden. Der Wochentag wDayOfWeek der SystemTime-Variablen wird ignoriert. Das Systemzeit-Jahr muss größer als 1601 und kleiner als 30827 sein.

FUNCTION SYSTEMTIME_TO_FILETIME64 : T_FILETIME64 [▶ 323]

VAR_INPUT

```
VAR_INPUT
    systemTime : TIMESTRUCT;
END_VAR
```

systemTime: Struktur mit der zu konvertierenden Windows-Systemzeit (Typ: [TIMESTRUCT](#) [▶ 327]).

Rückgabeparameter	Beschreibung
0	Fehler, falscher SystemTime Parameterwert.
> 0	Kein Fehler. File time.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.44.0

4.1.18 SYSTEMTIME_TO_ISO8601

Die Funktion konvertiert die Windows Systemzeit-Struktur in einen String mit dem Format der Norm ISO 8601.

Das Resultat entspricht folgendem Schema: **YYYY-MM-DDThh:mm:ss.xxxTZD**

FUNCTION SYSTEMTIME_TO_ISO8601 : STRING(39)

VAR_INPUT

```

VAR_INPUT
    systemTime : TIMESTRUCT; (* Input time in system time format (struct) *)
    nBias      : INT;        (* Specifies the current bias, in minutes, for local time translation
on this computer.
                                The bias is the difference between Coordinated Universal Time (UTC)
and local time.
                                UTC = local time + bias *)
    bUTC       : BOOL;      (* Specifies whether the systemTime is UTC or local time. *)
    nPrecision : USINT(0..9); (* Precision. Number of decimal places of seconds. (0..9) *)
END_VAR

```

systemTime: Struktur mit der zu konvertierenden Windows-Systemzeit (Typ: [TIMESTRUCT \[► 327\]](#)).

nBias: Gibt den aktuellen Zeitversatz in Minuten an, zwischen der Coordinated Universal Time (UTC) und der lokalen Zeit. Es gilt: UTC = lokale Zeit + Zeitversatz

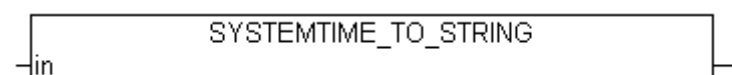
bUTC: Gibt an, ob die am Eingang angegebene Zeit der UTC oder der lokalen Zeit entspricht.

nPrecision: Gibt die Genauigkeit der Sekundendarstellung an als Anzahl der Nachkommastellen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.46.0

4.1.19 SYSTEMTIME_TO_STRING



Die Funktion konvertiert die Windows Systemzeit-Struktur in einen String mit folgendem Format: **YYYY-MM-DD-hh:mm:ss.xxx** :

- YYYY: Jahr (1601..9999)
- MM: Monat (01..12)
- DD: Tag (01..31)
- hh: Stunde (00..23)
- mm: Minuten (00..59)

- ss: Sekunden (00..59)
- xxx: Millisekunde (000..999)

FUNCTION SYSTEMTIME_TO_STRING : STRING(24)

VAR_INPUT

```
VAR_INPUT
  in : TIMESTRUCT;
END_VAR
```

in: Struktur mit der zu konvertierenden Windows-Systemzeit (Typ: [TIMESTRUCT](#) [[▶ 327](#)]).

Beispiel:

```
PROGRAM MAIN
VAR
  fbGetSystemTime : GETSYSTEMTIME;
  fileTime        : T_FILETIME;
  sTime           : STRING;
END_VAR
```

```
fbGetSystemTime(timeLoDW=>fileTime.dwLowDateTime, timeHiDW=>fileTime.dwHighDateTime );
sTime := SYSTEMTIME_TO_STRING( FILETIME_TO_SYSTEMTIME( fileTime ) );
```

Onlineansicht:

Expression	Type	Value	Prepared value	Address	Comment
fbGetSystemTime	GETSYSTEMTIME				
fileTime	T_FILETIME				
sTime	STRING	'2014-09-22-11:16:48.935'			

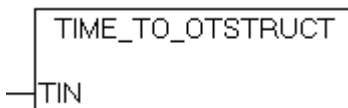
```

1 fbGetSystemTime(timeLoDW 16#B3D07B70 ->fileTime.dwLowDateTime 16#B3D07B70, timeHiDW 16#01CFD656 ->fileTime.dwHighDateTime 16#01CFD656 );
2 sTime '2014-09-22 >' := SYSTEMTIME_TO_STRING( FILETIME_TO_SYSTEMTIME( fileTime ) );
3 RETURN
  
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.1.20 TIME_TO_OTSTRUCT



Mit der Funktion "TIME_TO_OTSTRUCT" kann eine TIME-Konstante oder Variable in eine Struktur mit den aufgelösten Millisekunden, Sekunden, Minuten, Stunden, Tagen und Wochen konvertiert werden.

FUNCTION TIME_TO_OTSTRUCT : OTSTRUCT [[▶ 310](#)]

VAR_INPUT

```
VAR_INPUT
  TIN : TIME;
END_VAR
```

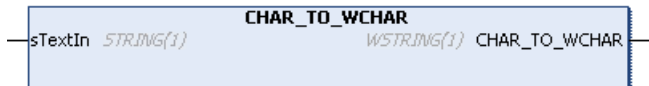

TIN: Die zu konvertierende TIME-Variable.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.2 Erweiterte STRING-Funktionen

4.2.1 CHAR_TO_WCHAR



Die Funktion konvertiert eine Variable vom Datentyp STRING in eine Variable vom Datentyp WSTRING (mit Null-Terminierung).

FUNCTION CHAR_TO_WCHAR : WSTRING(1)

VAR_INPUT

```
VAR_INPUT
    sTextIn : STRING(1);
END_VAR
```

sTextIn: STRING-Variable, die konvertiert werden soll.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.2 CONCAT2



Die Funktion verkettet zwei beliebig lange Zeichenfolgen vom Datentyp STRING und prüft, ob die resultierende Zeichenfolge länger als eine vorgegebene Ausgabe-Zeichenfolge ist. In diesem Fall wird die Zeichenfolge abgeschnitten.

Die Funktion liefert

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

Die Funktion stoppt die Überprüfung der Eingangslängen nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION CONCAT2 : BOOL

VAR_INPUT

```
VAR_INPUT
  pSrcString1 : POINTER TO STRING;
  pSrcString2 : POINTER TO STRING;
  pDstString  : POINTER TO STRING;
  nDstSize   : UDINT;
END_VAR
```

- pSrcString1:** Zeiger auf die erste der zu verkettenden STRING-Variable (Eingabe-Zeichenfolge)
- pSrcString2:** Zeiger auf die zweite der zu verkettenden STRING-Variable (Eingabe-Zeichenfolge)
- pDstString:** Zeiger auf die resultierende STRING-Variable nach der Verkettung (Ausgabe-Zeichenfolge)
- nDstSize:** Größe der resultierenden STRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator SIZEOF() verwendet werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.21.0

4.2.3 DATA_TO_HEXSTR2

Die Funktion konvertiert Binärdaten in einen Hexadezimalstring. Mit dieser Funktion können einfache Datentypen und Strukturvariablen konvertiert werden. Beim Überschreiten der maximal möglichen Länge des Ausgangs wird dem Ergebnisstring ein Punkt-Zeichen hinzugefügt ('.') und die Konvertierung abgebrochen. Die verbleibenden Datenbytes werden nicht mehr konvertiert.

FUNCTION DATA_TO_HEXSTR2 : UDINT

VAR_INPUT

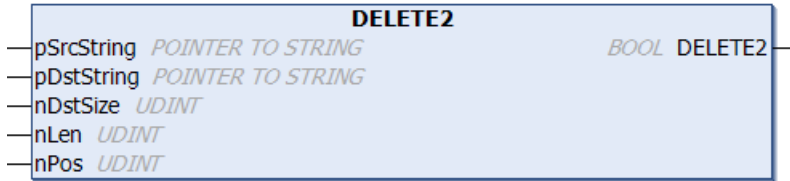
```
VAR_INPUT
  pSrcData      : POINTER TO BYTE; // pointer to data buffer
  nSrcSize      : UDINT;          // size of data buffer in bytes (= number of bytes to be converted)
  pDstHexStr    : POINTER TO STRING; // pointer to destination buffer
  nDstSize      : UDINT;          // size of destination buffer in bytes
  bLoCase       : BOOL;           // default: use "ABCDEF", if TRUE use "abcdef" characters
END_VAR
```

- pSrcData:** Anfangsadresse (Pointer) auf die zu konvertierenden Binärdaten. Die Adresse kann mit dem ADR-Operator ermittelt werden.
- nSrcSize:** Max. Größe (in Bytes) der zu konvertierenden Binärdaten. Die Größe kann mit dem SIZEOF-Operator ermittelt werden.
- pDstHexStr:** Anfangsadresse (Pointer) auf den Zielpuffer in den der konvertierte Hexadezimalstring geschrieben werden soll. Die Adresse kann mit dem ADR-Operator ermittelt werden.
- nDstSize:** Max. verfügbare Größe (in Bytes) des Zielpuffers. Die Größe kann mit dem SIZEOF-Operator ermittelt werden.
- bLoCase:** Dieser Parameter legt fest, ob Groß- oder Kleinbuchstaben bei der Konvertierung verwendet werden sollen. TRUE = Kleinbuchstaben, FALSE = Großbuchstaben.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.5.1.0

4.2.4 DELETE2



Die Funktion entfernt nLen Zeichen einer Zeichenfolge beginnend an der Position nPos.

Die Funktion liefert

- TRUE, wenn die Zeichen erfolgreich entfernt wurden.
- FALSE, wenn die resultierende Zeichenfolge länger ist als die Ausgabe-Zeichenfolge und nicht in den gegebenen Ausgangspuffer passt. Der Speicherbedarf der resultierenden Zeichenfolge ist dann größer als der der Ausgabe-Zeichenfolge. Die Zeichenfolge wird dann abgeschnitten.

Die Funktion stoppt die Überprüfung der Eingangslängen nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION DELETE2 : BOOL

VAR_INPUT

```
VAR_INPUT
    pSrcString : POINTER TO STRING;
    pDstString : POINTER TO STRING;
    nDstSize   : UDINT;
    nLen       : UDINT;
    nPos       : UDINT;
END_VAR
```

pSrcString: Zeiger auf die STRING-Variable (Eingabe-Zeichenfolge)

pDstString: Zeiger auf die resultierende STRING-Variable (Ausgabe-Zeichenfolge)

nDstSize: Größe der resultierenden STRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator `SIZEOF()` verwendet werden.

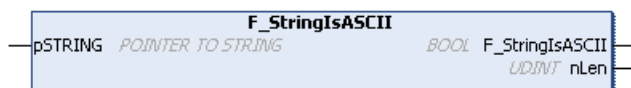
nLen: Anzahl der Zeichen, die entfernt werden sollen

nPos: Position des ersten Zeichens, das einschließlich der nachfolgenden Zeichen entfernt werden soll (nPos = 1 = erstes Zeichen)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.35.0

4.2.5 F_StringIsASCII



Die Funktion prüft, ob eine Zeichenfolge nur ASCII-Zeichen (0x000 bis 0x7F) enthält und gibt die Anzahl der ASCII-Zeichen zurück. Wenn die Zeichenfolge nur ASCII-Zeichen enthält, ist sie direkt kompatibel zu UTF-8.

Die Funktion stoppt die Überprüfung der Eingangslänge nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION F_StringIsASCII : BOOL

Der Rückgabewert ist TRUE, wenn die Zeichenfolge nur ASCII-Zeichen enthält.

VAR_INPUT

```
VAR_INPUT
  pSTRING : POINTER TO STRING;
END_VAR
```

pString: Zeiger auf die STRING-Variable.

VAR_OUTPUT

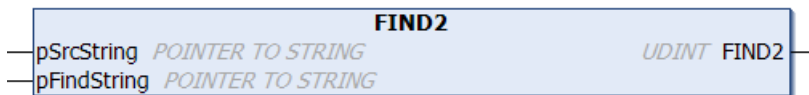
```
VAR_OUTPUT
  nLen : UDINT;
END_VAR
```

nLen: Anzahl der ASCII-Zeichen in der Zeichenfolge

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.21.0

4.2.6 FIND2



Die Funktion findet eine ggf. mehrfach vorkommende Zeichenfolge in einer anderen Zeichenfolge.

Die Funktion liefert

- die Position des ersten Zeichens der ersten gefundenen Zeichenfolge.
- den Wert 0, wenn die Zeichenfolge nicht gefunden wurde.

Die Funktion stoppt die Überprüfung der Eingangslängen nach Parameterlist.cMaxCharacters Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION FIND2 : UDINT

VAR_INPUT

```
VAR_INPUT
  pSrcString : POINTER TO STRING;
  pFindString : POINTER TO STRING;
END_VAR
```

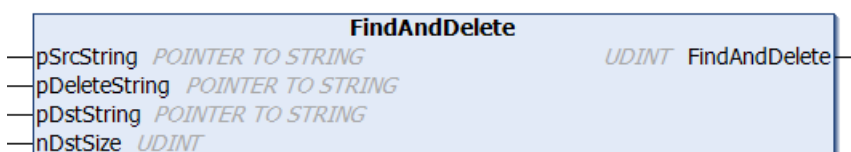
pSrcString: Zeiger auf die STRING-Variable, deren Zeichenfolge durchsucht werden soll

pFindString: Zeiger auf die STRING-Variable, deren Zeichenfolge gesucht wird

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.35.0

4.2.7 FindAndDelete



Die Funktion findet eine ggf. mehrfach vorkommende Zeichenfolge in einer anderen Zeichenfolge und entfernt diese.

Die Funktion liefert

- die Anzahl der entfernten Zeichenfolgen.
- den Wert 0, wenn die Zeichenfolge nicht gefunden wurde.

Die Funktion stoppt die Überprüfung der Eingangslängen nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION FindAndDelete : UDINT

VAR_INPUT

```
VAR_INPUT
  pSrcString      : POINTER TO STRING;
  pDeleteString   : POINTER TO STRING;
  pDstString      : POINTER TO STRING;
  nDstSize        : UDINT;
END_VAR
```

pSrcString: Zeiger auf die STRING-Variable, deren Zeichenfolge durchsucht werden soll (Eingabe-Zeichenfolge)

pDeleteString: Zeiger auf die STRING-Variable, deren Zeichenfolge gesucht und entfernt werden soll (Eingabe-Zeichenfolge)

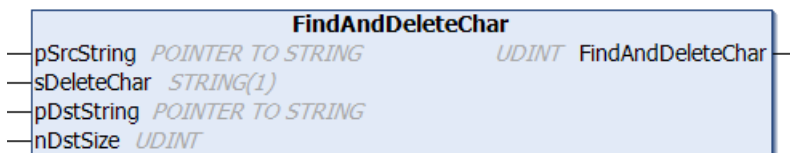
pDstString: Zeiger auf die resultierende STRING-Variable (Ausgabe-Zeichenfolge)

nDstSize: Größe der resultierenden STRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator `sizeof()` verwendet werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.35.0

4.2.8 FindAndDeleteChar



Die Funktion findet ein ggf. mehrfach vorkommendes Zeichen in einer Zeichenfolge und entfernt dieses.

Die Funktion liefert

- die Anzahl der entfernten Zeichen.
- den Wert 0, wenn das Zeichen nicht gefunden wurde.

Die Funktion stoppt die Überprüfung der Eingangslängen nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION FindAndDeleteChar : UDINT

VAR_INPUT

```
VAR_INPUT
  pSrcString      : POINTER TO STRING;
  sDeleteChar     : STRING(1);
  pDstString      : POINTER TO STRING;
  nDstSize        : UDINT;
END_VAR
```

pSrcString: Zeiger auf die STRING-Variable, deren Zeichenfolge durchsucht werden soll (Eingabe-Zeichenfolge)

sDeleteChar: Zeichen, das entfernt werden soll

pDstString: Zeiger auf die resultierende STRING-Variable (Ausgabe-Zeichenfolge)

nDstSize: Größe der resultierenden STRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator SIZEOF() verwendet werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.35.0

4.2.9 FindAndReplace



Die Funktion findet eine ggf. mehrfach vorkommende Zeichenfolge in einer anderen Zeichenfolge und ersetzt diese durch eine andere Zeichenfolge.

Die Funktion liefert

- die Anzahl der ersetzten Zeichenfolgen.
- den Wert 0, wenn die Zeichenfolge nicht gefunden wurde.

Die Funktion stoppt die Überprüfung der Eingangslängen nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION FindAndReplace : UDINT

VAR_INPUT

```

VAR_INPUT
  pSrcString      : POINTER TO STRING;
  pDeleteString  : POINTER TO STRING;
  pInsertString  : POINTER TO STRING;
  pDstString     : POINTER TO STRING;
  nDstSize       : UDINT;
END_VAR
    
```

pSrcString: Zeiger auf die STRING-Variable, deren Zeichenfolge durchsucht werden soll (Eingabe-Zeichenfolge)

pDeleteString: Zeiger auf die STRING-Variable, deren Zeichenfolge ersetzt werden soll (Eingabe-Zeichenfolge)

pInsertString: Zeiger auf die STRING-Variable, deren Zeichenfolge die andere Zeichenfolge ersetzen soll (Eingabe-Zeichenfolge)

pDstString: Zeiger auf die resultierende STRING-Variable (Ausgabe-Zeichenfolge)

nDstSize: Größe der resultierenden STRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator SIZEOF() verwendet werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.35.0

4.2.10 FindAndReplaceChar

FindAndReplaceChar	
pSrcString	POINTER TO STRING UDINT FindAndReplaceChar
sDeleteChar	STRING(1)
sInsertChar	STRING(1)
pDstString	POINTER TO STRING
nDstSize	UDINT

Die Funktion findet ein ggf. mehrfach vorkommendes Zeichen in einer Zeichenfolge und ersetzt dieses durch ein anderes Zeichen.

Die Funktion liefert

- die Anzahl der ersetzten Zeichen.
- den Wert 0, wenn das Zeichen nicht gefunden wurde.

Die Funktion stoppt die Überprüfung der Eingangslängen nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION FindAndReplaceChar : UDINT

VAR_INPUT

```

VAR_INPUT
  pSrcString      : POINTER TO STRING;
  sDeleteChar     : STRING(1);
  sInsertChar     : STRING(1);
  pDstString      : POINTER TO STRING;
  nDstSize        : UDINT;
END_VAR

```

pSrcString: Zeiger auf die STRING-Variable, deren Zeichenfolge durchsucht werden soll (Eingabe-Zeichenfolge)

sDeleteChar: Zeichen, das ersetzt werden soll

sInsertChar: Zeichen, das das andere Zeichen ersetzen soll

pDstString: Zeiger auf die resultierende STRING-Variable (Ausgabe-Zeichenfolge)

nDstSize: Größe der resultierenden STRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator `SIZEOF()` verwendet werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.35.0

4.2.11 FindAndSplit

FindAndSplit	
pSeparator	POINTER TO STRING BOOL FindAndSplit
pSrcString	POINTER TO STRING
pLeftString	POINTER TO STRING
nLeftSize	UDINT
pRightString	POINTER TO STRING
nRightSize	UDINT
bSearchFromRight	BOOL

Die Funktion zerlegt eine Zeichenfolge in zwei Zeichenfolgen.

Dabei wird die Zeichenfolge von links nach rechts nach einem Separator (z. B. „\“) durchsucht. Die Zeichenfolgen zu beiden Seiten dieses ersten Vorkommnisses des Separators werden ausgegeben.

Die Suchrichtung kann mittels des Parameters `bSearchFromRight` geändert werden, so dass die Zeichenfolge von rechts nach links durchsucht wird.

Die Funktion stoppt die Überprüfung der Eingangslänge nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION FindAndSplit : BOOL

Der Rückgabewert ist TRUE, wenn der Separator gefunden wurde sowie die Teilung und Ausgabe der Zeichenfolgen erfolgreich waren.

VAR_INPUT

```
VAR_INPUT
  pSeparator      : POINTER TO STRING;
  pSrcString      : POINTER TO STRING;
  pLeftString     : POINTER TO STRING;
  nLeftSize       : UDINT;
  pRightString    : POINTER TO STRING;
  nRightSize      : UDINT;
  bSearchFromRight : BOOL;
END_VAR
```

pSeparator: Zeiger auf die STRING-Variable, welche den Separator darstellt.

pSrcString: Zeiger auf die STRING-Variable, welche die Quell-Zeichenfolge darstellt.

pLeftString: Zeiger auf die STRING-Variable, in welcher die separierte linke Zeichenfolge ausgegeben werden soll.

nLeftSize: Maximale Größe der separierten linken Zeichenfolge.

pRightString: Zeiger auf die STRING-Variable, in welcher die separierte rechte Zeichenfolge ausgegeben werden soll.

nRightSize: Maximale Größe der separierten rechten Zeichenfolge.

bSearchFromRight: Falls der Eingang gesetzt ist, ändert sich die Suchrichtung, so dass die Zeichenfolge von rechts nach links nach dem Separator durchsucht wird.

Beispiel „Zerlegung in mehrere Teil-Zeichenfolgen“:

Dieses Beispiel zeigt, wie eine Zeichenfolge 'machines/machine1/module2/data/tx' in mehrere Zeichenfolgen ['machines', 'machine1', 'module2', 'data', 'tx'] zerlegt wird. Hierzu wird die Funktion FindAndSplit() wiederholt in einer Schleife aufgerufen.

```
PROGRAM MAIN
VAR
  sSrc      : STRING(255) := 'machines/machine1/module2/data/tx';
  sSeparator : STRING(1) := '/';
  aSplit    : ARRAY[1..cMax] OF STRING(255);
  bResultSplit : BOOL;
  i          : UDINT;
END_VAR
VAR CONSTANT
  cMax : UDINT := 9;
END_VAR

aSplit[1] := sSrc;
FOR i:=1 TO cMax-1 DO
  bResultSplit := FindAndSplit( pSeparator := ADR(sSeparator), pSrcString := ADR(aSplit[i]),
                                pLeftString := ADR(aSplit[i]),   nLeftSize := sizeof(aSplit[i])
  ),
                                pRightString := ADR(aSplit[i+1]), nRightSize := sizeof(aSplit[i+1])
  ),
                                bSearchFromRight := FALSE );
  IF NOT bResultSplit THEN
```



```

EXIT;
END_IF
END_FOR

```

Beispiel „Zusammenführen mehrerer Zeichenfolgen“:

Dieses Beispiel zeigt, wie mehrere Zeichenfolgen ['machines', 'machine1', 'module2', 'data', 'tx'] in einer Zeichenfolge 'machines/machine1/module2/data/tx' zusammengeführt werden.

```

PROGRAM MAIN
VAR
  sSeparator      : STRING(1) := '/';
  aSplit         : ARRAY[1..cMax] OF STRING(255) := ['machines', 'machine1', 'module2', 'data',
'tx'];
  sJoined        : STRING(255);
  bResultConcat  : BOOL;
  i              : UDINT;
END_VAR
VAR CONSTANT
  cMax           : UDINT := 5;
END_VAR

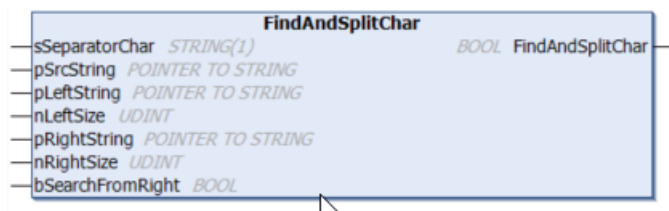
sJoined := aSplit[1];
FOR i:=1 TO cMax-1 DO
  bResultConcat := CONCAT2(pSrcString1 := ADR(sJoined), pSrcString2 := ADR(sSeparator), pDstString
:= ADR(sJoined), nDstSize := SIZEOF(sJoined));
  IF NOT bResultConcat THEN
    EXIT;
  END_IF
  bResultConcat := CONCAT2(pSrcString1 := ADR(sJoined), pSrcString2 := ADR(aSplit[i+1]), pDstString
:= ADR(sJoined), nDstSize := SIZEOF(sJoined));
  IF NOT bResultConcat THEN
    EXIT;
  END_IF
END_FOR

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024.11	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.39.0

4.2.12 FindAndSplitChar



Die Funktion zerlegt eine Zeichenfolge in zwei Zeichenfolgen.

Dabei wird die Zeichenfolge von links nach rechts nach einem Separator (z. B. „\“) durchsucht. Die Zeichenfolgen zu beiden Seiten dieses ersten Vorkommnisses des Separators werden ausgegeben.

Die Suchrichtung kann mittels des Parameters bSearchFromRight geändert werden, so dass die Zeichenfolge von rechts nach links durchsucht wird.

Die Funktion stoppt die Überprüfung der Eingangslänge nach Parameterlist.cMaxCharacters Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION FindAndSplitChar : BOOL

Der Rückgabewert ist TRUE, wenn der Separator gefunden wurde sowie die Teilung und Ausgabe der Zeichenfolgen erfolgreich waren.

VAR_INPUT

```

VAR_INPUT
  sSeparatorChar   : STRING(1);
  pSrcString       : POINTER TO STRING;
  pLeftString      : POINTER TO STRING;
  nLeftSize        : UDINT;
  pRightString     : POINTER TO STRING;
  nRightSize       : UDINT;
  bSearchFromRight : BOOL;
END_VAR

```

sSeparatorChar: Zeichen, welches den Separator darstellt.

pSrcString: Zeiger auf die STRING-Variable, welche die Quell-Zeichenfolge darstellt.

pLeftString: Zeiger auf die STRING-Variable, in welcher die separierte linke Zeichenfolge ausgegeben werden soll.

nLeftSize: Maximale Größe der separierten linken Zeichenfolge.

pRightString: Zeiger auf die STRING-Variable, in welcher die separierte rechte Zeichenfolge ausgegeben werden soll.

nRightSize: Maximale Größe der separierten rechten Zeichenfolge.

bSearchFromRight: Falls der Eingang gesetzt ist, ändert sich die Suchrichtung, so dass die Zeichenfolge von rechts nach links nach dem Separator durchsucht wird.

Beispiel „Zerlegung in mehrere Teil-Zeichenfolgen“:

Dieses Beispiel zeigt, wie eine Zeichenfolge 'machines/machine1/module2/data/tx' in mehrere Zeichenfolgen ['machines', 'machine1', 'module2', 'data', 'tx'] zerlegt wird. Hierzu wird die Funktion FindAndSplitChar() wiederholt in einer Schleife aufgerufen.

```

PROGRAM MAIN
VAR
  sSrc          : STRING(255) := 'machines/machine1/module2/data/tx';
  aSplit        : ARRAY[1..cMax] OF STRING(255);
  bResultSplit  : BOOL;
  i             : UDINT;
END_VAR
VAR CONSTANT
  cMax          : UDINT := 9;
END_VAR

aSplit[1] := sSrc;
FOR i:=1 TO cMax-1 DO
  bResultSplit := FindAndSplitChar( sSeparatorChar := '/', pSrcString := ADR(aSplit[i]),
  pLeftString   := ADR(aSplit[i]), nLeftSize := sizeof(aSplit[i]),
  pRightString  := ADR(aSplit[i+1]), nRightSize := sizeof(aSplit[i+1]),
  bSearchFromRight := FALSE );
  IF NOT bResultSplit THEN
    EXIT;
  END_IF
END_FOR

```

Beispiel „Zusammenführen mehrerer Zeichenfolgen“:

Dieses Beispiel zeigt, wie mehrere Zeichenfolgen ['machines', 'machine1', 'module2', 'data', 'tx'] in einer Zeichenfolge 'machines/machine1/module2/data/tx' zusammengeführt werden.

```

PROGRAM MAIN
VAR
  sSeparator      : STRING(1) := '/';
  aSplit          : ARRAY[1..cMax] OF STRING(255) := ['machines', 'machine1', 'module2', 'data',
  'tx'];
  sJoined         : STRING(255);
  bResultConcat  : BOOL;
  i              : UDINT;
END_VAR
VAR CONSTANT
  cMax          : UDINT := 5;
END_VAR

```

```

sJoined := aSplit[1];
FOR i:=1 TO cMax-1 DO
  bResultConcat := CONCAT2(pSrcString1 := ADR(sJoined), pSrcString2 := ADR(sSeparator), pDstString
:= ADR(sJoined), nDstSize := SIZEOF(sJoined));
  IF NOT bResultConcat THEN
    EXIT;
  END_IF
  bResultConcat := CONCAT2(pSrcString1 := ADR(sJoined), pSrcString2 := ADR(aSplit[i+1]), pDstString
:= ADR(sJoined), nDstSize := SIZEOF(sJoined));
  IF NOT bResultConcat THEN
    EXIT;
  END_IF
END_FOR

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024.11	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.39.0

4.2.13 HEXSTR_TO_DATA2

Die Funktion konvertiert einen Hexadezimalstring in Binärdaten und liefert die Anzahl der erfolgreich konvertierten Datenbytes als Ergebnis zurück. Es dürfen nur Leerzeichen als Trennzeichen in dem zu konvertierenden Hexadezimalstring verwendet werden. Bei den Hex-Zeichen sind Klein- und Großbuchstaben zugelassen. Bei einem Fehler oder unerlaubten Zeichen wird die Konvertierung abgebrochen und eine Null-Länge als Ergebnis zurückgeliefert.

FUNCTION HEXSTR_TO_DATA2 : UDINT

VAR_INPUT

```

VAR_INPUT
  pSrcHexStr : POINTER TO STRING; // hex string to convert (Example: "AF 34 55 EC")
  pDstData   : POINTER TO BYTE;   // pointer to destination buffer
  nDstSize   : UDINT;             // size of destination buffer in bytes
END_VAR

```

pSrcHexStr: Anfangsadresse (Pointer) auf den zu konvertierenden Hexadezimalstring (z.B.: 'AB CD 01 23'). Die Adresse kann mit dem ADR-Operator ermittelt werden.

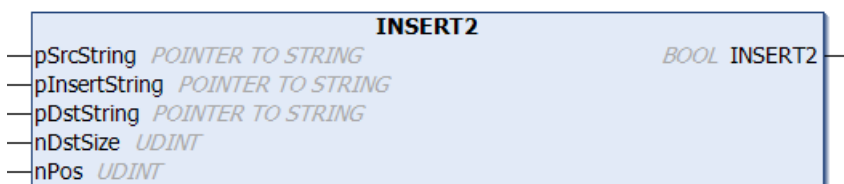
pDstData: Anfangsadresse (Pointer) auf den Zielpuffer in den die konvertierten Datenbytes geschrieben werden sollen. Die Adresse kann mit dem ADR-Operator ermittelt werden.

nDstSize: Max. verfügbare Größe (in Bytes) des Zielpuffers. Die Größe kann mit dem SIZEOF-Operator ermittelt werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.5.1.0

4.2.14 INSERT2



Die Funktion fügt eine Zeichenfolge in eine andere Zeichenfolge nach der Position nPos ein. Wenn nPos = 0 ist, wird die Zeichenfolge vor dem ersten Zeichen der anderen Zeichenfolge eingefügt.

Die Funktion liefert

- TRUE, wenn die Zeichenfolge erfolgreich eingefügt wurde.
- FALSE, wenn die resultierende Zeichenfolge länger ist als die Ausgabe-Zeichenfolge und nicht in den gegebenen Ausgangspuffer passt. Der Speicherbedarf der resultierenden Zeichenfolge ist dann größer als der der Ausgabe-Zeichenfolge. Die Zeichenfolge wird dann abgeschnitten.

Die Funktion stoppt die Überprüfung der Eingangslängen nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION INSERT2 : BOOL

VAR_INPUT

```
VAR_INPUT
  pSrcString      : POINTER TO STRING;
  pInsertString   : POINTER TO STRING;
  pDstString      : POINTER TO STRING;
  nDstSize        : UDINT;
  nPos            : UDINT;
END_VAR
```

pSrcString: Zeiger auf die STRING-Variable, deren Zeichenfolge durchsucht werden soll (Eingabe-Zeichenfolge)

pInsertString: Zeiger auf die STRING-Variable, deren Zeichenfolge in die andere Zeichenfolge eingefügt werden soll (Eingabe-Zeichenfolge)

pDstString: Zeiger auf die resultierende STRING-Variable (Ausgabe-Zeichenfolge)

nDstSize: Größe der resultierenden STRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator `sizeof()` verwendet werden.

nPos: Position des Zeichens, hinter dem die Zeichenfolge eingefügt werden soll

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.35.0

4.2.15 LEN2



Die Funktion gibt die Anzahl der Zeichen in einer Zeichenfolge zurück (Länge des STRINGS).

Die Funktion stoppt die Überprüfung der Eingangslänge nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION LEN2 : UDINT

VAR_INPUT

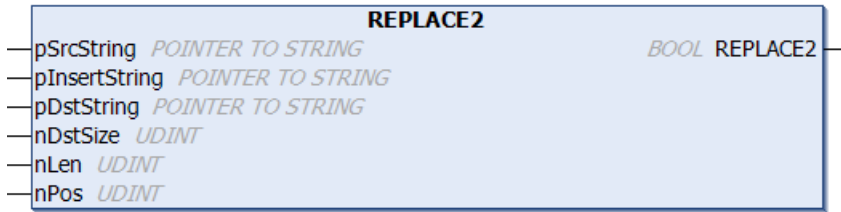
```
VAR_INPUT
  pSTRING : POINTER TO STRING;
END_VAR
```

pSTRING: Zeiger auf die STRING-Variable

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.16 REPLACE2



Die Funktion ersetzt nLen Zeichen einer Zeichenfolge durch eine andere Zeichenfolge beginnend an der Position nPos.

Die Funktion liefert

- TRUE, wenn die Zeichen erfolgreich ersetzt wurden.
- FALSE, wenn die resultierende Zeichenfolge länger ist als die Ausgabe-Zeichenfolge und nicht in den gegebenen Ausgangspuffer passt. Der Speicherbedarf der resultierenden Zeichenfolge ist dann größer als der der Ausgabe-Zeichenfolge. Die Zeichenfolge wird dann abgeschnitten.

Die Funktion stoppt die Überprüfung der Eingangslängen nach Parameterlist.cMaxCharacters Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION REPLACE2 : BOOL

VAR_INPUT

```
VAR_INPUT
    pSrcString      : POINTER TO STRING;
    pInsertString   : POINTER TO STRING;
    pDstString      : POINTER TO STRING;
    nDstSize        : UDINT;
    nLen            : UDINT;
    nPos            : UDINT;
END_VAR
```

pSrcString: Zeiger auf die STRING-Variable, deren Zeichenfolge durchsucht werden soll (Eingabe-Zeichenfolge)

pInsertString: Zeiger auf die STRING-Variable, deren Zeichenfolge die Zeichen ersetzen soll (Eingabe-Zeichenfolge)

pDstString: Zeiger auf die resultierende STRING-Variable (Ausgabe-Zeichenfolge)

nDstSize: Größe der resultierenden STRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator SIZEOF() verwendet werden.

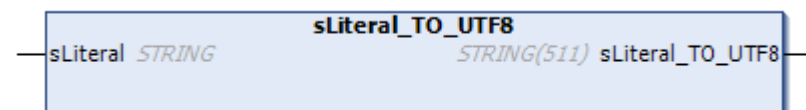
nLen: Anzahl der Zeichen, die ersetzt werden sollen

nPos: Position des Zeichens, das einschließlich der nachfolgenden Zeichen entfernt werden soll (nPos = 1 = erstes Zeichen)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.35.0

4.2.17 sLiteral_TO_UTF8



Die Funktion konvertiert eine beliebige Zeichenfolge vom Datentyp STRING in eine Zeichenfolge im UTF-8-Format. Die Funktion ist insbesondere für die Zuweisung von Literalen geeignet.

Bei der Zuweisung von Literalen zu einem UTF-8 STRING gilt:

- Literale, die nur den ASCII-Zeichensatz nutzen, können direkt zugewiesen werden.
- Literale, die den STRING-Zeichensatz nutzen, können mittels `sLiteral_TO_UTF8()` zugewiesen werden.
- Literale, die den WSTRING-Zeichensatz nutzen, können mittels `wsLiteral_TO_UTF8()` [► 182] zugewiesen werden.

Wenn das Literal länger ist als die mögliche Ausgabe-Zeichenfolge, wird ein Leerstring zurückgeliefert.

FUNCTION sLiteral_TO_UTF8 : STRING(511)

VAR_INPUT

```
VAR_IN_OUT CONSTANT
    sLiteral    : STRING;
END_VAR
```

sLiteral: STRING-Zeichenfolge, die konvertiert werden soll.

Beispiele

```
{attribute 'TcEncoding' := 'UTF-8'}
sMyText : STRING := sLiteral_TO_UTF8('Hühner legen Eier.');
```

```
{attribute 'TcEncoding' := 'UTF-8'}
sMyText1 : STRING := sLiteral_TO_UTF8('The dinner costs 30 €.);
```

● Dokumentation zum Attribut 'TcEncoding' := 'UTF-8'

i Für weitere Informationen zu Zeichenfolgen im UTF-8 Format siehe auch die Dokumentation zum Attribut 'TcEncoding'.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.34.0

4.2.18 STRING_TO_UTF8

```
STRING_TO_UTF8
pDstUTF8  PVOID          BOOL STRING_TO_UTF8
pSrcSTRING POINTER TO STRING
nDstSize  UDINT
```

Die Funktion konvertiert eine beliebige Zeichenfolge einer Variablen vom Datentyp STRING in eine Zeichenfolge im UTF-8-Format.

Die Funktion liefert

- TRUE, wenn die Konvertierung möglich war.
- FALSE, wenn die Konvertierung aufgrund der gegebenen Zeichensätze nicht möglich war.

Wenn die Eingabe-Zeichenfolge länger ist als die Ausgabe-Zeichenfolge, wird die Zeichenfolge abgeschnitten. Die Eingabe-Zeichenfolge ist zu lang, um auf die Ausgabe-Zeichenfolge codiert zu werden. Der Speicherbedarf der Ausgabe-Zeichenfolge kann bei der Konvertierung zu UTF-8 höher sein als der der Eingabe-Zeichenfolge.

Die Funktion stoppt die Konvertierung nach `Tc2_Utilities.Parameterlist.cMaxCharacters`. Bei entsprechender Parametrierung kann so eine Endlosschleife vermieden werden.

FUNCTION STRING_TO_UTF8 : BOOL**VAR_INPUT**

```

VAR_INPUT
  pDstUTF8      : PVOID;
  pSrcSTRING    : POINTER TO STRING;
  nDstSize      : UDINT;
END_VAR

```

pDstUTF8: Zeiger auf die Zeichenfolge im UTF-8-Format (Ausgabe-Zeichenfolge)

pSrcSTRING: Zeiger auf die zu konvertierende STRING-Variable (Eingabe-Zeichenfolge)

nDstSize: Größe der resultierenden Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator SIZEOF() verwendet werden.



Dokumentation zum Attribut 'TcEncoding' := 'UTF-8'

Für weitere Informationen zu Zeichenfolgen im UTF-8 Format siehe auch die Dokumentation zum Attribut 'TcEncoding'.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.19 STRING_TO_WSTRING2

```

STRING_TO_WSTRING2
  pDstWSTRING  POINTER TO WSTRING      BOOL  STRING_TO_WSTRING2
  pSrcSTRING   POINTER TO STRING
  nDstSize     UDINT

```

Die Funktion konvertiert eine Variable vom Datentyp STRING in eine Variable vom Datentyp WSTRING und prüft, ob die Eingabe-Zeichenfolge länger als die Ausgabe-Zeichenfolge ist. In diesem Fall wird die Zeichenfolge abgeschnitten.

Die Funktion liefert

- TRUE, wenn die Konvertierung der kompletten Zeichenfolge erfolgreich war.
- FALSE, wenn die Eingabe-Zeichenfolge länger als die Ausgabe-Zeichenfolge ist und das Ergebnis nicht in den gegebenen Ausgangspuffer passt. Der Speicherbedarf der Ausgabe-Zeichenfolge ist höher als der der Eingabe-Zeichenfolge. Die Zeichenfolge wird dann abgeschnitten.

Die Funktion stoppt die Konvertierung nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION STRING_TO_WSTRING2 : BOOL**VAR_INPUT**

```

VAR_INPUT
  pDstWSTRING  : POINTER TO WSTRING;
  pSrcSTRING   : POINTER TO STRING;
  nDstSize     : UDINT;
END_VAR

```

pDstWSTRING: Zeiger auf die konvertierte WSTRING-Variable (Ausgabe-Zeichenfolge)

pSrcSTRING: Zeiger auf die zu konvertierende STRING-Variable (Eingabe-Zeichenfolge)

nDstSize: Größe der resultierenden WSTRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator SIZEOF() verwendet werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.21.0

4.2.20 STRNCPY



Die Funktion kopiert die Zeichenfolge einer Variablen vom Datentyp STRING und prüft, ob die Zeichenfolge vollständig kopiert wurde.

Die Funktion liefert

- TRUE, wenn die Zeichenfolge (Inhalt des Quell-Arrays) vollständig kopiert werden konnte.
- FALSE, wenn die Zeichenfolge beim Kopieren abgeschnitten wurde. Wenn die Eingabe-Zeichenfolge länger als die Ausgabe-Zeichenfolge ist, wird nur die Anzahl an Zeichen (einschließlich Null-Terminierung) kopiert, die der Länge der Ausgabe-Zeichenfolge entspricht.

FUNCTION STRNCPY : BOOL

VAR_INPUT

```
VAR_INPUT
    pDst      : POINTER TO STRING;
    pSrc      : POINTER TO STRING;
    nDstSize  : UDINT;
END_VAR
```

pDst: Zeiger auf die kopierte STRING-Variable (Eingabe-Zeichenfolge)

pSrc: Zeiger auf die zu kopierende STRING-Variable (Ausgabe-Zeichenfolge)

nDstSize: Größe der resultierenden STRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator sizeof() verwendet werden.

VAR_OUTPUT

```
VAR_OUTPUT
    nSrcLen   : UDINT;
    nDstLen   : UDINT;
END_VAR
```

nSrcLen: Länge der angegebenen zu kopierenden STRING-Variablen

nDstLen: Länge der kopierten WSTRING-Variablen

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.21.0

4.2.21 UTF8_TO_STRING



Die Funktion konvertiert eine Zeichenfolge im UTF8-Format (Pointervariable vom Datentyp PVOID) in eine Zeichenfolge (Variable) vom Datentyp STRING.

Die Funktion liefert

- TRUE, wenn die Konvertierung möglich war.
- FALSE, wenn die Konvertierung aufgrund der gegebenen Zeichensätze nicht möglich war.

Wenn die Eingabe-Zeichenfolge länger als die Ausgabe-Zeichenfolge ist, wird die Zeichenfolge abgeschnitten. Unbekannte Zeichen werden übersprungen.

Die Funktion stoppt die Konvertierung nach `Tc2_Utilities.Parameterlist.cMaxCharacters`. Bei entsprechender Parametrierung kann so eine Endlosschleife vermieden werden.

FUNCTION UTF8_TO_STRING : BOOL

Der Rückgabewert ist TRUE, wenn die Konvertierung erfolgreich war.

VAR_INPUT

```
VAR_INPUT
  pDstSTRING : POINTER TO STRING;
  pSrcUTF8   : PVOID;
  nDstSize   : UDINT;
END_VAR
```

pDstSTRING: Zeiger auf die konvertierte STRING-Variable (Ausgabe-Zeichenfolge)

pSrcUTF8: Pointervariable (Eingabe-Zeichenfolge)

nDstSize: Größe der resultierenden STRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator `SIZEOF()` verwendet werden.

VAR_OUTPUT

```
VAR_OUTPUT
  nDstLen      : UDINT;
END_VAR
```

nDstLen: Tatsächliche Länge der Ausgabe-Zeichenfolge als Anzahl von Zeichen.

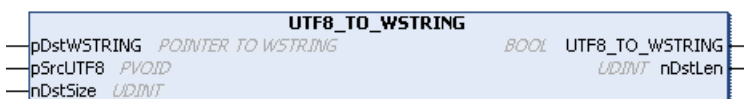
Dokumentation zum Attribut 'TcEncoding' := 'UTF-8'

I Für weitere Informationen zu Zeichenfolgen im UTF-8 Format siehe auch die Dokumentation zum Attribut 'TcEncoding'.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.22 UTF8_TO_WSTRING



Die Funktion konvertiert eine Zeichenfolge im UTF-8-Format in eine Zeichenfolge (Variable) vom Datentyp WSTRING.

Die Funktion liefert

- TRUE, wenn die Konvertierung möglich war.
- FALSE, wenn die Konvertierung aufgrund der gegebenen Zeichensätze nicht möglich war.

Wenn die Eingabe-Zeichenfolge länger als die Ausgabe-Zeichenfolge ist, wird die Zeichenfolge abgeschnitten. Die Eingabe-Zeichenfolge ist zu lang, um auf die Ausgabe-Zeichenfolge codiert zu werden. Der Speicherbedarf der Ausgabe-Zeichenfolge kann bei Konvertierung zu UTF-8 höher sein als der der Eingabe-Zeichenfolge. Unbekannte Zeichen werden übersprungen.

Die Funktion stoppt die Konvertierung nach `Tc2_Utilities.Parameterlist.cMaxCharacters`. Bei entsprechender Parametrierung kann so eine Endlosschleife vermieden werden.

FUNTION UTF8_TO_WSTRING : BOOL

Der Rückgabewert ist TRUE, wenn die Konvertierung erfolgreich war.

VAR_INPUT

```
VAR_INPUT
    pDstWSTRING : POINTER TO WSTRING;
    pSrcUTF8     : PVOID;
    nDstSize     : UDINT;
END_VAR
```

pDstWSTRING: Zeiger auf die konvertierte WSTRING-Variable (Ausgabe-Zeichenfolge)

pSrcUTF8: Pointervariable (Eingabe-Zeichenfolge)

nDstSize: Größe der resultierenden WSTRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator `sizeof()` verwendet werden.

VAR_OUTPUT

```
VAR_OUTPUT
    nDstLen : UDINT;
END_VAR
```

nDstLen: Tatsächliche Länge der Ausgabe-Zeichenfolge als Anzahl von Zeichen.

● Dokumentation zum Attribut 'TcEncoding' := 'UTF-8'

I Für weitere Informationen zu Zeichenfolgen im UTF-8 Format siehe auch die Dokumentation zum Attribut 'TcEncoding'.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.23 UTF8Len



Die Funktion gibt die Anzahl der Zeichen einer UTF-8-Zeichenfolge zurück.

Wenn die Zeichenfolge nicht mit dem UTF-8-Format übereinstimmt, gibt die Funktion den Wert 0 zurück.

Zusätzlich prüft die Funktion, ob alle Zeichen gültige ASCII-Zeichen sind und gibt dies über den Ausgang `bASCII` aus.

Die Funktion stoppt die Überprüfung nach `Tc2_Utilities.Parameterlist.cMaxCharacters`. Bei entsprechender Parametrierung kann so eine Endlosschleife vermieden werden.

FUNCTION UTF8Len : UDINT

Der Rückgabewert liefert die Anzahl der Zeichen in der UTF-8-Zeichenfolge.

VAR_INPUT

```
VAR_INPUT
    pUTF8 : PVOID;
END_VAR
```

pUTF8: Zeiger auf die null-terminierte UTF-8-Zeichenfolge

VAR_OUTPUT

```
VAR_OUTPUT
  bASCII      : BOOL;
  nSize       : UDINT;
END_VAR
```

bASCII: TRUE, wenn die UTF-8-Zeichen gültige ASCII-Zeichen sind.

nSize: Größe der Zeichenfolge in Bytes (ohne Null-Terminierung). Die Größe in Bytes kann je nach Zeichen größer sein als die Länge der Zeichenfolge.



Dokumentation zum Attribut 'TcEncoding' := 'UTF-8'

Für weitere Informationen zu Zeichenfolgen im UTF-8 Format siehe auch die Dokumentation zum Attribut 'TcEncoding'.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.21.0

4.2.24 WCHAR_TO_CHAR



Die Funktion konvertiert eine Variable vom Datentyp WSTRING in eine Variable vom Datentyp STRING (mit Null-Terminierung). Die Konvertierung ist nur möglich, wenn das WSTRING-Zeichen mit dem STRING-Zeichen übereinstimmt. Ansonsten wird kein Zeichen zurückgegeben.

FUNCTION WCHAR_TO_CHAR : STRING(1)

VAR_INPUT

```
VAR_INPUT
  sTextIn : WSTRING(1);
END_VAR
```

sTextIn: WSTRING-Variable, die konvertiert werden soll.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.21.0

4.2.25 WCONCAT2



Die Funktion verkettet zwei beliebig lange Zeichenfolgen vom Datentyp WSTRING und prüft, ob die resultierende Zeichenfolge länger als eine vorgegebene Ausgabe-Zeichenfolge ist. In diesem Fall wird die Zeichenfolge abgeschnitten.

Die Funktion liefert

- TRUE, wenn die Verkettung erfolgreich war.

- FALSE, wenn die resultierende Zeichenfolge länger ist als die Ausgabe-Zeichenfolge und nicht in den gegebenen Ausgangspuffer passt. Der Speicherbedarf der resultierenden Zeichenfolge ist größer als der der Ausgabe-Zeichenfolge. Die Zeichenfolge wird dann abgeschnitten.

Die Funktion stoppt die Überprüfung der Eingangslängen nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden

FUNCTION WCONCAT2 : BOOL

VAR_INPUT

```
VAR_INPUT
  pSrcWString1 : POINTER TO WSTRING;
  pSrcWString2 : POINTER TO WSTRING;
  pDstWString  : POINTER TO WSTRING;
  nDstSize     : UDINT;
END_VAR
```

pSrcWString1: Zeiger auf die erste der zu verkettenden WSTRING-Variable (Eingabe-Zeichenfolge)

pSrcWString2: Zeiger auf die zweite der zu verkettenden WSTRING-Variable (Eingabe-Zeichenfolge)

pDstWString: Zeiger auf die resultierende WSTRING-Variable nach der Verkettung (Ausgabe-Zeichenfolge)

nDstSize: Größe der resultierenden WSTRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator `sizeof()` verwendet werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.26 WLEN2



Die Funktion gibt die Anzahl der Zeichen in einer Unicode-Zeichenfolge vom Datentyp WSTRING zurück (Länge des WSTRINGs).

Die Funktion stoppt die Überprüfung der Eingangslänge nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION WLEN2 : UDINT

VAR_INPUT

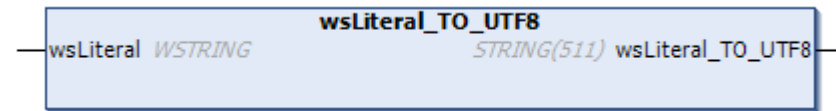
```
VAR_INPUT
  pWSTRING : POINTER TO WSTRING;
END_VAR
```

pWSTRING: Zeiger auf die WSTRING-Variable

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.27 wsLiteral_TO_UTF8



Die Funktion konvertiert eine beliebige Zeichenfolge vom Datentyp WSTRING in eine Zeichenfolge im UTF-8-Format. Die Funktion ist insbesondere für die Zuweisung von Literalen geeignet.

Bei der Zuweisung von Literalen zu einem UTF-8 STRING gilt:

- Literale, die nur den ASCII-Zeichensatz nutzen, können direkt zugewiesen werden.
- Literale, die den STRING-Zeichensatz nutzen, können mittels `sLiteral TO UTF8()` [174] zugewiesen werden.
- Literale, die den WSTRING-Zeichensatz nutzen, können mittels `wsLiteral_TO_UTF8()` zugewiesen werden.

Wenn das Literal länger ist als die mögliche Ausgabe-Zeichenfolge, wird ein Leerstring zurückgeliefert.

FUNCTION wsLiteral_TO_UTF8 : STRING(511)

VAR_INPUT

```
VAR_IN_OUT CONSTANT
  wsLiteral    : WSTRING;
END_VAR
```

sLiteral: WSTRING-Zeichenfolge, die konvertiert werden soll.

Beispiele

```
{attribute 'TcEncoding' := 'UTF-8'}
sMyText : STRING := wsLiteral_TO_UTF8("Hühner legen Eier.");
```

```
{attribute 'TcEncoding' := 'UTF-8'}
sMyText2 : STRING := wsLiteral_TO_UTF8("The dinner costs 30 €.");
```



Dokumentation zum Attribut 'TcEncoding' := 'UTF-8'

Für weitere Informationen zu Zeichenfolgen im UTF-8 Format siehe auch die Dokumentation zum Attribut 'TcEncoding'.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.34.0

4.2.28 WSTRING_TO_STRING2



Die Funktion konvertiert eine Variable vom Datentyp WSTRING in eine Variable vom Datentyp STRING.

Die Funktion liefert

- TRUE, wenn die Konvertierung der kompletten Zeichenfolge erfolgreich war.
- FALSE, wenn die Eingabe-Zeichenfolge länger als die Ausgabe-Zeichenfolge ist und das Ergebnis nicht in den gegebenen Ausgangspuffer passt.

Zeichen, die nicht konvertiert werden können, werden bei der Konvertierung übersprungen.

Die Funktion stoppt die Konvertierung nach `Parameterlist.cMaxCharacters` Zeichen, um eine Endlosschleife zu vermeiden.

FUNCTION WSTRING_TO_STRING2 : BOOL

VAR_INPUT

```
VAR_INPUT
  pDstString : POINTER TO STRING;
  pSrcWString : POINTER TO WSTRING;
  nDstSize : UDINT;
END_VAR
```

pDstSTRING: Zeiger auf die konvertierte STRING-Variable (Ausgabe-Zeichenfolge)

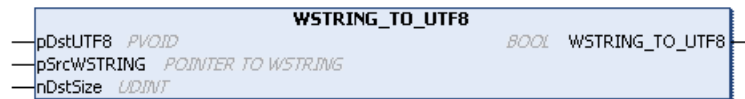
pSrcWSTRING: Zeiger auf die zu konvertierende WSTRING-Variable (Eingabe-Zeichenfolge)

nDstSize: Größe der resultierenden STRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator sizeof() verwendet werden.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.29 WSTRING_TO_UTF8



Die Funktion konvertiert eine Zeichenfolge einer Variablen vom Datentyp WSTRING in eine Zeichenfolge im UTF-8-Format.

Die Funktion liefert

- TRUE, wenn die Konvertierung möglich war.
- FALSE, wenn die Konvertierung aufgrund der gegebenen Zeichensätze nicht möglich war.

Wenn die Eingabe-Zeichenfolge länger als die Ausgabe-Zeichenfolge ist, wird die Zeichenfolge abgeschnitten. Die Eingabe-Zeichenfolge ist zu lang, um auf die Ausgabe-Zeichenfolge codiert zu werden. (Der Speicherbedarf der Ausgabe-Zeichenfolge kann bei Konvertierung zu UTF-8 höher als der der Eingabe-Zeichenfolge sein.) Unbekannte Zeichen werden übersprungen.

Die Funktion stoppt die Konvertierung nach `Tc2_Utilities.Parameterlist.cMaxCharacters`. Bei entsprechender Parametrierung kann so eine Endlosschleife vermieden werden.

FUNCTION WSTRING_TO_UTF8 : BOOL

VAR_INPUT

```
VAR_INPUT
  pDstUTF8 : PVOID;
  pSrcWSTRING : POINTER TO WSTRING;
  nDstSize : UDINT;
END_VAR
```

pDstUTF8: Pointervariable (Ausgabe-Zeichenfolge)

pSrcWSTRING: Zeiger auf die WSTRING-Variable (Eingabe-Zeichenfolge)

nDstSize: Größe der resultierenden Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator sizeof() verwendet werden.



Dokumentation zum Attribut 'TcEncoding' := 'UTF-8'

Für weitere Informationen zu Zeichenfolgen im UTF-8 Format siehe auch die Dokumentation zum Attribut 'TcEncoding'.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.21.0

4.2.30 WSTRNCPY

Die Funktion kopiert die Zeichenfolge einer Variablen vom Datentyp WSTRING und prüft, ob die Zeichenfolge vollständig kopiert wurde.

Die Funktion liefert

- TRUE, wenn die Zeichenfolge (Inhalt des Quell-Arrays) vollständig kopiert werden konnte.
- FALSE, wenn die Zeichenfolge beim Kopieren abgeschnitten wurde. Wenn die Eingabe-Zeichenfolge länger als die Ausgabe-Zeichenfolge ist, wird nur die Anzahl an Zeichen (einschließlich Null-Terminierung) kopiert, die der Länge der Ausgabe-Zeichenfolge entspricht.

FUNCTION WSTRNCPY : BOOL**VAR_INPUT**

```
VAR_INPUT
    pDst      : POINTER TO WSTRING;
    pSrc      : POINTER TO WSTRING;
    nDstSize  : UDINT;
END_VAR
```

pDst: Zeiger auf die kopierte WSTRING-Variable (Eingabe-Zeichenfolge)

pSrc: Zeiger auf die zu kopierende WSTRING-Variable (Ausgabe-Zeichenfolge)

nDstSize: Größe der resultierenden WSTRING-Variablen (Ausgabe-Zeichenfolge) in Bytes. Bei der Zuweisung kann der Operator sizeof() verwendet werden.

VAR_OUTPUT

```
VAR_OUTPUT
    nSrcLen   : UDINT;
    nDstLen   : UDINT;
END_VAR
```

nSrcLen: Länge der angegebenen zu kopierenden WSTRING-Variablen

nDstLen: Länge der kopierten WSTRING-Variablen

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.21.0

4.3 Byte-Order-Konvertierungsfunktionen**4.3.1 Host Byte Order / Network Byte Order**

Bei Netzwerkprotokollen wird die Byte-Reihenfolge festgeschrieben. Diese wird als Netzwerk-Byte-Reihenfolge (Network Byte Order) bezeichnet. Die natürliche Byte-Reihenfolge des TwinCAT-Systems wird als Host-Byte-Reihenfolge (Host Byte Order) bezeichnet. In den meisten Fällen entspricht die geforderte Netzwerk-Byte-Reihenfolge dem Big-Endian-Format (MOTOROLA). Das TwinCAT-SPS-System arbeitet

aber mit dem Little-Endian-Format (INTEL). Damit ein fehlerfreier Datenaustausch zwischen dem TwinCAT-SPS-System und einer anderen Plattform stattfinden kann muss die Byte-Reihenfolge im Anwendungsprogramm entsprechend umgewandelt werden.

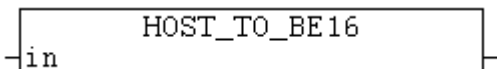
Daten, die über ein Netzwerkprotokoll vom TwinCAT System (Host) zu einem Fremdsystem übertragen werden sollen, können mit folgenden Funktionen ins Netzwerk-Format konvertiert werden:

- [HOST TO BE16](#) [▶ 185]
- [HOST TO BE32](#) [▶ 185]
- [HOST TO BE64](#) [▶ 186]
- [HOST TO BE64EX](#) [▶ 186]
- [HOST TO BE128](#) [▶ 187]

Die empfangenen Netzwerkdaten (Fremdsystem) können wiederum mit folgenden Funktionen ins Host-Format (TwinCAT-System) konvertiert werden:

- [BE16 TO HOST](#) [▶ 187]
- [BE32 TO HOST](#) [▶ 188]
- [BE64 TO HOST](#) [▶ 188]
- [BE64 TO HOSTEX](#) [▶ 188]
- [BE128 TO HOST](#) [▶ 189]

4.3.2 HOST_TO_BE16



Die Funktion führt eine Host-To-Network-Umwandlung einer 16 Bit Zahl durch. Siehe auch unter: [Byte Order](#) [▶ 184].

FUNCTION HOST_TO_BE16 : WORD

VAR_INPUT

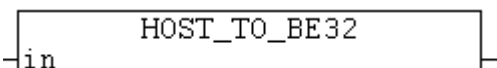
```
VAR_INPUT
  in : WORD;
END_VAR
```

in: Die zu konvertierende Zahl.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.3 HOST_TO_BE32



Die Funktion führt eine Host-To-Network-Umwandlung einer 32 Bit Zahl durch. Siehe auch unter: [Byte Order](#) [▶ 184].

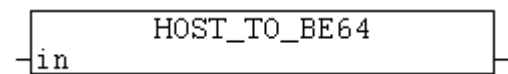
FUNCTION HOST_TO_BE32 : DWORD**VAR_INPUT**

```
VAR_INPUT
  in : DWORD;
END_VAR
```

in: Die zu konvertierende Zahl.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.4 HOST_TO_BE64

Die Funktion führt eine Host-To-Network-Umwandlung einer 64 Bit Zahl durch („legacy“-Typ: [T_ULARGE_INTEGER](#) [▶ 327]). Siehe auch unter: [Byte Order](#) [▶ 184].

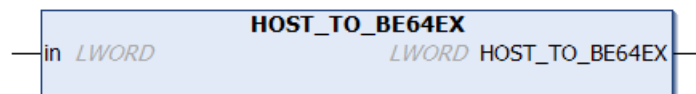
FUNCTION HOST_TO_BE64 : T_ULARGE_INTEGER**VAR_INPUT**

```
VAR_INPUT
  in : T_ULARGE_INTEGER;
END_VAR
```

in: Die zu konvertierende Zahl (Typ: [T_ULARGE_INTEGER](#) [▶ 327]).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.5 HOST_TO_BE64EX

Die Funktion führt eine Host-To-Network-Umwandlung einer 64 Bit Zahl durch („native“-Typ: LWORD). Siehe auch unter: [Byte Order](#) [▶ 184].

FUNCTION HOST_TO_BE64EX : LWORD**VAR_INPUT**

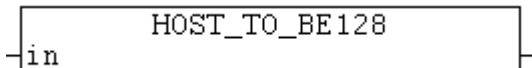
```
VAR_INPUT
  in : LWORD;
END_VAR
```

in: Die zu konvertierende Zahl.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.3.6 HOST_TO_BE128



Die Funktion führt eine Host-To-Network-Umwandlung einer 128 Bit Zahl durch („legacy“-Typ: [T_UHUGE_INTEGER \[▶ 326\]](#)). Siehe auch unter: [Byte Order \[▶ 184\]](#).

FUNCTION HOST_TO_BE128 : T_UHUGE_INTEGER

VAR_INPUT

```

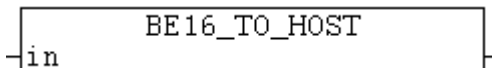
VAR_INPUT
  in : T_UHUGE_INTEGER;
END_VAR
  
```

in: Die zu konvertierende Zahl (Typ: [T_UHUGE_INTEGER \[▶ 326\]](#)).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.3.7 BE16_TO_HOST



Die Funktion führt eine Network-To-Host-Umwandlung einer 16 Bit Zahl durch. Siehe auch unter: [Byte Order \[▶ 184\]](#).

FUNCTION BE16_TO_HOST : WORD

VAR_INPUT

```

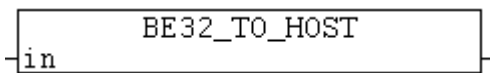
VAR_INPUT
  in : WORD;
END_VAR
  
```

in: Die zu konvertierende Zahl.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.3.8 BE32_TO_HOST



Die Funktion führt eine Network-To-Host-Umwandlung einer 32 Bit Zahl durch. Siehe auch unter: [Byte Order](#) [► 184].

FUNCTION BE32_TO_HOST : DWORD

VAR_INPUT

```

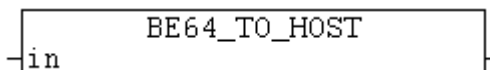
VAR_INPUT
  in : DWORD;
END_VAR
  
```

in: Die zu konvertierende Zahl.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.3.9 BE64_TO_HOST



Die Funktion führt eine Network-To-Host-Umwandlung einer 64 Bit Zahl durch („legacy“-Typ: [T_ULARGE_INTEGER](#) [► 327]). Siehe auch unter: [Byte Order](#) [► 184].

FUNCTION BE64_TO_HOST : T_ULARGE_INTEGER

VAR_INPUT

```

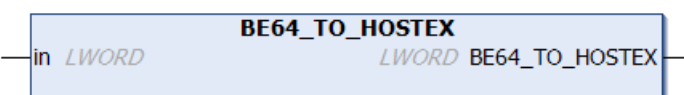
VAR_INPUT
  in : T_ULARGE_INTEGER;
END_VAR
  
```

in: Die zu konvertierende Zahl (Typ: [T_ULARGE_INTEGER](#) [► 327]).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.3.10 BE64_TO_HOSTEX



Die Funktion führt eine Network-To-Host-Umwandlung einer 64 Bit Zahl durch („native“-Typ: LWORD). Siehe auch unter: [Byte Order](#) [► 184].

FUNCTION BE64_TO_HOSTEX : LWORD

VAR_INPUT

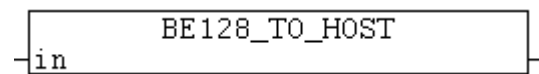
```
VAR_INPUT
  in : LWORD;
END_VAR
```

in: Die zu konvertierende Zahl.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.11 BE128_TO_HOST



Die Funktion führt eine Network-To-Host-Umwandlung einer 128 Bit Zahl durch („legacy“-Typ: [T_UHUGE_INTEGER \[▶ 326\]](#)). Siehe auch unter: [Byte Order \[▶ 184\]](#).

FUNCTION BE128_TO_HOST : T_UHUGE_INTEGER

VAR_INPUT

```
VAR_INPUT
  in : T_UHUGE_INTEGER;
END_VAR
```

in: Die zu konvertierende Zahl (Typ: [T_UHUGE_INTEGER \[▶ 326\]](#)).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4 FLOAT-Funktionen

4.4.1 BOOL_TO_FLOAT

FUNCTION BOOL_TO_FLOAT : LREAL

VAR_INPUT

```
VAR_INPUT
  in : BOOL;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.2 DINT_TO_FLOAT

FUNCTION DINT_TO_FLOAT : FLOAT

VAR_INPUT

```
VAR_INPUT
  in      : DINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.3 FLOAT_TO_BOOL

FUNCTION FLOAT_TO_BOOL : BOOL

VAR_INPUT

```
VAR_INPUT
  in      : LREAL;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.4 FLOAT_TO_DINT

FUNCTION FLOAT_TO_DINT : DINT

VAR_INPUT

```
VAR_INPUT
  in      : FLOAT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.5 FLOAT_TO_INT

FUNCTION FLOAT_TO_INT : INT

VAR_INPUT

```
VAR_INPUT
  in      : FLOAT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.4.6 FLOAT_TO_SINT

FUNCTION FLOAT_TO_SINT : SINT

VAR_INPUT

```
VAR_INPUT
    in          : FLOAT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.4.7 FLOAT_TO_STRING

FUNCTION FLOAT_TO_STRING : STRING

VAR_INPUT

```
VAR_INPUT
    in          : FLOAT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.4.8 FLOAT_TO_TIME

FUNCTION FLOAT_TO_TIME : TIME

VAR_INPUT

```
VAR_INPUT
    in          : FLOAT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.4.9 FLOAT_TO_UDINT

FUNCTION FLOAT_TO_UDINT : UDINT

VAR_INPUT

```
VAR_INPUT
    in          : FLOAT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.10 FLOAT_TO_UINT

FUNCTION FLOAT_TO_UINT : UINT

VAR_INPUT

```
VAR_INPUT
    in          : FLOAT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.11 INT_TO_FLOAT

FUNCTION INT_TO_FLOAT : FLOAT

VAR_INPUT

```
VAR_INPUT
    in          : INT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.12 SINT_TO_FLOAT

FUNCTION SINT_TO_FLOAT : FLOAT

VAR_INPUT

```
VAR_INPUT
    in          : SINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.4.13 TIME_TO_FLOAT

FUNCTION TIME_TO_FLOAT : FLOAT

VAR_INPUT

```
VAR_INPUT
    in:      TIME;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.4.14 UDINT_TO_FLOAT

FUNCTION UDINT_TO_FLOAT : FLOAT

VAR_INPUT

```
VAR_INPUT
    in      : UDINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.4.15 UINT_TO_FLOAT

FUNCTION UINT_TO_FLOAT : FLOAT

VAR_INPUT

```
VAR_INPUT
    in      : UINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.4.16 LrealsFinite

FUNCTION LrealsFinite : BOOL

VAR_INPUT

```
VAR_INPUT
  x          : REFERENCE TO LREAL;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4020	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >=3.3.16.0

4.4.17 LrealsNaN

Diese Funktion testet, ob ein Wert NaN (Not-a-Number) ist. Ist der Rückgabewert TRUE, dann ist der Wert NaN.

```
FUNCTION LrealIsNaN : BOOL
VAR_INPUT
  x          : REFERENCE TO LREAL;
END_VAR
```

Zu den wesentlichen Eigenschaften von NaN-Werten zählen die folgenden Punkte:

- Alle arithmetischen Operationen, die NaN als Eingangsdaten verwenden, liefern wiederum NaN als Ergebnis.
- Alle relationalen Operatoren =, !=, > < >= <= liefern stets den Wert False, wenn mindestens einer der Operanden NaN ist.
- Die Standard-C-Funktion `isnan()` bzw. `_isnan()` oder die SPS Funktion `LrealsNaN()` [► 194] (Tc2_Uilities Bibliothek) liefert den Wert True, wenn das Argument den Wert NaN hat.
- Der Ausdruck `isnan(a)` ist äquivalent dem Ausdruck `!(a == a)` bzw. `NOT(a = a)`.

Die Tatsache, dass NaN-Werte sich bei der Verwendung in weiteren Berechnungen fortpflanzen, hat den Vorteil, dass ungültige Werte nicht übersehen werden können.

⚠ VORSICHT

Fehlfunktionen der Software

NaN-Werte dürfen in SPS-Bibliotheken, insbesondere als Stellwerte in Funktionen für Motion Control und zur Antriebssteuerung, nur verwendet werden, wenn sie ausdrücklich zugelassen sind! Anderenfalls können NaN-Werte zu potenziell gefährlichen Fehlfunktionen der betreffenden Software führen!

⚠ VORSICHT

Floating Point Exceptions

Falls NaNs in der Applikation verwendet und verarbeitet werden sollen, müssen die FP Exceptions ausgeschaltet werden. Andernfalls können Vergleiche mit NaN zu einer Exception führen, welche einen Stopp der Laufzeit und möglichen Maschinenschaden nach sich zieht.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4020	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >=3.3.16.0

4.4.18 ReallsFinite

FUNCTION ReallsFinite : BOOL

VAR_INPUT

```
VAR_INPUT
  x      : REFERENCE TO REAL;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024.32	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.50.0

4.4.19 ReallsNaN

Diese Funktion testet, ob ein Wert NaN (Not-a-Number) ist. Ist der Rückgabewert TRUE, dann ist der Wert NaN.

```
FUNCTION RealIsNaN : BOOL
VAR_INPUT
  x      : REFERENCE TO REAL;
END_VAR
```

Zu den wesentlichen Eigenschaften von NaN-Werten zählen die folgenden Punkte:

- Alle arithmetischen Operationen, die NaN als Eingangsdaten verwenden, liefern wiederum NaN als Ergebnis.
- Alle relationalen Operatoren =, !=, > < >= <= liefern stets den Wert False, wenn mindestens einer der Operanden NaN ist.
- Die Standard-C-Funktion `isnan()` bzw. `_isnan()` oder die SPS Funktion `LreallsNaN()` [► 194] (Tc2_Utilities Bibliothek) liefert den Wert True, wenn das Argument den Wert NaN hat.
- Der Ausdruck `isnan(a)` ist äquivalent dem Ausdruck `!(a == a)` bzw. `NOT(a = a)`.

Die Tatsache, dass NaN-Werte sich bei der Verwendung in weiteren Berechnungen fortpflanzen, hat den Vorteil, dass ungültige Werte nicht übersehen werden können.

⚠ VORSICHT

Fehlfunktionen der Software
 NaN-Werte dürfen in SPS-Bibliotheken, insbesondere als Stellwerte in Funktionen für Motion Control und zur Antriebssteuerung, nur verwendet werden, wenn sie ausdrücklich zugelassen sind! Anderenfalls können NaN-Werte zu potenziell gefährlichen Fehlfunktionen der betreffenden Software führen!

⚠ VORSICHT

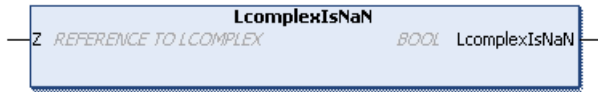
Floating Point Exceptions
 Falls NaNs in der Applikation verwendet und verarbeitet werden sollen, müssen die FP Exceptions ausgeschaltet werden. Andernfalls können Vergleiche mit NaN zu einer Exception führen, welche einen Stopp der Laufzeit und möglichen Maschinenschaden nach sich zieht.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024.32	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.50.0

4.5 LCOMPLEX-Funktionen

4.5.1 LcomplexIsNaN



Die Funktion liefert TRUE, wenn das Argument vom Typ LCOMPLEX einen undefinierten Wert besitzt (NaN).

FUNCTION LcomplexIsNaN : BOOL

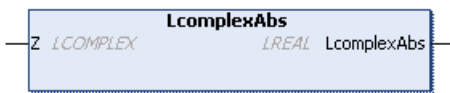
VAR_INPUT

```
VAR_INPUT
  Z : REFERENCE TO LCOMPLEX;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4020	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >=3.3.16.0

4.5.2 LcomplexAbs



Die Funktion liefert den Absolutwert zur übergebenen komplexen Zahl.

FUNCTION LcomplexAbs : LREAL

VAR_INPUT

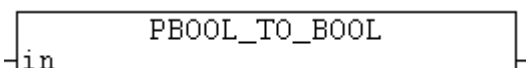
```
VAR_INPUT
  Z : LCOMPLEX;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.21.0

4.6 P[TYPE]_TO_[TYPE]-Konvertierungsfunktionen

4.6.1 PBOOL_TO_BOOL



Die Funktion liefert den Inhalt einer BOOL-Pointervariablen.

FUNCTION PBOOL_TO_BOOL: BOOL

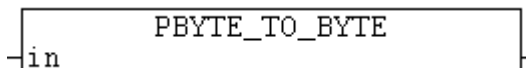
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO BOOL;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.2 PBYTE_TO_BYTE



Die Funktion liefert den Inhalt einer BYTE-Pointervariablen.

FUNCTION PBYTE_TO_BYTE: BYTE

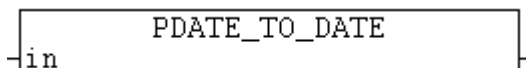
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO BYTE;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.3 PDATE_TO_DATE



Die Funktion liefert den Inhalt einer DATE-Pointervariablen.

FUNCTION PDATE_TO_DATE: DATE

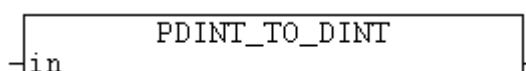
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO DATE;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.4 PDINT_TO_DINT



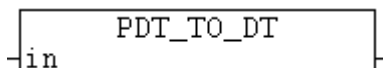
Die Funktion liefert den Inhalt einer DINT-Pointervariablen.

FUNCTION PDINT_TO_DINT: DINT**VAR_INPUT**

```
VAR_INPUT
  in : POINTER TO DINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.5 PDT_TO_DT

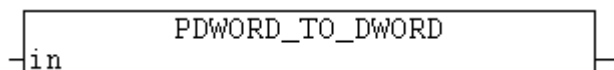
Die Funktion liefert den Inhalt einer DT-Pointervariablen.

FUNCTION PDT_TO_DT: DT**VAR_INPUT**

```
VAR_INPUT
  in : POINTER TO DT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.6 PDWORD_TO_DWORD

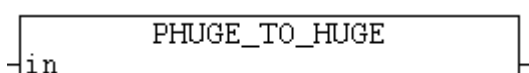
Die Funktion liefert den Inhalt einer DWORD-Pointervariablen.

FUNCTION PDWORD_TO_DWORD: DWORD**VAR_INPUT**

```
VAR_INPUT
  in : POINTER TO DWORD;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.7 PHUGE_TO_HUGE

Die Funktion liefert den Inhalt einer T_HUGE_INTEGER [► 326]-Pointervariablen.

FUNCTION PHUGE_TO_HUGE: T_HUGE_INTEGER

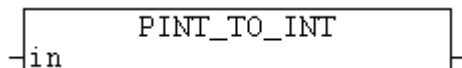
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO T_HUGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.8 PINT_TO_INT



Die Funktion liefert den Inhalt einer INT-Pointervariablen.

FUNCTION PINT_TO_INT: INT

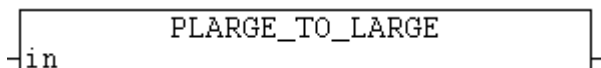
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO INT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.9 PLARGE_TO_LARGE



Die Funktion liefert den Inhalt einer [T_LARGE_INTEGER](#) [► 326]-Pointervariablen.

FUNCTION PLARGE_TO_LARGE: T_LARGE_INTEGER

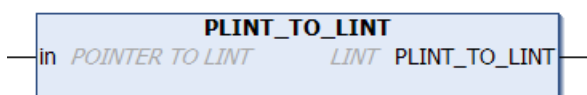
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO T_LARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.10 PLINT_TO_LINT



Die Funktion liefert den Inhalt einer LINT-Pointervariablen.

FUNCTION PLINT_TO_LINT : LINT

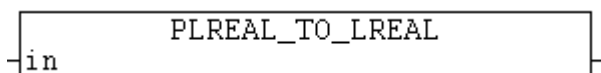
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO LINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.11 PLREAL_TO_LREAL



Die Funktion liefert den Inhalt einer LREAL-Pointervariablen.

FUNCTION PLREAL_TO_LREAL: LREAL

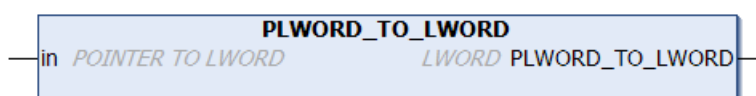
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO LREAL;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.12 PLWORD_TO_LWORD



Die Funktion liefert den Inhalt einer LWORD-Pointervariablen.

FUNCTION PLWORD_TO_LWORD : LWORD

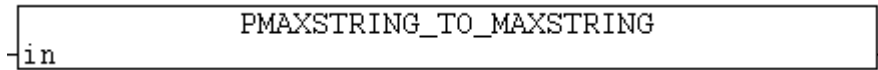
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO LWORD;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.13 PMAXSTRING_TO_MAXSTRING



Die Funktion liefert den Inhalt einer T_MaxString-Pointervariablen.

FUNCTION PMAXSTRING_TO_MAXSTRING: T_MaxString

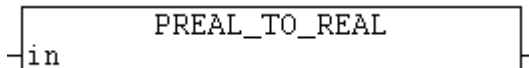
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO T_MaxString;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.14 PREAL_TO_REAL



Die Funktion liefert den Inhalt einer REAL-Pointervariablen.

FUNCTION PREAL_TO_REAL: REAL

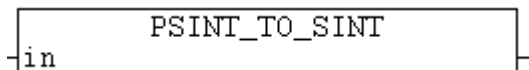
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO REAL;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.15 PSINT_TO_SINT



Die Funktion liefert den Inhalt einer SINT-Pointervariablen.

FUNCTION PSINT_TO_SINT: SINT

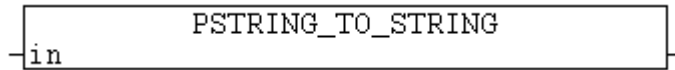
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO SINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.16 PSTRING_TO_STRING



Die Funktion liefert den Inhalt einer STRING-Pointervariablen.

FUNCTION PSTRING_TO_STRING: STRING

VAR_INPUT

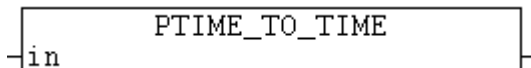
```

VAR_INPUT
  in : POINTER TO STRING;
END_VAR
  
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.17 PTIME_TO_TIME



Die Funktion liefert den Inhalt einer TIME-Pointervariablen.

FUNCTION PTIME_TO_TIME: TIME

VAR_INPUT

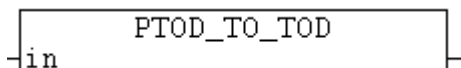
```

VAR_INPUT
  in : POINTER TO TIME;
END_VAR
  
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.18 PTOD_TO_TOD



Die Funktion liefert den Inhalt einer TOD-Pointervariablen.

FUNCTION PTOD_TO_TOD: TOD

VAR_INPUT

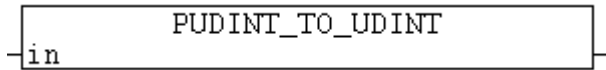
```

VAR_INPUT
  in : POINTER TO TOD;
END_VAR
  
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.19 PUDINT_TO_UDINT



Die Funktion liefert den Inhalt einer UDINT-Pointervariablen.

FUNCTION PUDINT_TO_UDINT: UDINT

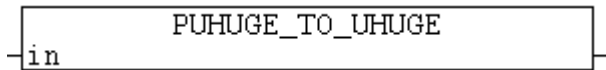
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO UDINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.20 PUHUGE_TO_UHUGE



Die Funktion liefert den Inhalt einer T_UHUGE_INTEGER [▶ 326]-Pointervariablen.

FUNCTION PUHUGE_TO_UHUGE: T_UHUGE_INTEGER

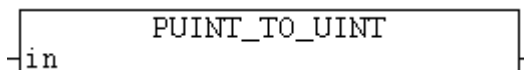
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO T_UHUGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.21 PUINT_TO_UINT



Die Funktion liefert den Inhalt einer UINT-Pointervariablen.

FUNCTION PUINT_TO_UINT: UINT

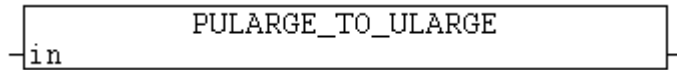
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO UINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.22 PULARGE_TO_ULARGE



Die Funktion liefert den Inhalt einer `T_ULARGE_INTEGER` [► 327]-Pointervariablen.

FUNCTION PULARGE_TO_ULARGE: T_ULARGE_INTEGER

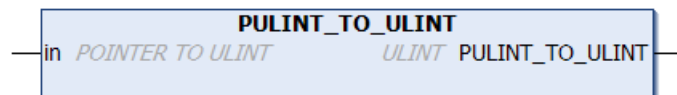
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.23 PULINT_TO_ULINT



Die Funktion liefert den Inhalt einer ULINT-Pointervariablen.

FUNCTION PULINT_TO_ULINT : ULINT

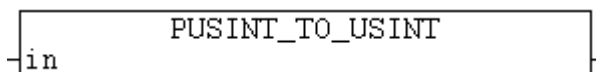
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO ULINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.6.24 PUSINT_TO_USINT



Die Funktion liefert den Inhalt einer USINT-Pointervariablen.

FUNCTION PUSINT_TO_USINT: USINT

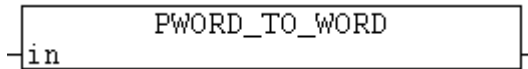
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO USINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.25 PWORD_TO_WORD



Die Funktion liefert den Inhalt einer WORD-Pointervariablen.

FUNCTION PWORD_TO_WORD: WORD

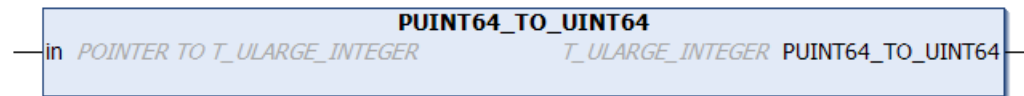
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO WORD;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.26 PUINT64_TO_UINT64



Die Funktion PUINT64_TO_UINT64 liefert den Inhalt einer T_ULARGE_INTEGER [▶ 327]-Pointervariablen.

FUNCTION PUINT64_TO_UINT64: T_ULARGE_INTEGER

VAR_INPUT

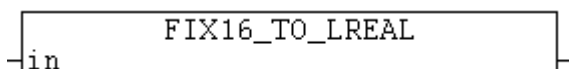
```
VAR_INPUT
    in : POINTER TO T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7 16-Bit-Festkommazahl-Funktionen (signed)

4.7.1 FIX16_TO_LREAL



Konvertiert eine vorzeichenbehaftete 16 Bit Festkommazahl in eine Fließkommazahl von Typ: LREAL.

FUNCTION FIX16_TO_LREAL : LREAL**VAR_INPUT**

```
VAR_INPUT
  in : T_FIX16;
END_VAR
```

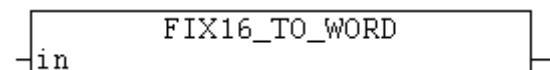
in: Die zu konvertierende Festkommazahl (Typ: [T_FIX16](#) [[▶ 324](#)]).

Beispiel:

Siehe in der Beschreibung der Funktion: [LREAL_TO_FIX16](#) [[▶ 210](#)].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.7.2 FIX16_TO_WORD

Diese Funktion konvertiert eine 16 Bit Festkommazahl in eine WORD-Variable (die WORD-Variable beinhaltet die Vor- und Nachkommastellen der Festkommazahl).

FUNCTION FIX16_TO_WORD : WORD**VAR_INPUT**

```
VAR_INPUT
  in : T_FIX16;
END_VAR
```

in: Die zu konvertierende Festkommazahl (Typ: [T_FIX16](#) [[▶ 324](#)]).

Beispiel:

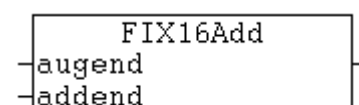
```
PROGRAM FIX_TO_WORD
VAR
  fp16 : WORD;
END_VAR

fp16 := FIX16_TO_WORD(LREAL_TO_FIX16(12.5, 8));
```

Der Wert der *fp16*-Variablen ist: 2#0000110010000000.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.7.3 FIX16Add

Diese Funktion addiert zwei vorzeichenbehaftete 16 Bit Festkommazahlen. Die Auflösung (Anzahl der Nachkommastellen) der Zahlen muss nicht gleich sein. Die Auflösung der Zahl mit der höheren Anzahl an Nachkommastellen wird vor der Addition reduziert. D.h. die Nachkommastellen der Zahl mit der höheren Auflösung werden abgeschnitten. Das Ergebnis der Addition ist eine vorzeichenbehaftete 16 Bit Festkommazahl.

FUNCTION FIX16Add : T_FIX16 [► 324]

VAR_INPUT

```
VAR_INPUT
    augend : T_FIX16;
    addend : T_FIX16;
END_VAR
```

augend: Der erste Summand (Typ: T_FIX16 [► 324]).

addend: Der zweite Summand (Typ: T_FIX16 [► 324]).

Beispiel:

```
PROGRAM FIXADD
VAR
    a, b : T_FIX16;
    result : LREAL;
END_VAR

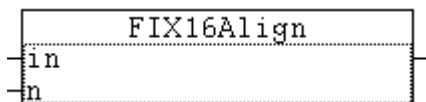
a := LREAL_TO_FIX16( 0.5, 8 );
b := LREAL_TO_FIX16( -0.25, 8 );

result := FIX16_TO_LREAL( FIX16Add( a, b ) ); (* The result is: 0.25 *)
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7.4 FIX16Align



Mit dieser Funktion kann die Auflösung (Anzahl der Nachkommastellen) einer vorzeichenbehafteten 16 Bit Festkommazahl geändert werden. Die neue Festkommazahl wird von der Funktion als Rückgabeparameter geliefert.

FUNCTION FIX16Align: T_FIX16 [► 324]

VAR_INPUT

```
VAR_INPUT
    in : T_FIX16;
    n : BYTE(0..15);
END_VAR
```

in: Festkommazahl deren Auflösung geändert werden soll (Typ: T_FIX16 [► 324]).

n: Die neue Anzahl der Nachkommastellen.

Beispiel:

```
PROGRAM FIXALIGN
VAR
    q8, q4 : T_FIX16;
    result : LREAL;
END_VAR
```

```

q8 := LREAL_TO_FIX16( 0.6, 8 );
result := FIX16_TO_LREAL( q8 ); (* The result is: 0.6015625 *)

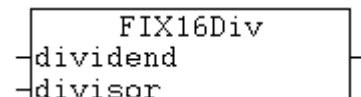
q4 := FIX16Align( q8, 4 );
result := FIX16_TO_LREAL( q4 ); (* The result is: 0.5625 *)

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7.5 FIX16Div



Diese Funktion dividiert zwei vorzeichenbehaftete 16 Bit Festkommazahlen. Die Auflösung (Anzahl der Nachkommastellen) der Zahlen muss nicht gleich sein. Die Auflösung der Zahl mit der höheren Anzahl an Nachkommastellen wird vor der Division reduziert. D.h. die Nachkommastellen der Zahl mit der höheren Auflösung werden abgeschnitten. Das Ergebnis der Division ist eine vorzeichenbehaftete 16 Bit Festkommazahl.

FUNCTION FIX16Div: [T_FIX16](#) [[▶ 324](#)]

VAR_INPUT

```

VAR_INPUT
    dividend : T_FIX16;
    divisor  : T_FIX16;
END_VAR

```

dividend: Zahl, die geteilt wird (Typ: [T_FIX16](#) [[▶ 324](#)]).

divisor: Zahl, durch die geteilt wird (Typ: [T_FIX16](#) [[▶ 324](#)]).

Beispiel:

```

PROGRAM FIXDIV
VAR
    a, b    : T_FIX16;
    result  : LREAL;
END_VAR

a := LREAL_TO_FIX16( -22.5, 8 );
b := LREAL_TO_FIX16( 10.0, 8 );

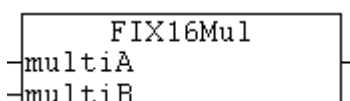
result := FIX16_TO_LREAL( FIX16Div( a, b ) ); (* The result is: -2.25 *)

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7.6 FIX16Mul



Diese Funktion multipliziert zwei vorzeichenbehaftete 16 Bit Festkommazahlen. Die Auflösung (Anzahl der Nachkommastellen) der Zahlen muss nicht gleich sein. Die Auflösung der Zahl mit der höheren Anzahl an Nachkommastellen wird vor der Multiplikation reduziert. D.h. die Nachkommastellen der Zahl mit der höheren Auflösung werden abgeschnitten. Das Ergebnis der Multiplikation ist eine vorzeichenbehaftete 16 Bit Festkommazahl.

FUNCTION FIX16Mul: T_FIX16 [▶ 324]

VAR_INPUT

```
VAR_INPUT
    multiA : T_FIX16;
    multiB : T_FIX16;
END_VAR
```

multiA: Der erste Multiplikator (Typ: T_FIX16 [▶ 324]).

multiB: Der zweite Multiplikator (Typ: T_FIX16 [▶ 324]).

Beispiel:

```
PROGRAM FIXMUL
VAR
    a, b : T_FIX16;
    result : LREAL;
END_VAR

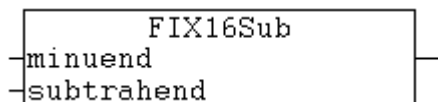
a := LREAL_TO_FIX16( 0.25, 8 );
b := LREAL_TO_FIX16( 10.0, 8 );

result := FIX16_TO_LREAL( FIX16Mul( a, b ) ); (* The result is: 2.5 *)
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7.7 FIX16Sub



Diese Funktion subtrahiert zwei vorzeichenbehaftete 16 Bit Festkommazahlen. Die Auflösung (Anzahl der Nachkommastellen) der Zahlen muss nicht gleich sein. Die Auflösung der Zahl mit der höheren Anzahl an Nachkommastellen wird vor der Subtraktion reduziert. D.h. die Nachkommastellen der Zahl mit der höheren Auflösung werden abgeschnitten. Das Ergebnis der Subtraktion ist eine vorzeichenbehaftete 16 Bit Festkommazahl.

FUNCTION FIX16Sub : T_FIX16 [▶ 324]

VAR_INPUT

```
VAR_INPUT
    minuend : T_FIX16;
    subtrahend : T_FIX16;
END_VAR
```

minuend: Zahl von der etwas abgezogen wird (Typ: T_FIX16 [▶ 324]).

subtrahend: Zahl die abgezogen wird (Typ: T_FIX16 [▶ 324]).

Beispiel:


```
PROGRAM FIXSUB
VAR
  a, b    : T_FIX16;
  result : LREAL;
END_VAR

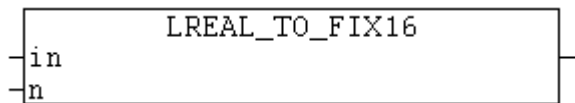
a := LREAL_TO_FIX16( 0.5, 8 );
b := LREAL_TO_FIX16( 0.75, 8 );

result := FIX16_TO_LREAL( FIX16Sub( a, b ) ); (* The result is: -0.25 *)
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.7.8 LREAL_TO_FIX16



Konvertiert eine Fließkommazahl vom Typ: LREAL in eine vorzeichenbehaftete 16 Bit Festkommazahl mit einer gewünschten Anzahl der Nachkommastellen.

FUNCTION LREAL_TO_FIX16 : T_FIX16 [► 324]

VAR_INPUT

```
VAR_INPUT
  in : LREAL;
  n  : WORD(0..15) := 15;
END_VAR
```

in: Die zu konvertierende LREAL-Zahl.

n: Anzahl der gewünschten Nachkommastellen.

Beispiel:

Im folgenden Beispiel werden einige Konstanten in Festkommazahlen konvertiert. Die Anzahl der Nachkommastellen kann bei der Konvertierung festgelegt werden. Bitte beachten Sie, dass bei der Konvertierung ähnlich wie bei den Fließkommazahlen Rundungsfehler entstehen können (in unserem Beispiel: q2 und q15).

```
PROGRAM TEST
VAR
  q2, q4, q8, q12, q15 : T_FIX16;
  r2, r4, r8, r12, r15 : LREAL;
END_VAR

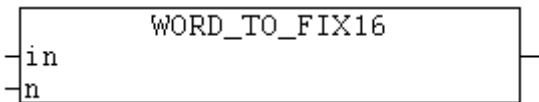
q2 := LREAL_TO_FIX16( 0.6, 2 );
q4 := LREAL_TO_FIX16( -0.25, 4 );
q8 := LREAL_TO_FIX16( -0.75, 8 );
q12 := LREAL_TO_FIX16( 2.30078125, 12 );
q15 := LREAL_TO_FIX16( 0.6, 15 );

r2 := FIX16_TO_LREAL( q2 ); (* 0.5 *)
r4 := FIX16_TO_LREAL( q4 ); (* -0.25 *)
r8 := FIX16_TO_LREAL( q8 ); (* -0.75 *)
r12 := FIX16_TO_LREAL( q12 ); (* 2.30078125 *)
r15 := FIX16_TO_LREAL( q15 ); (* 0.600006103515625 *)
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.7.9 WORD_TO_FIX16



Die Funktion konvertiert eine WORD-Variable in eine 16 Bit Festkommazahl (die WORD-Variable beinhaltet die kodierten Vor- und Nachkommastellen der Festkommazahl).

FUNCTION WORD_TO_FIX16 : T_FIX16 [▶ 324]

VAR_INPUT

```
VAR_INPUT
  in : WORD; (* 16 bit fixed point number *)
  n : WORD(0..15); (* number of fractional bits *)
END_VAR
```

Beispiel:

```
PROGRAM WORD_TO_FIX
VAR
  double : LREAL;
END_VAR
```

```
double := FIX16_TO_LREAL(WORD_TO_FIX16(2#0000110010000000, 8));
```

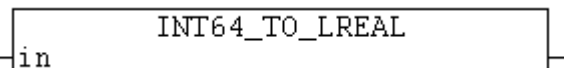
Der Wert der *double*-Variablen ist: 12.5

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8 64-Bit-Funktionen (signed)

4.8.1 INT64_TO_LREAL



Die Funktion konvertiert eine TwinCAT 2 vorzeichenbehaftete 64 Bit Zahl („legacy“-Typ: T_LARGE_INTEGER [▶ 326]) in eine Fließkommazahl vom Typ LREAL.

FUNCTION INT64_TO_LREAL: LREAL

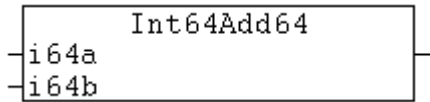
VAR_INPUT

```
VAR_INPUT
  in : T_LARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.2 Int64Add64



Die Funktion addiert zwei TwinCAT 2 vorzeichenbehaftete 64 Bit Zahlen („legacy“-Typ: T_LARGE_INTEGER [[▶ 326](#)]). Das Ergebnis ist eine vorzeichenbehaftete 64 Bit Zahl.

FUNCTION Int64Add64: T_LARGE_INTEGER

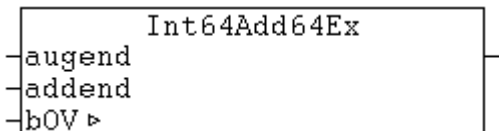
VAR_INPUT

```
VAR_INPUT
    i64a : T_LARGE_INTEGER;
    i64b : T_LARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.3 Int64Add64Ex



Die Funktion addiert zwei TwinCAT 2 vorzeichenbehaftete 64 Bit Zahlen („legacy“-Typ: T_LARGE_INTEGER [[▶ 326](#)]). Das Ergebnis ist eine vorzeichenbehaftete 64 Bit Zahl.

FUNCTION Int64Add64Ex: T_LARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    augend : T_LARGE_INTEGER;
    addend : T_LARGE_INTEGER;
    bOV    : BOOL;
END_VAR
```

VAR_IN_OUT

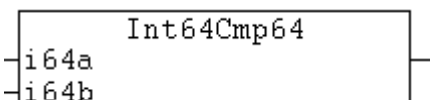
```
VAR_IN_OUT
    bOV    : BOOL; (* TRUE => arithmetic overflow, FALSE => no overflow *)
END_VAR
```

bOV: Arithmetischer Überlauf. TRUE => Überlauf, FALSE => kein Überlauf.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.4 Int64Cmp64



Die Funktion vergleicht zwei TwinCAT 2 vorzeichenbehaftete 64 Bit Zahlen („legacy“-Typ: T_LARGE_INTEGER [[► 326](#)]).

FUNCTION Int64Cmp64: DINT

VAR_INPUT

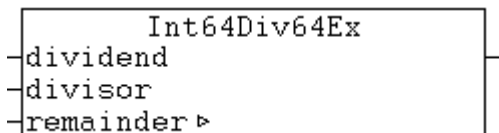
```
VAR_INPUT
  i64a : T_LARGE_INTEGER;
  i64b : T_LARGE_INTEGER;
END_VAR
```

Rückgabeparameter	Beschreibung
-1	i64a kleiner als <i>i64b</i>
0	i64a identisch mit <i>i64b</i>
1	i64a größer als <i>i64b</i>

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.5 Int64Div64Ex



Die Funktion dividiert zwei TwinCAT 2 vorzeichenbehaftete 64 Bit Zahlen („legacy“-Typ: T_LARGE_INTEGER [[► 326](#)]). Das Ergebnis ist eine vorzeichenbehaftete 64 Bit Zahl.

FUNCTION Int64Div64Ex: T_LARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
  dividend : T_LARGE_INTEGER;
  divisor : T_LARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

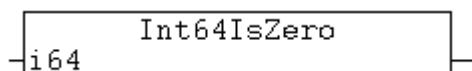
```
VAR_IN_OUT
  remainder : T_LARGE_INTEGER;
END_VAR
```

remainder: Rest.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.6 Int64isZero



Die Funktion liefert TRUE wenn die TwinCAT 2 vorzeichenbehaftete 64 Bit Zahl („legacy“-Typ: T_LARGE_INTEGER [► 326]) den Wert Null hat.

FUNCTION Int64isZero: BOOL

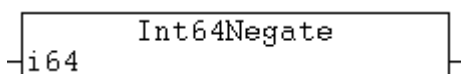
VAR_INPUT

```
VAR_INPUT
  i64 : T_LARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.8.7 Int64Negate



Die Funktion negiert eine TwinCAT 2 vorzeichenbehaftete 64 Bit Zahl („legacy“-Typ: T_LARGE_INTEGER [► 326]).

FUNCTION Int64Negate: T_LARGE_INTEGER

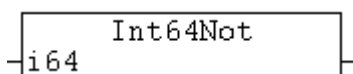
VAR_INPUT

```
VAR_INPUT
  i64 : T_LARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.8.8 Int64Not



Bitweises NOT einer TwinCAT 2 vorzeichenbehafteten 64 Bit Zahl („legacy“-Typ: T_LARGE_INTEGER [► 326]). Das Ergebnis ist eine vorzeichenbehaftete 64 Bit Zahl.

FUNCTION Int64Not: T_LARGE_INTEGER

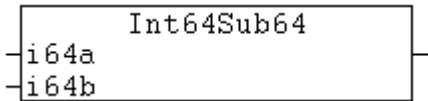
VAR_INPUT

```
VAR_INPUT
  i64 : T_LARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.8.9 Int64Sub64



Die Funktion substrahiert zwei TwinCAT 2 vorzeichenbehaftete 64 Bit Zahlen („legacy“-Typ: T_LARGE_INTEGER [▶ 326]). Das Ergebnis ist eine vorzeichenbehaftete 64 Bit Zahl.

FUNCTION Int64Sub64 : T_LARGE_INTEGER

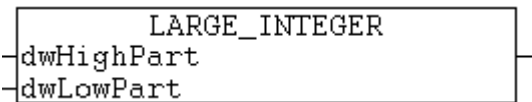
VAR_INPUT

```
VAR_INPUT
  i64a : T_LARGE_INTEGER;
  i64b : T_LARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.8.10 LARGE_INTEGER



Die Funktion initialisiert eine TwinCAT 2 vorzeichenbehaftete 64 Bit Zahl („legacy“-Typ: T_LARGE_INTEGER [▶ 326]).

FUNCTION LARGE_INTEGER: T_LARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
  dwHighPart : DWORD;
  dwLowPart : DWORD;
END_VAR
```

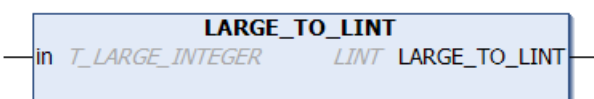
dwHighPart: Oberen 32 bit.

dwLowPart: Unteren 32 bit.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.8.11 LARGE_TO_LINT



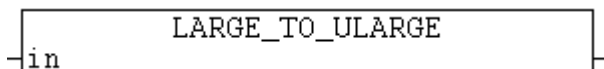
Die Funktion konvertiert eine TwinCAT 2 vorzeichenbehaftete 64 Bit Zahl („legacy“-Typ: T_LARGE_INTEGER [▶ 326]) in eine TwinCAT 3 vorzeichenbehaftete 64 Bit Zahl („native“-Typ).

FUNCTION LARGE_TO_LINT : LINT**VAR_INPUT**

```
VAR_INPUT
  in : T_LARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.12 LARGE_TO_ULARGE

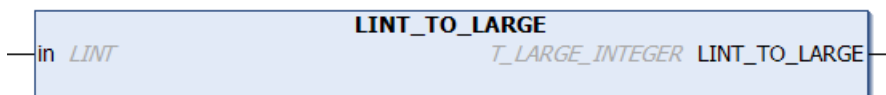
Die Funktion konvertiert eine TwinCAT 2 vorzeichenbehaftete 64 Bit Zahl („legacy“-Typ: T_LARGE_INTEGER [► 326]) in eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [► 327]).

FUNCTION LARGE_TO_ULARGE: T_ULARGE_INTEGER**VAR_INPUT**

```
VAR_INPUT
  in : T_LARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.13 LINT_TO_LARGE

Die Funktion konvertiert eine TwinCAT 3 vorzeichenbehaftete 64 Bit Zahl („native“-Typ) in eine TwinCAT 2 vorzeichenbehaftete 64 Bit Zahl („legacy“-Typ: T_LARGE_INTEGER [► 326]).

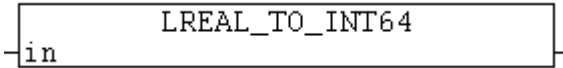
FUNCTION LINT_TO_LARGE : T_LARGE_INTEGER**VAR_INPUT**

```
VAR_INPUT
  in : LINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.14 LREAL_TO_INT64



Die Funktion konvertiert eine LREAL Zahl in eine TwinCAT 2 vorzeichenbehaftete 64 Bit Zahl („legacy“-Typ: T_LARGE_INTEGER [[▶ 326](#)]).

FUNCTION LREAL_TO_INT64: T_LARGE_INTEGER

VAR_INPUT

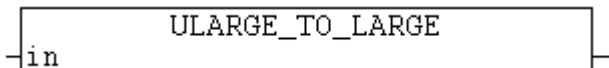
```

VAR_INPUT
  in : LREAL;
END_VAR
  
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.8.15 ULARGE_TO_LARGE



Die Funktion konvertiert eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [[▶ 327](#)]) in eine TwinCAT 2 vorzeichenbehaftete 64 Bit Zahl („legacy“-Typ: T_LARGE_INTEGER [[▶ 326](#)]).

FUNCTION ULARGE_TO_LARGE: T_LARGE_INTEGER

VAR_INPUT

```

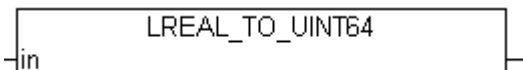
VAR_INPUT
  in : T_ULARGE_INTEGER;
END_VAR
  
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9 64-Bit-Funktionen (unsigned)

4.9.1 LREAL_TO_UINT64



Die Funktion konvertiert eine LREAL Zahl in eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [[▶ 327](#)]).

FUNCTION LREAL_TO_UINT64: T_ULARGE_INTEGER

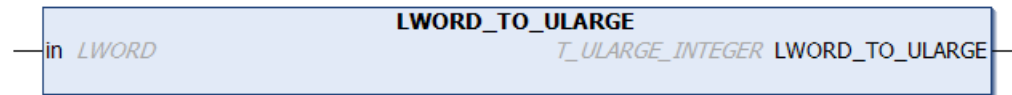
VAR_INPUT

```

VAR_INPUT
  in : LREAL;
END_VAR
  
```


Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.2 LWORD_TO_ULARGE

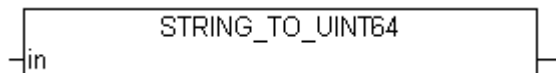
Die Funktion konvertiert eine TwinCAT 3 vorzeichenlose 64 Bit Zahl („native“-Typ) in eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: [T_ULARGE_INTEGER](#) | [327](#)).

FUNCTION LWORD_TO_ULARGE : T_ULARGE_INTEGER**VAR_INPUT**

```
VAR_INPUT
  in : LWORD;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.3 STRING_TO_UINT64

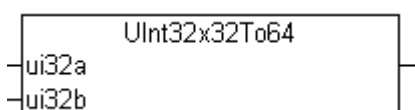
Die Funktion konvertiert einen String in eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: [T_ULARGE_INTEGER](#) | [327](#)).

FUNCTION STRING_TO_UINT64: T_ULARGE_INTEGER**VAR_INPUT**

```
VAR_INPUT
  in : STRING(21);
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.4 UInt32x32To64

Die Funktion multipliziert zwei vorzeichenlose 32 Bit Zahlen. Das Ergebnis ist eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: [T_ULARGE_INTEGER](#) | [327](#)).

FUNCTION UInt32x32To64: T_ULARGE_INTEGER

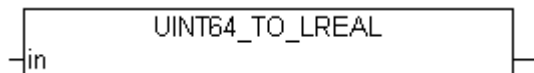
VAR_INPUT

```
VAR_INPUT
    ui32a : DWORD;
    ui32b : DWORD;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.5 UINT64_TO_LREAL



Die Funktion konvertiert eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [[▶ 327](#)]) in eine Fließkommazahl vom Typ LREAL.

FUNCTION UINT64_TO_LREAL: LREAL

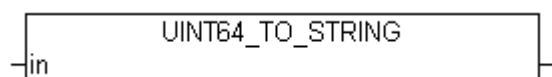
VAR_INPUT

```
VAR_INPUT
    in : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.6 UINT64_TO_STRING



Die Funktion konvertiert eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [[▶ 327](#)]) in einen String.

FUNCTION UINT64_TO_STRING: STRING(21)

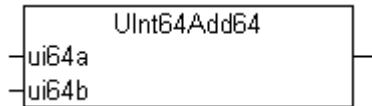
VAR_INPUT

```
VAR_INPUT
    in : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.7 UInt64Add64



Die Funktion addiert zwei TwinCAT 2 vorzeichenlose 64 Bit Zahlen („legacy“-Typ: `T_ULARGE_INTEGER` [► 327]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Add64: T_ULARGE_INTEGER

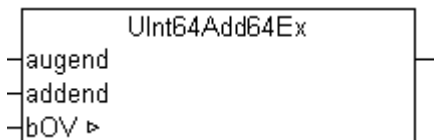
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.8 UInt64Add64Ex



Die Funktion addiert zwei TwinCAT 2 vorzeichenlose 64 Bit Zahlen („legacy“-Typ: `T_ULARGE_INTEGER` [► 327]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Add64Ex: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    augend : T_ULARGE_INTEGER;
    addend : T_ULARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

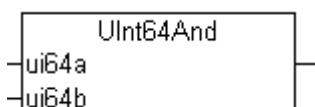
```
VAR_IN_OUT
    boV : BOOL; (* TRUE => arithmetic overflow, FALSE => no overflow *)
END_VAR
```

boV: Arithmentischer Überlauf. TRUE => Überlauf, FALSE => kein Überlauf.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.9 UInt64And



Bitweises AND von zwei TwinCAT 2 vorzeichenlosen 64 Bit Zahlen („legacy“-Typ: T_ULARGE_INTEGER [► 327]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64And: T_ULARGE_INTEGER

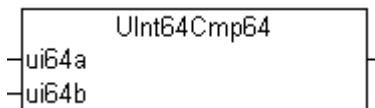
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.10 UInt64Cmp64



Die Funktion vergleicht zwei TwinCAT 2 vorzeichenlose 64 Bit Zahlen („legacy“-Typ: T_ULARGE_INTEGER [► 327]).

FUNCTION UInt64Cmp64: DINT

VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Rückgabeparameter	Beschreibung
-1	ui64a kleiner als <i>ui64b</i>
0	ui64a identisch mit <i>ui64b</i>
1	ui64a größer als <i>ui64b</i>

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.11 UInt64Div16Ex



Die Funktion dividiert eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [► 327]) durch eine 16 Bit vorzeichenlose Zahl. Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Div16Ex : T_ULARGE_INTEGER**VAR_INPUT**

```

VAR_INPUT
  dividend : T_ULARGE_INTEGER;
  divisor   : WORD;
END_VAR

```

VAR_IN_OUT

```

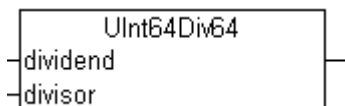
VAR_IN_OUT
  remainder : T_ULARGE_INTEGER;
END_VAR

```

remainder: Rest.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.12 UInt64Div64

Die Funktion dividiert zwei TwinCAT 2 vorzeichenlose 64 Bit Zahlen („legacy“-Typ: T_ULARGE_INTEGER [[▶ 327](#)]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Div64: T_ULARGE_INTEGER**VAR_INPUT**

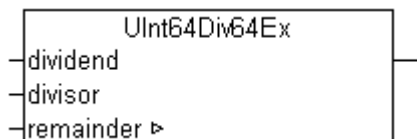
```

VAR_INPUT
  dividend : T_ULARGE_INTEGER;
  divisor   : T_ULARGE_INTEGER;
END_VAR

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.13 UInt64Div64Ex

Die Funktion dividiert zwei TwinCAT 2 vorzeichenlose 64 Bit Zahlen („legacy“-Typ: T_ULARGE_INTEGER [[▶ 327](#)]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Div64Ex : T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    dividend : T_ULARGE_INTEGER;
    divisor   : T_ULARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

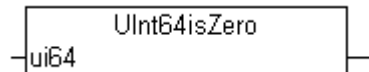
```
VAR_IN_OUT
    remainder : T_ULARGE_INTEGER;
END_VAR
```

remainder: Rest.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.14 UInt64isZero



Die Funktion liefert TRUE wenn die TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [▶ 327]) den Wert Null hat.

FUNCTION UInt64isZero: BOOL

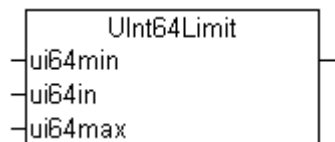
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.15 UInt64Limit



Limitierung. Das Ergebnis ist eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [▶ 327]).

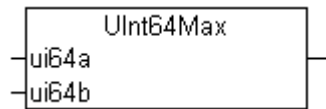
FUNCTION UInt64Limit: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    ui64min : T_ULARGE_INTEGER;
    ui64in  : T_ULARGE_INTEGER;
    ui64max : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.16 UInt64Max

Maximumsfunktion. Liefert von zwei Werten den größeren Wert („legacy“-Typ: [T_ULARGE_INTEGER](#) [► 327]).

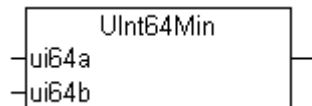
FUNCTION UInt64Max: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.17 UInt64Min

Minimumsfunktion. Liefert von zwei Werten den kleineren Wert („legacy“-Typ: [T_ULARGE_INTEGER](#) [► 327]).

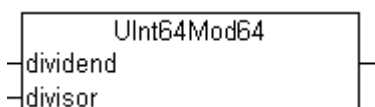
FUNCTION UInt64Min: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.18 UInt64Mod64

Modulo-Division einer TwinCAT 2 vorzeichenlosen 64 Bit Zahl durch eine andere Zahl („legacy“-Typ: [T_ULARGE_INTEGER](#) [► 327]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Mod64: T_ULARGE_INTEGER

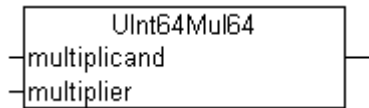
VAR_INPUT

```
VAR_INPUT
    dividend : T_ULARGE_INTEGER;
    divisor  : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.19 UInt64Mul64



Die Funktion multipliziert zwei TwinCAT 2 vorzeichenlose 64 Bit Zahlen („legacy“-Typ: T_ULARGE_INTEGER [► 327]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Mul64 : T_ULARGE_INTEGER

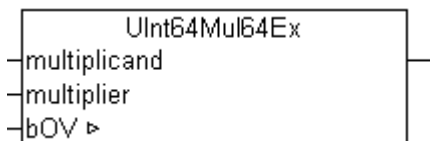
VAR_INPUT

```
VAR_INPUT
    multiplicand : T_ULARGE_INTEGER;
    multiplier   : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.20 UInt64Mul64Ex



Die Funktion multipliziert zwei TwinCAT 2 vorzeichenlose 64 Bit Zahlen („legacy“-Typ: T_ULARGE_INTEGER [► 327]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Mul64Ex: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    multiplicand : T_ULARGE_INTEGER;
    multiplier   : T_ULARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

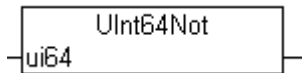
```
VAR_IN_OUT
    bOV : BOOL; (* TRUE => Arithmetic overflow, FALSE => no overflow *)
END_VAR
```


bOV: Arithmetischer Überlauf. TRUE => Überlauf, FALSE => kein Überlauf.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.21 UInt64Not



Bitweises NOT einer TwinCAT 2 vorzeichenlosen 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [► 327]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Not: T_ULARGE_INTEGER

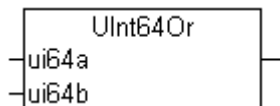
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.22 UInt64Or



Bitweises OR von zwei TwinCAT 2 vorzeichenlosen 64 Bit Zahlen („legacy“-Typ: T_ULARGE_INTEGER [► 327]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Or: T_ULARGE_INTEGER

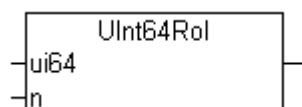
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.23 UInt64Rol



Bitweise Linksrotation einer TwinCAT 2 vorzeichenlosen 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [[▶ 327](#)]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Rol: T_ULARGE_INTEGER

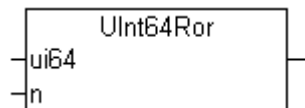
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
    n    : DWORD;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.24 UInt64Ror



Bitweise Rechtsrotation einer TwinCAT 2 vorzeichenlosen 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [[▶ 327](#)]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Ror: T_ULARGE_INTEGER

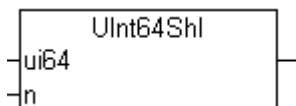
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
    n    : DWORD;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.25 UInt64Shl



Bitweises Links-Shift einer TwinCAT 2 vorzeichenlosen 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [[▶ 327](#)]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

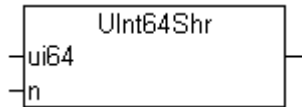
FUNCTION UInt64Shl: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
    n    : DWORD;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.26 UInt64Shr

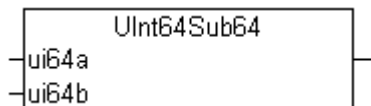
Bitweises Rechts-Shift einer TwinCAT 2 vorzeichenlosen 64 Bit Zahl („legacy“-Typ: T_ULONG_INTEGER [► 327]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Shr: T_ULONG_INTEGER**VAR_INPUT**

```
VAR_INPUT
  ui64 : T_ULONG_INTEGER;
  n    : DWORD;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.27 UInt64Sub64

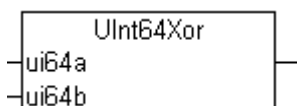
Die Funktion substrahiert zwei TwinCAT 2 vorzeichenlose 64 Bit Zahlen („legacy“-Typ: T_ULONG_INTEGER [► 327]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Sub64 : T_ULONG_INTEGER**VAR_INPUT**

```
VAR_INPUT
  ui64a : T_ULONG_INTEGER;
  ui64b : T_ULONG_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.28 UInt64Xor

Bitweises XOR von zwei TwinCAT 2 vorzeichenlosen 64 Bit Zahlen („legacy“-Typ: T_ULARGE_INTEGER [► 327]). Das Ergebnis ist eine vorzeichenlose 64 Bit Zahl.

FUNCTION UInt64Xor: T_ULARGE_INTEGER

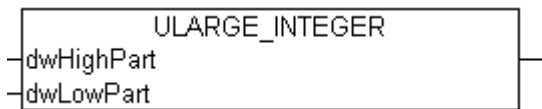
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.29 ULARGE_INTEGER



Die Funktion initialisiert eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [► 327]).

FUNCTION ULARGE_INTEGER: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    dwHighPart : DWORD;
    dwLowPart  : DWORD;
END_VAR
```

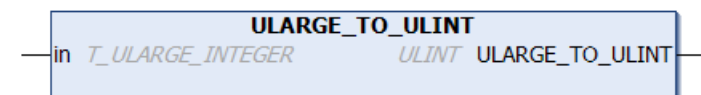
dwHighPart: Obere 32 bit.

dwLowPart: Untere 32 bit.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.30 ULARGE_TO_ULINT



Die Funktion konvertiert eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [► 327]) in eine TwinCAT 3 vorzeichenlose 64 Bit Zahl („native“-Typ).

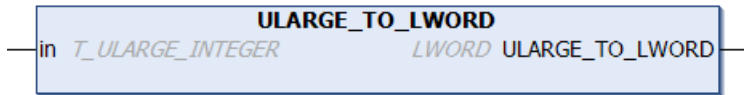
FUNCTION ULARGE_TO_ULINT : ULINT

VAR_INPUT

```
VAR_INPUT
    in : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.9.31 ULARGE_TO_LWORD

Die Funktion konvertiert eine TwinCAT 2 vorzeichenlose 64 Bit Zahl („legacy“-Typ: `T_ULARGE_INTEGER` [[▶ 327](#)]) in eine TwinCAT 3 vorzeichenlose 64 Bit Zahl („native“-Typ).

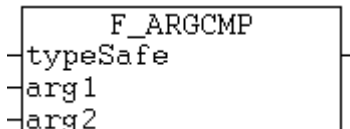
FUNCTION ULARGE_TO_LWORD : LWORD**VAR_INPUT**

```

VAR_INPUT
  in : T_ULARGE_INTEGER;
END_VAR
  
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10 T_Arg-Hilfsfunktionen**4.10.1 F_ARGCMP**

Diese Funktion vergleicht zwei Variablen vom Typ `T_Arg` und liefert das Ergebnis des Vergleichs als Rückgabeparameter zurück.

FUNCTION F_ARGCMP: DINT**VAR_INPUT**

```

VAR_INPUT
  typeSafe : BOOL;
  arg1     : T_Arg;
  arg2     : T_Arg;
END_VAR
  
```

typeSafe: Wenn TRUE => Gleiche Typen können verglichen werden (Typsicherer-Vergleich). FALSE => Unterschiedliche Typen können verglichen werden (Typunabhängiger-Vergleich).

arg1: Erste Variable die verglichen werden soll (Typ: `T_Arg` [[▶ 323](#)]).

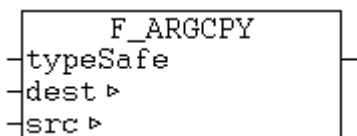
arg2: Zweite Variable die verglichen werden soll (Typ: `T_Arg` [[▶ 323](#)]).

Rückgabeparameter	Verhältnis des ersten unterschiedlichen Bytes (Typ, Länge, Wert) in der ersten und zweiten Variablen
-3	Länge von <i>arg1</i> kleiner als <i>arg2</i>
-2	Typ von <i>arg1</i> kleiner als <i>arg2</i>
-1	Wert von <i>arg1</i> kleiner als <i>arg2</i>
0	<i>arg1</i> identisch mit <i>arg2</i>
1	Wert von <i>arg1</i> größer als <i>arg2</i>
2	Typ von <i>arg1</i> größer als <i>arg2</i>
3	Länge von <i>arg1</i> größer als <i>arg2</i>
0xFF	Falsche Parameterwerte, Typ, Länge, Wert von <i>arg1</i> oder <i>arg2</i> = 0

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.2 F_ARGCPY



Diese Funktion kopiert den Wert einer Variablen vom Typ *T_Arg* in eine andere Variable und liefert die Anzahl der erfolgreich kopierten Datenbytes als Rückgabeparameter zurück.

FUNCTION F_ARGCPY: UDINT

VAR_INPUT

```
VAR_INPUT
    typeSafe : BOOL;
END_VAR
```

typeSafe: Wenn TRUE => Gleiche Typen können verglichen werden (Typsicherer-Vergleich). FALSE => Unterschiedliche Typen können verglichen werden (Typunabhängiger-Vergleich).

VAR_IN_OUT

```
VAR_IN_OUT
    dest : T_Arg;
    src : T_Arg;
END_VAR
```

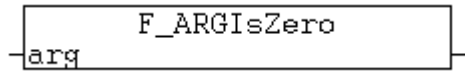
dest: Zielvariable in die hineinkopiert werden soll (Typ: *T_Arg* [▸ 323]).

src: Quellvariable von der kopiert werden soll (Typ: *T_Arg* [▸ 323]).

Rückgabeparameter	Bedeutung
0	Falsche Parameterwerte. Der Typ, Länge oder Wert von <i>dest</i> oder <i>src</i> == 0
> 0	Bei Erfolg, die Anzahl der kopierten Bytes.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.3 F_ARGISZERO

Diese Funktion liefert TRUE wenn eine der T_Arg-Membervariablen den Wert Null hat oder nicht initialisiert wurde.

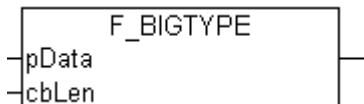
FUNCTION F_ARGISZERO: BOOL**VAR_INPUT**

```
VAR_INPUT
    arg : T_Arg;
END_VAR
```

arg: Variable die geprüft werden soll (Typ : [T_Arg](#) [► 323]).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.4 F_BIGTYPE

Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer Struct- oder Array-Variablen zurückliefert.

FUNCTION F_BIGTYPE : T_Arg**VAR_INPUT**

```
VAR_INPUT
    pData : POINTER TO BYTE;
    cbLen : DWORD;
END_VAR
```

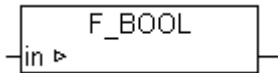
pData: Adresspointer (kann mit dem ADR-Operator ermittelt werden).

cbLen: Anzahl der Bytes, die im Speicher belegt werden (kann mit SIZEOF-Operator ermittelt werden).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.5 F_BOOL



Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer BOOL-Variablen zurückliefert.

FUNCTION F_BOOL : T_Arg

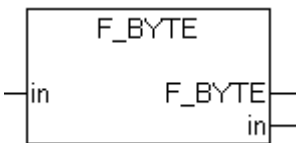
VAR_IN_OUT

```
VAR_IN_OUT
  in : BOOL;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.6 F_BYTE



Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer BYTE-Variablen zurückliefert.

FUNCTION F_BYTE : T_Arg

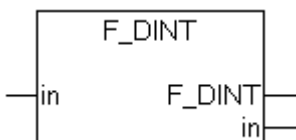
VAR_IN_OUT

```
VAR_IN_OUT
  in : BYTE;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.7 F_DINT



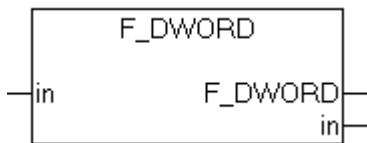
Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer DINT-Variablen zurückliefert.

FUNCTION F_DINT : T_Arg**VAR_IN_OUT**

```
VAR_IN_OUT
  in : DINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.8 F_DWORD

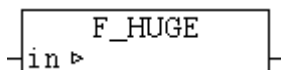
Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer DWORD-Variablen zurückliefert.

FUNCTION F_DWORD : T_Arg**VAR_IN_OUT**

```
VAR_IN_OUT
  in : DWORD;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.9 F_HUGE

Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer [T_HUGE_INTEGER](#) [► 326]-Variablen zurückliefert (signed 128 bit integer, TwinCAT 2 „legacy“-Typ).

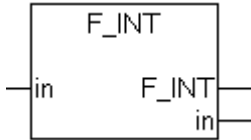
FUNCTION F_HUGE : T_Arg**VAR_IN_OUT**

```
VAR_IN_OUT
  in : T_HUGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.10 F_INT



Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [[▶ 323](#)]) zu einer INT-Variablen zurückliefert.

FUNCTION F_INT : T_Arg

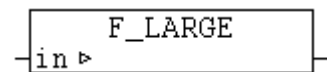
VAR_IN_OUT

```
VAR_IN_OUT
  in : INT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.11 F_LARGE



Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [[▶ 323](#)]) zu einer [T_LARGE_INTEGER](#) [[▶ 326](#)]-Variablen zurückliefert (signed 64 bit integer, TwinCAT 2 „legacy“-Typ).

FUNCTION F_LARGE : T_Arg

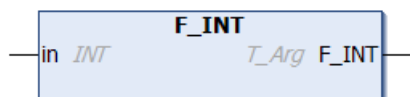
VAR_IN_OUT

```
VAR_IN_OUT
  in : T_LARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.12 F_LINT



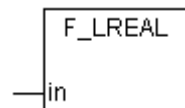
Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [[▶ 323](#)]) zu einer LINT-Variablen zurückliefert.

FUNCTION F_LINT : T_Arg**VAR_IN_OUT**

```
VAR_IN_OUT
  in : LINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.13 F_LREAL

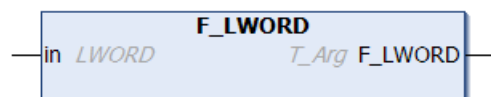
Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer LREAL-Variablen zurückliefert.

FUNCTION F_LREAL : T_Arg**VAR_IN_OUT**

```
VAR_IN_OUT
  in : LREAL;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.14 F_LWORD

Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer LWORD-Variablen zurückliefert.

FUNCTION F_LWORD : T_Arg**VAR_IN_OUT**

```
VAR_IN_OUT
  in : LWORD;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.15 F_REAL



Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer REAL-Variablen zurückliefert.

FUNCTION F_REAL : T_Arg

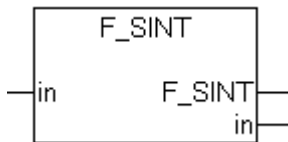
VAR_IN_OUT

```
VAR_IN_OUT
    in : REAL;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.16 F_SINT



Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer SINT-Variablen zurückliefert.

FUNCTION F_SINT : T_Arg

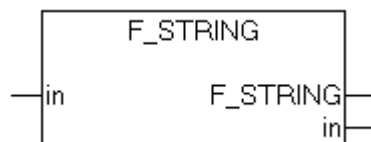
VAR_IN_OUT

```
VAR_IN_OUT
    in : SINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.17 F_STRING



Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer T_MaxString-Variablen zurückliefert.

FUNCTION F_STRING : T_Arg**VAR_IN_OUT**

```
VAR_IN_OUT
  in : T_MaxString;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.18 F_STRINGEx

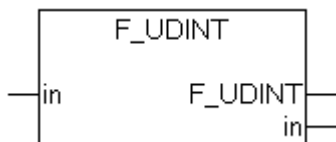
Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [▶ 323]) zu einer STRING-Variablen zurückliefert. Im Gegensatz zur Funktion [F_STRING](#) [▶ 237] ist die Länge der übergebenen STRING-Variablen beliebig.

FUNCTION F_STRINGEx : T_Arg**VAR_IN_OUT**

```
VAR_IN_OUT CONSTANT
  in : STRING;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4022	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= v3.3.34.0

4.10.19 F_UDINT

Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [▶ 323]) zu einer UDINT-Variablen zurückliefert.

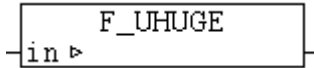
FUNCTION F_UDINT : T_Arg**VAR_IN_OUT**

```
VAR_IN_OUT
  in : UDINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.20 F_UHUGE



Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [[▶ 323](#)]) zu einer [T_UHUGE INTEGER](#) [[▶ 326](#)]-Variablen zurückliefert (unsigned 128 bit integer, TwinCAT 2 „legacy“-Typ).

FUNCTION F_UHUGE : T_Arg

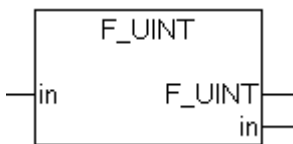
VAR_IN_OUT

```
VAR_IN_OUT
  in : T_UHUGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.21 F_UINT



Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [[▶ 323](#)]) zu einer UINT-Variablen zurückliefert.

FUNCTION F_UINT : T_Arg

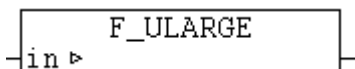
VAR_IN_OUT

```
VAR_IN_OUT
  in : UINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.22 F_ULARGE



Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [[▶ 323](#)]) zu einer [T_ULARGE INTEGER](#) [[▶ 327](#)]-Variablen zurückliefert (unsigned 64 bit number, TwinCAT 2. „legacy“-Typ).

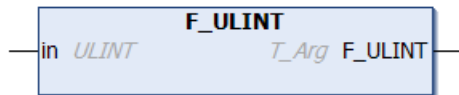
FUNCTION F_ULARGE : T_Arg

VAR_IN_OUT

```
VAR_IN_OUT
  in : T_ULARGE_INTEGER;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.23 F_ULINT

Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer ULINT-Variablen zurückliefert.

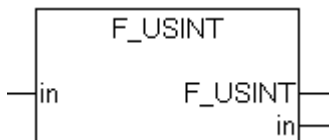
FUNCTION F_ULINT : T_Arg

VAR_IN_OUT

```
VAR_IN_OUT
  in : ULINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.24 F_USINT

Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer USINT-Variablen zurückliefert.

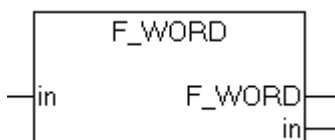
FUNCTION F_USINT : T_Arg

VAR_IN_OUT

```
VAR_IN_OUT
  in : USINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.25 F_WORD

Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer WORD-Variablen zurückliefert.

FUNCTION F_WORD : T_Arg

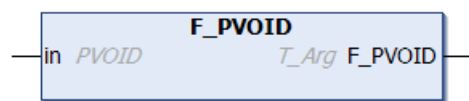
VAR_IN_OUT

```
VAR_IN_OUT
  in : WORD;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.26 F_PVOID



Eine Hilfsfunktion, die in einer Struktur Informationen (Typ: [T_Arg](#) [► 323]) zu einer PVOID-Variablen zurückliefert.

FUNCTION F_PVOID : T_Arg

VAR_IN_OUT

```
VAR_IN_OUT
  in : PVOID;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.10.27 IsFinite



Die Funktion IsFinite() liefert TRUE zurück, wenn deren Argument einen endlichen Wert besitzt ($INF < x < +INF$). D.h. die Funktion liefert FALSE zurück, wenn der Argument unendlich oder NaN ist (NaN = Not a number). IsFinite() überprüft, ob die Formatierung einer LREAL oder REAL-Variablen der IEEE entspricht.

INF-Zahlen können auf einem Laufzeitsystem dann vorkommen, wenn das Ergebnis einer mathematischen Operation den darstellbaren Bereich überschreitet oder unterschreitet. Z.B.:

```
PROGRAM MAIN
VAR
  fSingle : REAL := 12.34;
END_VAR
```

```
(*Cyclic called program code*)
fSingle := fSingle*2;
```

NaN-Zahlen können im Laufzeitsystem dann vorkommen, wenn deren eigentliche Formatierung (Speicherinhalte) durch unerlaubten Zugriff (z.B. durch Benutzung der MEMCPY, MEMSET Funktionen) überschrieben wurden. Z.B.:


```
PROGRAM MAIN
VAR
  fSingle : REAL := 12.34;
END_VAR

(*Cyclic called program code*)
MEMSET( ADR( fSingle ), 16#FF, SIZEOF( fSingle ) ); (* Invalid initialization of REAL variable *)
```

Beim Aufruf einer Konvertierungsfunktion mit einer NaN oder INF-Zahl als Parameter wird auf einem PC-System (x86,x64) eine FPU-Exception ausgelöst. Diese Exception führt anschließend zum Stopp der SPS. Mit der Funktion IsFinite() kann der Wert der Variablen überprüft, die FPU-Exception vermieden und die Programmausführung fortgesetzt werden.

FUNCTION IsFinite : BOOL

VAR_INPUT

```
VAR_INPUT
  x : T_Arg;
END_VAR
```

x: Eine Hilfsstruktur mit Informationen zu der zu überprüfenden REAL oder LREAL-Variablen (Typ: T_Arg [► 323]). Die Strukturparameter müssen beim Aufruf von IsFinite() mit Hilfsfunktionen F_REAL [► 237] oder F_LREAL [► 236] erzeugt und als Parameter übergeben werden.

Beispiel 1:

Im folgenden Beispiel wird die Formatierung einer REAL- und einer LREAL-Variablen überprüft und eine FPU-Exception vermieden.

```
PROGRAM MAIN
VAR
  fSingle      : REAL := 12.34;
  fDouble      : LREAL := 56.78;
  singleAsString : STRING;
  doubleAsString : STRING;
END_VAR

fSingle := fSingle*2;
IF IsFinite( F_REAL( fSingle ) ) THEN
  singleAsString := REAL_TO_STRING( fSingle );
ELSE
  (* report error !*)
  fSingle := 12.34;
END_IF

fDouble := fDouble*2;
IF IsFinite( F_LREAL( fDouble ) ) THEN
  doubleAsString := LREAL_TO_STRING( fDouble );
ELSE
  (* report error !*)
  fDouble := 56.78;
END_IF
```

Beispiel 2:

Im folgenden Fall kann eine FPU-Exception durch Überprüfung mit IsFinite() nicht vermieden werden:

```
PROGRAM MAIN
VAR
  bigFloat  : LREAL := 3.0E100;
  smallDigit : INT;
END_VAR

IF IsFinite( F_LREAL( bigFloat ) ) THEN
  smallDigit := LREAL_TO_INT( bigFloat );
END_IF
```

Die bigFloat-Variable besitzt zwar richtige Formatierung, der Variablenwert ist aber zu groß um diesen in einen INT-Typ konvertieren zu können. Auf einem PC-System (x86,x64) wird eine Exception ausgelöst und das Laufzeitsystem gestoppt.

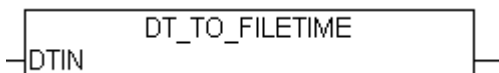
Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.11 [veraltete Funktionen]

4.11.1 Zeitfunktionen

4.11.1.1 DT_TO_FILETIME



Mit der Funktion "DT_TO_FILETIME" kann eine im DATE_AND_TIME-Format (DT) SPS-Variable in das FILETIME-Format (64Bit) konvertiert werden.

FUNCTION DT_TO_FILETIME : T_FILETIME [▶ 323]

VAR_INPUT

```
VAR_INPUT
    DTIN : DT;
END_VAR
```

DTIN: Das zu konvertierende Datum und Uhrzeit in DATE_AND_TIME-Format.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.11.1.2 F_TranslateFileTimeBias



Diese Funktion rechnet die Eingangszeit mit Hilfe der angegebenen Bias-Zeitverschiebung in die Zeit einer anderen Zeitzone um. Mit dieser Funktion kann z.B. die Lokalzeit in UTC-Zeit (Universal Time Coordinates) und umgekehrt umgerechnet werden.

FUNCTION F_TranslateFileTimeBias: T_FILETIME [▶ 323]

VAR_INPUT

```
VAR_INPUT
    in : T_FILETIME;
    bias : DINT;
    toUTC : BOOL;
END_VAR
```

in: Eingangszeit die umgerechnet werden soll (Typ: T_FILETIME [▶ 323]).

bias: Differenz zwischen der UTC-Zeit und der Lokalzeit in Minuten (positive oder negative Werte sind zulässig).

toUTC: Über diesen Parameter kann die Richtung angegeben werden in die die Eingangszeit konvertiert werden soll.

toUTC	Direction	Internal formula
FALSE	UTC -> Lokalzeit	Lokalzeit := UTC - Bias
TRUE	Lokalzeit -> UTC	UTC := Lokalzeit + Bias

Beispiel:

Die *in*-Variable enthält die zu konvertierende Zeit. Die *bToUTC*-Variable bestimmt die Konvertierungsrichtung. Bei *bToUTC* = TRUE wird die Lokalzeit in UTC-Zeit konvertiert und bei *bToUTC* = FALSE die UTC-Zeit in Lokalzeit. Die *WEST_EUROPE_TZI*-Konstante enthält die Zeitzoneinformationen für Westeuropa. Der benötigte Bias-Wert wird aus der Zeitzoneinformation in der Konstanten und der aktuellen bDST-Einstellung (Daylight Saving Time) errechnet. Die aktuelle Zeitzoneinformation eines TwinCAT-Systems kann alternativ mit dem Funktionsbaustein: [FB_GetTimeZoneInformation](#) [► 78] ermittelt werden.

Wichtiger Hinweis: Der Datentyp DT als Eingangszeit wurde nur wegen der Möglichkeit der optischen Kontrolle im Online-Mode gewählt. Konvertierungen in einem anderen Zeitformat sind nur bedingt zu empfehlen da die Konvertierungsfunktionen sehr rechenintensiv sein können.

```
PROGRAM MAIN
VAR
    bDST      : BOOL := TRUE; (* TRUE => Daylight saving time, FALSE => Standard time *)
    bToUTC    : BOOL := FALSE;
(* TRUE => Convert local time to UTC time, FALSE => Convert UTC time to local time *)
    in       : DT := DT#2011-08-29-15:15:31;
    out      : DT;
    bias     : DINT;
END_VAR

IF bDST THEN
    bias := WEST_EUROPE_TZI.bias + WEST_EUROPE_TZI.daylightBias;
ELSE
    bias := WEST_EUROPE_TZI.bias + WEST_EUROPE_TZI.standardBias;
END_IF

out := FILETIME_TO_DT( F_TranslateFileTimeBias( DT_TO_FILETIME( in ), bias, bToUTC ) );
```

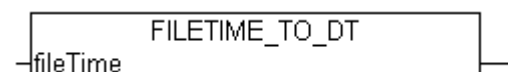
Weitere Zeit-, Zeitzone-Funktionen und -Funktionsbausteine:

- FB_TzSpecificLocalTimeToSystemTime
- FB_TzSpecificLocalTimeToFileTime
- FB_SystemTimeToTzSpecificLocalTime
- FB_FileTimeTimeToTzSpecificLocalTime
- FB_GetTimeZoneInformation
- FB_SetTimeZoneInformation
- NT_SetLocalTime
- NT_GetTime
- NT_SetTimeToRTCTime
- FB_LocalSystemTime

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11.1.3 FILETIME_TO_DT



Die Funktion "FILETIME_TO_DT" konvertiert die Zeit im FILETIME-Format in das DATE_AND_TIME-Format (DT). Das DT-Format hat einen kleineren Wertebereich als das FILETIME-Format und nur eine Sekundengenauigkeit. Aus diesem Grund wird der zu konvertierende FILETIME-Wert limitiert. Das zulässige Minimum entspricht dem Wert DT#1970-01-01-00:00:00 und das Maximum dem Wert DT#2106-02-06-06:28:15. Millisekunden werden bei der Konvertierung nicht berücksichtigt und auf den DATE_AND_TIME-Rückgabewert entsprechend abgerundet.

FUNCTION FILETIME_TO_DT : DT

VAR_INPUT

```
VAR_INPUT
    fileTime : T_FILETIME;
END_VAR
```

fileTime: Die zu konvertierende Zeit im FILETIME-Format (Typ: [T_FILETIME \[▶ 323\]](#)).

Beispiel:

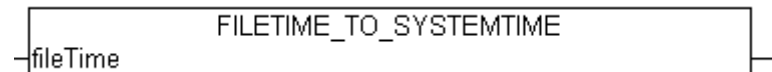
```
PROGRAM MAIN
VAR
    fbSystemTime : GETSYSTEMTIME;
    timeAsFileTime : T_FILETIME;
    timeAsDT : DT;
END_VAR

fbSystemTime( timeLoDW=>timeAsFileTime.dwLowDateTime, timeHiDW=>timeAsFileTime.dwHighDateTime );
timeAsDT := FILETIME_TO_DT( timeAsFileTime );
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.11.1.4 FILETIME_TO_SYSTEMTIME



Die Funktion "FILETIME_TO_SYSTEMTIME" konvertiert die Zeit im FILETIME-Format in das "lesbare" SYSTEMTIME-Format. Die Konvertierung schlägt fehl, wenn das höchstwertige Bit der 64 Bit fileTime-Variablen gesetzt ist. Die TIMESTRUCT-Membervariablen haben in diesem Fall den Wert Null.

FUNCTION FILETIME_TO_SYSTEMTIME: TIMESTRUCT [▶ 327]

VAR_INPUT

```
VAR_INPUT
    fileTime : T_FILETIME;
END_VAR
```

fileTime: Die zu konvertierende Zeit im FILETIME-Format (Typ: [T_FILETIME \[▶ 323\]](#)).

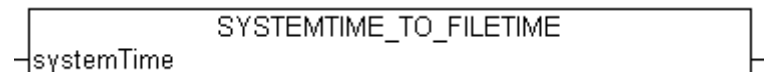
Beispiel:

```
PROGRAM MAIN
VAR
    fbSystemTime : GETSYSTEMTIME;
    timeAsFileTime : T_FILETIME;
    timeAsSystemTime : TIMESTRUCT;
END_VAR

fbSystemTime( timeLoDW=>timeAsFileTime.dwLowDateTime, timeHiDW=>timeAsFileTime.dwHighDateTime );
timeAsSystemTime := FILETIME_TO_SYSTEMTIME( timeAsFileTime );
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11.1.5 SYSTEMTIME_TO_FILETIME

Mit der Funktion kann die Windows Systemzeit-Struktur in das Filetime-Format konvertiert werden. Der Wochentag `wDayOfWeek` der `SystemTime`-Variablen wird ignoriert. Das Systemzeit-Jahr muss größer als 1601 und kleiner als 30827 sein.

FUNCTION SYSTEMTIME_TO_FILETIME : T FILETIME [► 323]**VAR_INPUT**

```

VAR_INPUT
    systemTime : TIMESTRUCT;
END_VAR

```

systemTime: Struktur mit der zu konvertierenden Windows-Systemzeit (Typ: [TIMESTRUCT \[► 327\]](#)).

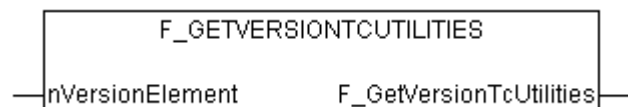
Rückgabeparameter	Beschreibung
0	Fehler, falscher SystemTime Parameterwert.
> 0	Kein Fehler. File time.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11.2 F_GetVersionTcUtilities

Diese Funktion ist veraltet und sollte nicht verwendet werden. Verwenden Sie bitte die globale Konstante: [stLibVersion_Tc2_Utilities \[► 329\]](#) um Versionsinformationen der SPS-Bibliothek auszulesen.



Mit dieser Funktion können Versionsinformationen der SPS-Bibliothek ausgelesen werden.

FUNCTION F_GetVersionTcUtilities : UINT**VAR_INPUT**

```

VAR_INPUT
    nVersionElement : INT;
END_VAR

```

nVersionElement : Versionselement, das gelesen werden soll. Mögliche Parameter:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11.3 FLOATIsFinite

Veraltete Funktion
i

Verwenden Sie stattdessen die Funktion [LrealsFinite \[► 194\]](#)().

FUNCTION FLOATIsFinite : BOOL

VAR_INPUT

```
VAR_INPUT
  x      : LREAL;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11.4 FLOATIsNaN

Veraltete Funktion
i

Verwenden Sie stattdessen die Funktion [LrealsNaN \[► 194\]](#)().

FUNCTION FLOATIsNaN : BOOL

VAR_INPUT

```
VAR_INPUT
  x      : LREAL;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.12 ARG_TO_CSVFIELD



Die Funktion konvertiert den Wert einer SPS-Variablen in ein Datenfeld im CSV-Format. Einfache Einführungszeichen in den Quelldaten werden durch doppelte Einführungszeichen ersetzt. Beim gesetzten bQM-Parameter (QM = quotation marks) werden auch die äußeren (das CSV-Datenfeld einschließende) Einführungszeichen hinzugefügt. Beim Erfolg liefert die Funktion die Länge der konvertierten Daten als

Ergebnis zurück. Die Funktion liefert Null wenn bei der Konvertierung Fehler aufgetreten sind oder bei fehlenden Daten. Das Ergebnis wird in den bereitgestellten Bytepuffer geschrieben. Die Applikation muss dafür sorgen, dass die Puffergröße so dimensioniert ist, damit das Ergebnis dort hineinpasst.

Die Funktion wird normalerweise zusammen mit dem Funktionsbaustein [FB_CSVMemBufferWriter \[► 50\]](#) verwendet um Datensätze im SPS-Speicher im CSV-Format zu erzeugen. Im nächsten Schritt kann der Speicherinhalt dann in die Datei geschrieben werden. Im Gegensatz zu der [STRING TO CSVFIELD \[► 290\]](#)-Funktion lassen sich mit dieser Funktion auch SPS-Variablen mit Binärdaten in CSV-Datenfelder konvertieren.

FUNCTION ARG_TO_CSVFIELD : UDINT

VAR_INPUT

```
VAR_INPUT
  in      : T_Arg;
  bQM    : BOOL;
  pOutput : POINTER TO BYTE;
  cbOutput : UDINT;
END_VAR
```

in: SPS-Quellvariable deren Wert in ein Datenfeld im CSV-Format konvertiert werden soll (Typ: [T_Arg \[► 323\]](#)).

bQM: Bei TRUE an diesem Eingang werden die konvertierten Felddaten in Einführungszeichen eingeschlossen.

pOutput: Anfangsadresse (Pointer) auf den Ausgangspuffer. Die Pufferadresse kann mit dem ADR-Operator ermittelt werden. In diesen Puffer werden die Ergebnisdaten hineingeschrieben.

cbOutput: Die maximal verfügbare Größe des Ausgangspuffers in Byte. Die Länge des Ausgangspuffers kann mit dem SIZEOF-Operator ermittelt werden.

Beispiel:

Im folgenden Beispiel wird gezeigt wie SPS-Variablen unterschiedlichen Typs in das CSV-Format und umgekehrt konvertiert werden können. Bei der ARG_TO_CSVFIELD-Konvertierung wird das Ergebnis in den Bytepuffer hineinkopiert (field1..field6). Bei der CSVFIELD_TO_ARG [\[► 253\]](#)-Konvertierung liegen die Quelldaten im Bytepuffer (field1..field6) und das Ergebnis wird in die TwinCAT SPS-Variable kopiert.

```
PROGRAM P_ArgToConvExample
VAR
  (* PLC data to be converted to or from CSV format *)
  bOperating : BOOL := TRUE;
  fAxPos     : LREAL := 12.2;
  nCounter   : UDINT := 7;
  sName      : T_MaxString := 'Module: "XAF", $04$05, 20';
  binData    : ARRAY[0..9] OF BYTE := [0, 1, 2, 3, 4, 5, 6, 7, 8, 9];
  sShort     : STRING(10) := 'XAF';

  (* conversion buffer *)
  field1 : ARRAY[0..50 ] OF BYTE;
  field2 : ARRAY[0..50 ] OF BYTE;
  field3 : ARRAY[0..50 ] OF BYTE;
  field4 : ARRAY[0..50 ] OF BYTE;
  field5 : ARRAY[0..50 ] OF BYTE;
  field6 : ARRAY[0..50 ] OF BYTE;

  cbField1 : UDINT;
  cbField2 : UDINT;
  cbField3 : UDINT;
  cbField4 : UDINT;
  cbField5 : UDINT;
  cbField6 : UDINT;

  cbVar1 : UDINT;
  cbVar2 : UDINT;
  cbVar3 : UDINT;
  cbVar4 : UDINT;
  cbVar5 : UDINT;
  cbVar6 : UDINT;
END_VAR
```

```

cbField1 := ARG_TO_CSVFIELD( F_BOOL( bOperating ), TRUE, ADR( field1 ), SIZEOF( field1 ) );
cbField2 := ARG_TO_CSVFIELD( F_LREAL( fAxPos ), TRUE, ADR( field2 ), SIZEOF( field2 ) );
cbField3 := ARG_TO_CSVFIELD( F_UDINT( nCounter ), TRUE, ADR( field3 ), SIZEOF( field3 ) );
cbField4 := ARG_TO_CSVFIELD( F_STRING( sName ), TRUE, ADR( field4 ), SIZEOF( field4 ) );
cbField5 := ARG_TO_CSVFIELD( F_BIGTYPE( ADR( binData ), SIZEOF( binData ) ), TRUE, ADR( field5 ), SIZEOF( field5 ) );
cbField6 := ARG_TO_CSVFIELD( F_BIGTYPE( ADR( sShort ), LEN( sShort ) ), TRUE, ADR( field6 ), SIZEOF( field6 ) );

cbVar1 := CSVFIELD_TO_ARG( ADR( field1 ), cbField1, TRUE, F_BOOL( bOperating ) );
cbVar2 := CSVFIELD_TO_ARG( ADR( field2 ), cbField2, TRUE, F_LREAL( fAxPos ) );
cbVar3 := CSVFIELD_TO_ARG( ADR( field3 ), cbField3, TRUE, F_UDINT( nCounter ) );
cbVar4 := CSVFIELD_TO_ARG( ADR( field4 ), cbField4, TRUE, F_STRING( sName ) );
cbVar5 := CSVFIELD_TO_ARG( ADR( field5 ), cbField5, TRUE, F_BIGTYPE( ADR( binData ), SIZEOF( binData ) ) );
cbVar6 := CSVFIELD_TO_ARG( ADR( field6 ), cbField6, TRUE, F_BIGTYPE( ADR( sShort ), LEN( sShort ) ) );
    
```

Das Ergebnis (Bytepuffer als hexadezimaler String):

```

cbField1 = 3, field1 = '22 01 22'
cbField2 = 10, field2 = '22 66 66 66 66 66 66 28 40 22'
cbField3 = 6, field3 = '22 07 00 00 00 22'
cbField4 = 25, field4 = '22 4D 6F 64 75 6C 65 3A 20 22 22 58 41 46 22 22 2C 20 04 05 2C 20 32 30 22'
cbField5 = 12, field5 = '22 00 01 02 03 04 05 06 07 08 09 22'
cbField6 = 5, field6 = '22 58 41 46 22'

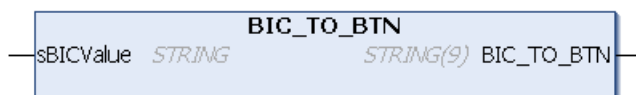
cbVar1 = 1
cbVar2 = 8
cbVar3 = 4
cbVar4 = 22
cbVar5 = 10
cbVar6 = 3
    
```

Weitere Informationen finden Sie hier: [Beispiel: Schreiben/lesen einer CSV-Datei \[► 351\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.13 BIC_TO_BTN



Die Funktion BIC_TO_BTN extrahiert vom Beckhoff Identification Code in sBICValue die Beckhoff Traceability Number (BTN) und liefert diese als Rückgabewert. Leerzeichen am Ende der BTN werden automatisch entfernt. Wird keine BTN gefunden, dann wird ein Leerstring zurückgeliefert.

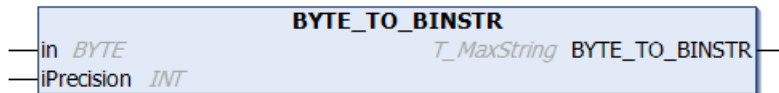
FUNCTION BIC_TO_BTN : STRING(9)**VAR_INPUT**

```
VAR_INPUT
  sBICValue   : STRING;
END_VAR
```

Name	Typ	Beschreibung
sBICValue	STRING	Dieser Eingang muss den Beckhoff Identification Code (BIC) enthalten, aus dem die Funktion BIC_TO_BTN die Beckhoff Tracability Number (BTN) extrahiert, z.B. liefert BIC_TO_BTN den Wert „0002agdw“ bei der BIC „1P193995SBTN0002agdw1KEL7411 Q1 2P112104020018“.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.14 BYTE_TO_BINSTR

Die Funktion konvertiert eine Dezimalzahl in einen Binärstring (Basis 2).

FUNCTION BYTE_TO_BINSTR : T_MaxString**VAR_INPUT**

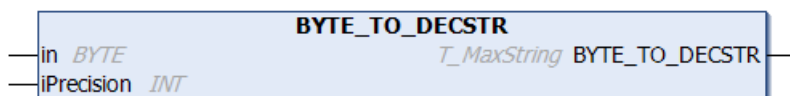
```
VAR_INPUT
  in       : BYTE;
  iPrecision : INT;
END_VAR
```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (Digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.15 BYTE_TO_DECSTR

Die Funktion konvertiert eine Dezimalzahl in einen Dezimalstring (Basis 10).

FUNCTION BYTE_TO_DECSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in      : BYTE;
    iPrecision : INT;
END_VAR
```

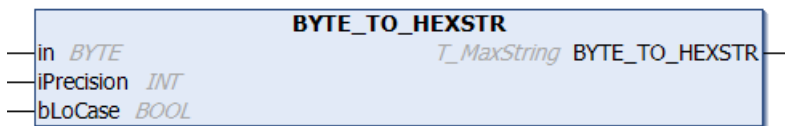
in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.16 BYTE_TO_HEXSTR



Die Funktion konvertiert eine Dezimalzahl in einen Hexadezimalstring (Basis 16).

FUNCTION BYTE_TO_HEXSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in      : BYTE;
    iPrecision : INT;
    bLoCase  : BOOL := FALSE;
END_VAR
```

in: Die zu konvertierende Dezimalzahl.

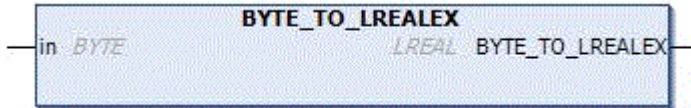
iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

bLoCase: Dieser Parameter bestimmt ob Klein- oder Großbuchstaben bei der Konvertierung benutzt werden sollen. FALSE => "ABCDEF", TRUE => "abcdef".

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.17 BYTE_TO_LREAL



In TwinCAT 2 auf der ARM-Plattform wird die Konvertierung der vorzeichenlosen Zahlen in Fließkommazahlen von Typ: LREAL nicht unterstützt. Vorzeichenlose Zahlen mit einem gesetzten höchstwertigen Bit werden möglicherweise implizit in negative Fließkommazahlen konvertiert. Die hier beschriebene Funktion erlaubt in TwinCAT 2 eine explizite Konvertierung vom Typ BYTE in eine positive Fließkommazahl vom Typ LREAL (auch wenn das höchstwertige Bit gesetzt wurde und ohne Compiler-Warnung). Diese Funktion benötigen Sie nur, um konvertierte TwinCAT 2 Projekte ohne Änderungen in TwinCAT 3 übersetzen zu können.

In TwinCAT 3 werden vorzeichenlose Zahlen von Typ BYTE immer (implizit und explizit) in eine positive Fließkommazahl Zahl konvertiert. Aus diesem Grund kann auf die Verwendung dieser Funktion verzichtet werden.

FUNCTION BYTE_TO_LREAL : LREAL

VAR_INPUT

```
VAR_INPUT
  in : BYTE;
END_VAR
```

Beispiel:

```
PROGRAM MAIN
VAR
  nByte : BYTE := 16#FF;
  fLreal : LREAL := 0.0;
END_VAR
```

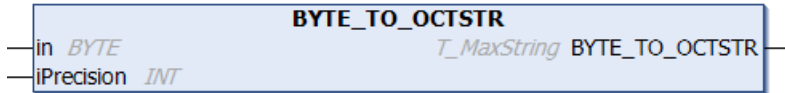
fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := nByte	+255, Warning 1105*	+255	+255
fLreal := BYTE_TO_LREAL(nByte)	+255, Warning 1105*	+255	+255
fLreal := BYTE#16#FF	+255, Warning 1105*	+255	+255
fLreal := 16#FF	+255	+255	+255
fLreal := BYTE_TO_LREAL(nByte)	+255	+255	+255
fLreal := BYTE_TO_LREAL(BYTE#16#FF)	+255	+255	+255
fLreal := BYTE_TO_LREAL(16#FF)	+255	+255	+255

*Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead.

Voraussetzungen

Entwicklungsumgebung	Zielformat	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.18 BYTE_TO_OCTSTR



Die Funktion konvertiert eine Dezimalzahl in einen Oktalstring (Basis 8).

FUNCTION BYTE_TO_OCTSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in      : BYTE;
    iPrecision : INT;
END_VAR
```

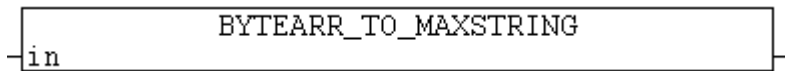
in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.19 BYTEARR_TO_MAXSTRING



Konvertiert die einzelnen ASCII-Codes eines Byte-Arrays in einen String.

FUNCTION BYTEARR_TO_MAXSTRING : T_MaxString

VAR_INPUT

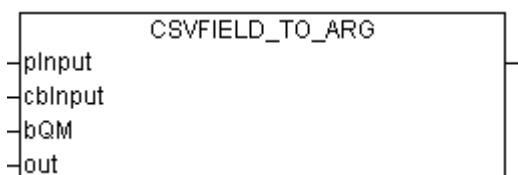
```
VAR_INPUT
    in : ARRAY[0..MAX_STRING_LENGTH] OF BYTE;
END_VAR
```

in: Byte-Array variable (MAX_STRING_LENGTH default value: 255).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.20 CSVFIELD_TO_ARG



Die Funktion konvertiert den Wert von einem Datenfeld im CSV-Format das als Bytepuffer vorliegt in eine SPS-Variable. Doppelte Einführungszeichen im Datenfeld werden durch einfache Einführungszeichen ersetzt. Beim gesetzten bQM-Parameter (QM = quotation marks) werden auch die äußeren (das Datenfeld einschließende) Einführungszeichen aus den Eingangsdaten entfernt. Beim Erfolg liefert die Funktion die Länge der konvertierten Daten zurück. Beim Fehler und bei der Länge der Eingangsdaten Null liefert die Funktion den Wert Null. Die Applikation muss dafür sorgen das die SPS-Zielvariable groß genug ist um den Wert aufnehmen zu können.

Die Funktion wird normalerweise zusammen mit dem Funktionsbaustein [FB_CSVMemBufferReader \[▶ 49\]](#) verwendet um Datensätze im SPS-Speicher die im CSV-Format vorliegen zu lesen (parsen). Vorher werden die CSV-Datensätze meistens aus einer Datei in den SPS-Speicher eingelesen. Im Gegensatz zu der [CSVFIELD_TO_STRING \[▶ 254\]](#)-Funktion lassen sich mit dieser Funktion auch CSV-Datenfelder mit Binärdaten in SPS-Variablen konvertieren.

FUNCTION CSVFIELD_TO_ARG : UDINT

VAR_INPUT

```
VAR_INPUT
  pInput   : POINTER TO BYTE;
  cbInput  : UDINT;
  bQM     : BOOL;
  out     : T_Arg;
END_VAR
```

pInput: Anfangsadresse (Pointer) auf einen Bytepuffer mit dem zu konvertierenden Datenfeld im CSV-Format. Die Adresse kann mit dem ADR-Operator ermittelt werden.

cbInput: Die Länge des zu konvertierenden Datenfeldes in Byte. Die Länge kann mit dem SIZEOF-Operator ermittelt werden.

bQM: Bei TRUE an diesem Eingang werden aus den Felddaten auch die einschließenden Einführungszeichen entfernt.

out: SPS-Zielvariable in die der Wert des Datenfeldes hineingeschrieben werden soll (Typ: [T_Arg \[▶ 323\]](#)).

Beispiel:

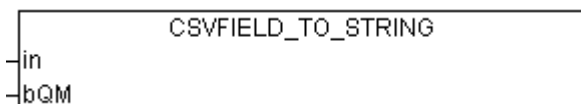
Siehe Beispiel in der Dokumentation des [ARG_TO_CSVFIELD \[▶ 247\]](#)-Funktionsbausteins.

Weitere Informationen finden Sie hier: [Beispiel: Schreiben/lesen einer CSV-Datei \[▶ 351\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.21 CSVFIELD_TO_STRING



Die Funktion konvertiert ein Datenfeld im CSV-Datenfeldformat der als Quellstring vorliegt in einen Wert im SPS-Stringformat. Doppelte Einführungszeichen im Quellstring werden durch ein einfaches Einführungszeichen ersetzt. Beim gesetzten bQM-Parameter (QM = quotation marks) werden auch die äußeren (das Datenfeld einschließenden) Einführungszeichen aus dem Quellstring entfernt. Beim Erfolg liefert die Funktion den konvertierten String als Ergebnis zurück. Die Funktion liefert einen Leerstring wenn bei der Konvertierung ein Fehler aufgetreten ist aber nur dann wenn der Quellstring auch kein Leerstring war.

Die Funktion wird normalerweise zusammen mit dem Funktionsbaustein [FB_CSVMemBufferReader \[► 49\]](#) verwendet um Datensätze im SPS-Speicher die im CSV-Format vorliegen zu lesen (zu interpretieren). Vorher werden die CSV-Datensätze meistens aus einer Datei in den SPS-Speicher eingelesen. Der Quellstring darf keine Binärdaten enthalten. Binärdaten mit dem Wert Null werden den String an der falschen Stelle terminieren und abschneiden! Um Datenfelder mit Binärdaten zu konvertieren verwenden Sie bitte die Funktion: [CSVFIELD_TO_ARG \[► 253\]](#).

FUNCTION CSFIELD_TO_STRING : T_MaxString

VAR_INPUT

```
VAR_INPUT
  in : T_MaxString;
  bQM : BOOL;
END_VAR
```

in: Quellstring mit einem Datenfeld im CSV-Format der in einen Wert im SPS-Stringformat konvertiert werden soll (Typ: T_MaxString).

bQM: Bei TRUE an diesem Eingang werden aus dem Quellstring auch die einschließenden Einführungszeichen entfernt.

bQM	Beschreibung	Quellstring	Ergebnisstring	CSV-Konform
FALSE	Ein Quellstring , der nicht mit den einschließenden Einführungszeichen versehen wird sollte möglichst nur aus Buchstaben und Zahlen bestehen. In diesem Fall darf der Quellstring keine nicht-druckbaren Steuerzeichen, Einführungszeichen, Semikolons, Kommas (US-CSV-Format) oder Binärdaten enthalten.	'Module_XA5'	'Module_XA5'	Ja
		'123456'	'123456'	Ja
		"	"	Ja
		'A''''''B'	'A''B'	Nein
		'A''B'	'A''B'	Nein
		','	','	Nein
		'\$R\$N'	'\$R\$N'	Nein
TRUE	Ein Quellstring , der mit den einschließenden Einführungszeichen versehen wird darf auch die nicht-druckbaren Steuerzeichen, Einführungszeichen, Semikolons, Kommas (US-CSV-Format) enthalten. Binärdaten sind nicht zulässig.	""Module_XA5""	'Module_XA5'	Ja
		""123456""	'123456'	Ja
		""""	"	Ja
		""A''''''B""	'A''B'	Ja
		""A''B""	'A''B'	Ja
		""','""	','	Ja
		""\$R\$N""	'\$R\$N'	Ja
		""AB\$00CD""	'AB' (String wurde abgeschnitten)	Nein

Beispiel:

```
PROGRAM MAIN
VAR
  s1 : STRING;
  s2 : STRING;
END_VAR

s1 := CSVFIELD_TO_STRING( '"ab_$04_$05_cd-"ALFA"_5"', TRUE );
s2 := CSVFIELD_TO_STRING( 'Module_50', FALSE );
```

Das Ergebnis:

s1 = 'ab_\$04_\$05_cd-"ALFA"_5'

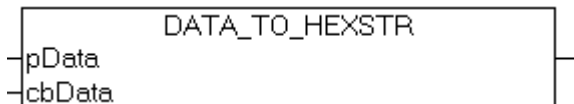
s2 = 'Module_50'

Weitere Informationen finden Sie hier: [Beispiel: Schreiben/lesen einer CSV-Datei \[▶ 351\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.22 DATA_TO_HEXSTR



Die Funktion konvertiert Binärdaten in einen Hexadezimalstring. Mit dieser Funktion können einfache Datentypen und Struktur-Variablen konvertiert werden. Die maximale Länge der Binärdaten darf aber 85 Byte nicht überschreiten! Beim Überschreiten der maximalen Länge wird dem Ergebnisstring ein Punkt-Zeichen hinzugefügt ('.') und die Konvertierung abgebrochen. Die verbleibenden Datenbytes werden nicht mehr konvertiert. Bei fehlerhaften Funktionsparametern (*pData* = Null oder *cbData* = Null) liefert die Funktion ein Leerstring als Ergebnis.

FUNCTION DATA_TO_HEXSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
  pData : POINTER TO BYTE;
  cbData : UDINT(0..85);
  bLoCase : BOOL := FALSE;
END_VAR
```

pData: Anfangsadresse (Pointer) auf die zu konvertierenden Binärdaten. Die Adresse kann mit dem ADR-Operator ermittelt werden.

cbData: Max. Länge der zu konvertierenden Binärdaten. Die Länge darf 85 Bytes nicht überschreiten! Die Länge kann mit dem SIZEOF-Operator ermittelt werden.

bLoCase: Dieser Parameter legt fest, ob Groß- oder Kleinbuchstaben bei der Konvertierung verwendet werden sollen. TRUE = Kleinbuchstaben, FALSE = Großbuchstaben.

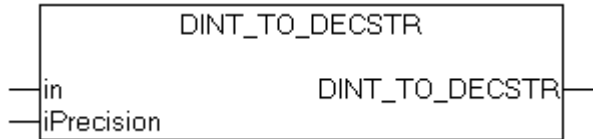
Beispiel:

Bitte beachten Sie, dass die Datengröße der *overflow*-Variablen die 85 Bytes überschreitet. Der Ergebnisstring *sh5* wird aus diesem Grund mit einem Punkt abgeschlossen.

Bei der *number*-Variablen ist die Bytefolge vertauscht, weil die Speicherorganisation der Zählervariablen dem Little-Endian-Format entspricht (auch Intel-Format genannt).

```
PROGRAM MAIN
VAR
  str : T_MaxString := 'abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
  number : DWORD := 16#BECF1234;
  char : BYTE := 16#07;
```


4.24 DINT_TO_DECSTR



Die Funktion konvertiert eine vorzeichenbehaftete Dezimalzahl in einen Dezimalstring (Basis 10).

FUNCTION DINT_TO_DECSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : DINT;
    iPrecision  : INT;
END_VAR
```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer ist als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring. Bei negativen Zahlen erscheint im resultierenden String zusätzlich das negative Vorzeichen.

Beispiel:

```
PROGRAM MAIN
VAR
    s1   : STRING;
    s2   : STRING;
    s3   : STRING;
    s4   : STRING;
    iCnt : INT;
END_VAR

iCnt := -1234;
s1   := DINT_TO_DECSTR( iCnt, 1);
s2   := DINT_TO_DECSTR( iCnt, 10 );
iCnt := 0;
s3   := DINT_TO_DECSTR( iCnt, 0 );
iCnt := 1234;
s4   := DINT_TO_DECSTR( iCnt, 10 );
```

Das Ergebnis:

s1 = '-1234'

s2 = '-0000001234'

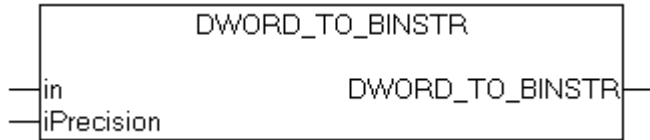
s3 = ''

s4 = '0000001234'

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.25 DWORD_TO_BINSTR



Die Funktion konvertiert eine Dezimalzahl in einen Binärstring (Basis 2).

FUNCTION DWORD_TO_BINSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in      : DWORD;
    iPrecision : INT;
END_VAR
```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Beispiel:

```
PROGRAM MAIN
VAR
    s1 : STRING;
    s2 : STRING;
    s3 : STRING;
    nCnt : BYTE;
END_VAR

s1 := DWORD_TO_BINSTR( 16#81, 16 );
nCnt := 15;
s2 := DWORD_TO_BINSTR( nCnt, 1 );
nCnt := 0;
s3 := DWORD_TO_BINSTR( nCnt, 0 );
```

Das Ergebnis:

s1 = '0000000010000001'

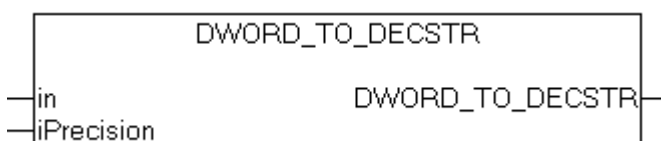
s2 = '1111'

s3 = ''

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.26 DWORD_TO_DECSTR



Die Funktion konvertiert eine Dezimalzahl in einen Dezimalstring (Basis 10).

FUNCTION DWORD_TO_DECSTR : T_MaxString**VAR_INPUT**

```

VAR_INPUT
  in      : DWORD;
  iPrecision : INT;
END_VAR

```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Beispiel:

```

PROGRAM MAIN
VAR
  s1 : STRING;
  s2 : STRING;
  s3 : STRING;
  nCnt : WORD;
END_VAR

nCnt := 43981;
s1 := DWORD_TO_DECSTR( nCnt, 1 );
s2 := DWORD_TO_DECSTR( nCnt, 10 );
nCnt := 0;
s3 := DWORD_TO_DECSTR( nCnt, 0 );

```

Das Ergebnis:

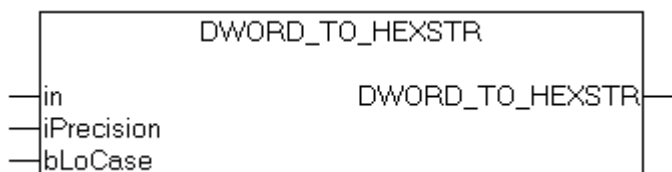
s1 = '43981'

s2 = '0000043981'

s3 = ''

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.27 DWORD_TO_HEXSTR

Die Funktion konvertiert eine Dezimalzahl in einen Hexadezimalstring (Basis 16).

FUNCTION DWORD_TO_HEXSTR : T_MaxString**VAR_INPUT**

```

VAR_INPUT
  in      : DWORD;
  iPrecision : INT;
  bLoCase : BOOL;
END_VAR

```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

bLoCase: Dieser Parameter bestimmt ob Klein- oder Großbuchstaben bei der Konvertierung benutzt werden sollen. FALSE => "ABCDEF", TRUE => "abcdef".

Beispiel:

```
PROGRAM MAIN
VAR
  s1   : STRING;
  s2   : STRING;
  s3   : STRING;
  s4   : STRING;
  nCnt : WORD;
END_VAR

nCnt := 43981;
s1 := DWORD_TO_HEXSTR( nCnt, 1, FALSE );
s2 := DWORD_TO_HEXSTR( nCnt, 1, TRUE );
nCnt := 15;
s3 := DWORD_TO_HEXSTR( nCnt, 4, FALSE );
nCnt := 0;
s4 := DWORD_TO_HEXSTR( nCnt, 0, FALSE );
```

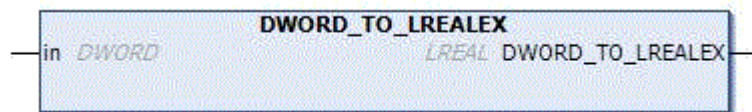
Das Ergebnis:

s1 = 'ABCD'
s2 = 'abcd'
s3 = '000F'
s4 = ''

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.28 DWORD_TO_LREALX



In TwinCAT 2 auf der ARM-Plattform wird die Konvertierung der vorzeichenlosen Zahlen in Fließkommazahlen von Typ: LREAL nicht unterstützt. Vorzeichenlose Zahlen mit einem gesetztem höchstwertigen Bit werden möglicherweise implizit in negative Fließkommazahlen konvertiert. Die hier beschriebene Funktion erlaubt in TwinCAT 2 eine explizite Konvertierung vom Typ DWORD in eine positive Fließkommazahl vom Typ LREAL (auch wenn das höchstwertige Bit gesetzt wurde und ohne Compiler-Warnung). Diese Funktion benötigen Sie nur um konvertierte TwinCAT 2 Projekte ohne Änderungen in TwinCAT 3 übersetzen zu können.

In TwinCAT 3 werden vorzeichenlose Zahlen von Typ DWORD immer (implizit und explizit) in eine positive Fließkommazahl Zahl konvertiert. Aus diesem Grund kann auf die Verwendung dieser Funktion verzichtet werden.

FUNCTION DWORD_TO_LREAL : LREAL**VAR_INPUT**

```
VAR_INPUT
  in : DWORD;
END_VAR
```

Beispiel:

```
PROGRAM MAIN
VAR
  nDword : DWORD := 16#FFFFFFFF;
  fLreal : LREAL := 0.0;
END_VAR
```

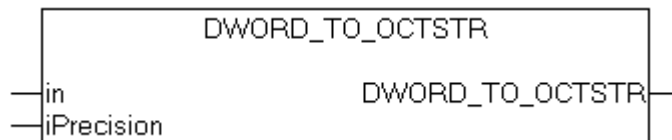
fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := nDword	-*1, Warning 1105**	4294967295	4294967295
fLreal := DWORD_TO_LREAL(nDword)	-*1, Warning 1105**	4294967295	4294967295
fLreal := DWORD#16#FFFFFFFF	-*1, Warning 1105**	4294967295	4294967295
fLreal := 16#FFFFFFF	4294967295	4294967295	4294967295
fLreal := DWORD_TO_LREAL(nDword)	4294967295	4294967295	4294967295
fLreal := DWORD_TO_LREAL(DWORD#16#FFFFFFFF)	4294967295	4294967295	4294967295
fLreal := DWORD_TO_LREAL(16#FFFFFFF)	4294967295	4294967295	4294967295

*not supported

**Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.29 DWORD_TO_OCTSTR

Die Funktion konvertiert eine Dezimalzahl in einen Oktalstring (Basis 8).

FUNCTION DWORD_TO_OCTSTR : T_MaxString**VAR_INPUT**

```
VAR_INPUT
  in : DWORD;
  iPrecision : INT;
END_VAR
```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Beispiel:

```
PROGRAM MAIN
VAR
  s1   : STRING;
  s2   : STRING;
  s3   : STRING;
  nCnt : WORD;
END_VAR

nCnt := 43981;
s1   := DWORD_TO_OCTSTR( nCnt, 1 );
s2   := DWORD_TO_OCTSTR( nCnt, 10 );
nCnt := 0;
s3   := DWORD_TO_OCTSTR( nCnt, 0 );
```

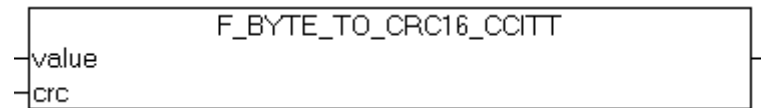
Das Ergebnis:

s1 = '125715'
s2 = '0000125715'
s3 = ''

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.30 F_BYTE_TO_CRC16_CCITT



Mit der Funktion "F_BYTE_TO_CRC16_CCITT" kann eine 16-Bit-CRC-CCITT (zyklische Redundanzprüfung) von einzelnen Datenbytes ermittelt werden.

Der benutzte generator Polynom: Name : CRC-16 CCITT

- Standard : CRC-CCITT
- Referenzen : ITU X.25/T.30, ADCCP, SDLC/HDLC, ...
- Polynomial value : 0x1021
- Polynom : $x^{16} + x^{12} + x^5 + 1$

FUNCTION F_BYTE_TO_CRC16_CCITT : WORD

VAR_INPUT

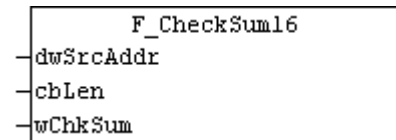
```
VAR_INPUT
  value : BYTE; (* Data value *)
  crc   : WORD; (* Initial value (16#FFFF or 16#0000) or previous CRC-16 result *)
END_VAR
```

value: Der zu konvertierende Datenbyte.

crc: Initialwert = 16#FFFF oder 16#0000 oder der letzte CRC.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.31 F_CheckSum16

Mit der Funktion "F_CheckSum16" kann eine 16-Bit-Checksumme von beliebigen Daten ermittelt werden.

FUNCTION F_CheckSum16 : WORD**VAR_INPUT**

```

VAR_INPUT
  dwSrcAddr  : POINTER TO BYTE;
  cbLen      : UDINT;
  wChkSum    : WORD;
END_VAR

```

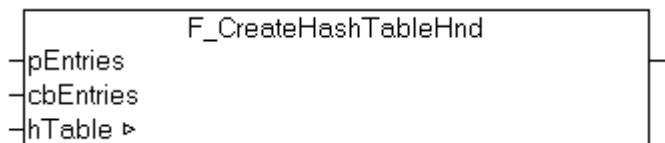
dwSrcAddr: Adresse des Datenpuffers.

cbLen: Länge des Datenpuffers.

wChkSum: Initialwert = 0 oder letzte Checksumme.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.32 F_CreateHashTableHnd

Die Funktion Initialisiert das Hash-Table-Handle. Das Tabellen-Handle muss einmalig durch Aufruf der F_CreateHashTableHnd-Funktion initialisiert werden.

FUNCTION F_CreateHashTableHnd : BOOL**VAR_INPUT**

```

VAR_INPUT
  pEntries  : POINTER TO T_HashTableEntry := 0;
  cbEntries : UDINT := 0;
END_VAR

```

pEntries: Adresse von dem ersten T_HashTableEntry-Arrayelement. Die Adresse kann mit dem ADR-Operator ermittelt werden (Typ: [T_HashTableEntry](#) [▶ 325]).

cbEntries: Die T_HashTableEntry-Bytegröße . Die Bytegröße kann mit dem SIZEOF-Operator ermittelt werden.

VAR_IN_OUT

```
VAR_IN_OUT
    hTable : T_HHASHTABLE;
END_VAR
```

hTable: Hash-Tabellen-Handle welches initialisiert werden soll (Typ: [T_HHASHTABLE \[▶ 325\]](#)). Das Handle wird bei den Zugriffen auf die Hash-Tabelle von dem Funktionsbaustein [FB_HashTableCtrl \[▶ 80\]](#) benötigt.

Rückgabeparameter	Beschreibung
TRUE	Kein Fehler
FALSE	Fehler

Beispiel:

Siehe: [Beispiel: Hash-Tabelle \(FB_HashTableCtrl\). \[▶ 341\]](#)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.33 F_CreateLinkedListHnd



Die Funktion Initialisiert das Linked-List-Handle. Das List-Handle muss einmalig durch Aufruf der `F_CreateLinkedListHnd`-Funktion initialisiert werden.

FUNCTION F_CreateLinkedListHnd : BOOL

VAR_INPUT

```
VAR_INPUT
    pEntries : POINTER TO T_LinkedListEntry := 0;
    cbEntries : UDINT := 0;
END_VAR
```

pEntries: Adresse von dem ersten `T_LinkedListEntry`-Arrayelement. Die Adresse kann mit dem `ADR`-Operator ermittelt werden (Typ: [T_LinkedListEntry \[▶ 326\]](#)).

cbEntries: Die Bytegröße des `T_LinkedListEntry`-Arrays. Die Bytegröße kann mit dem `SIZEOF`-Operator ermittelt werden.

VAR_IN_OUT

```
VAR_IN_OUT
    hList : T_HLINKEDLIST;
END_VAR
```

hList: Hash-Tabellen-Handle welches initialisiert werden soll (Typ: [T_HLINKEDLIST \[▶ 325\]](#)). Das Handle wird bei den Zugriffen auf die Liste von dem Funktionsbaustein [FB_LinkedListCtrl \[▶ 88\]](#) benötigt.

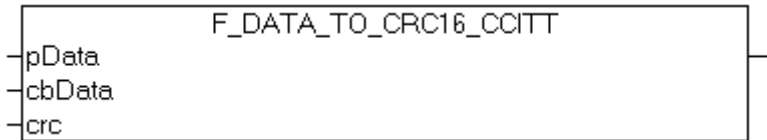
Rückgabeparameter	Beschreibung
TRUE	Kein Fehler
FALSE	Fehler

Beispiel:

Siehe: [Beispiel: Verkettete-Liste \(FB_LinkedListCtrl\). \[▶ 345\]](#)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.34 F_DATA_TO_CRC16_CCITT

Mit der Funktion "F_DATA_TO_CRC16_CCITT" kann eine 16-Bit-CRC-CCITT (zyklische Redundanzprüfung) von beliebigen Daten ermittelt werden. Intern wird die Funktion F_BYTE_TO_CRC16_CCITT [[▶ 263](#)] benutzt.

Weitere Informationen zum benutzten Algorithmus finden Sie in der Dokumentation der F_BYTE_TO_CRC16_CCITT [[▶ 263](#)]-Funktion.

FUNCTION F_DATA_TO_CRC16_CCITT: WORD**VAR_INPUT**

```

VAR_INPUT
  pData  : POINTER TO BYTE; (* Pointer to first data byte *)
  cbData : UDINT; (* Length of data *)
  crc    : WORD; (* Initial value (16#FFFF or 16#0000) or previous CRC-16 result *)
END_VAR

```

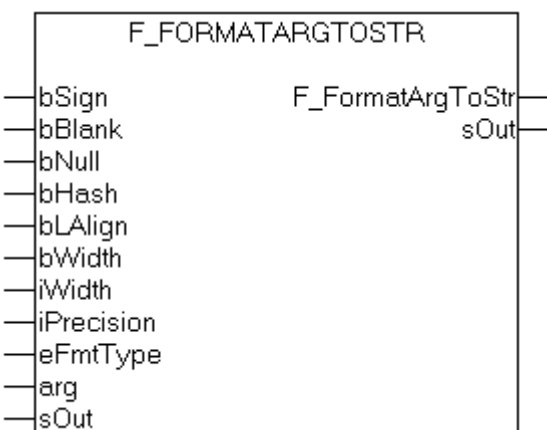
pData: Adresse des Datenpuffers.

cbData: Länge des Datenpuffers.

crc: Initialwert = 16#FFFF oder 16#0000 oder der letzte CRC.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.35 F_FormatArgToStr

Format-Hilfsfunktion. Diese Funktion wird auch intern von dem [FB FormatString \[▶ 64\]](#)-Funktionsbaustein benutzt. Mit dieser Funktion kann eine Variable vom Typ [T_Arg \[▶ 323\]](#) entsprechend der [Formatspezifikation \[▶ 356\]](#) in einen formatierten String konvertiert werden.

FUNCTION F_FormatArgToStr : UDINT

VAR_INPUT

```
VAR_INPUT
  bSign      : BOOL; (* Sign prefix flag *)
  bBlank     : BOOL; (* Blank prefix flag *)
  bNull      : BOOL; (* Null prefix flag *)
  bHash      : BOOL; (* Hash prefix flag *)
  bLAlign    : BOOL; (* FALSE => Right align (default), TRUE => Left align *)
  bWidth     : BOOL; (* FALSE => no width padding, TRUE => blank or zeros padding enabled *)
  iWidth     : INT; (* Width length parameter *)
  iPrecision : INT; (* Precision length parameter *)
  eFmtType   : E_TypeFieldParam; (* Format type field parameter *)
  arg        : T_Arg; (* Format argument *)
END_VAR
```

bSign: Das Vorzeichen-Flag.

bBlank: Das Leerzeichen-Flag.

bNull: Das Null-Flag.

bHash: Das Hash-Prefix-Flag.

bLAlign: Das Ausrichtungs-Flag (TRUE=left align).

bWidth: Wenn TRUE, dann wird der iWidth-Parameter ausgewertet, sonst nicht.

iWidth: Width-Parameter.

iPrecision: Precision-Parameter.

eFmtType: Type-Parameter (Typ: [E_TypeFieldParam \[▶ 308\]](#)).

arg: Das zu formatierende Argument. Folgende Hilfsfunktionen können benutzt werden um SPS-Variablen unterschiedlichsten Typs in den benötigten Datentyp [T_Arg \[▶ 323\]](#) zu konvertieren: [F_BYTE \[▶ 233\]](#), [F_WORD \[▶ 240\]](#), [F_DWORD \[▶ 234\]](#), [F_LWORD \[▶ 236\]](#), [F_SINT \[▶ 237\]](#), [F_INT \[▶ 235\]](#), [F_DINT \[▶ 233\]](#), [F_LINT \[▶ 235\]](#), [F_USINT \[▶ 240\]](#), [F_UINT \[▶ 239\]](#), [F_UDINT \[▶ 238\]](#), [F_ULINT \[▶ 240\]](#), [F_STRING \[▶ 237\]](#), [F_REAL \[▶ 237\]](#), [F_LREAL \[▶ 236\]](#).

VAR_IN_OUT

```
VAR_IN_OUT
  sOut      : T_MaxString;
END_VAR
```

sOut: Beim Erfolg liefert diese Variable den formatierten Ausgabestring (Typ: [T_MaxString](#)).

Rückgabeparameter	Bedeutung
0	Kein Fehler
<> 0	Fehler. Fehlerbeschreibung finden Sie unter: Format Fehlercodes [▶ 359]

Beispiele:

Formatieren einer BYTE-Variablen als Binärstring.

```
PROGRAM MAIN
VAR
  s1      : T_MaxString;
  s2      : T_MaxString;
  s3      : T_MaxString;
  s4      : T_MaxString;
  s5      : T_MaxString;
  errID   : UDINT;
```

```

    varByte : BYTE;
    double  : LREAL;
    L1      : INT;
    L2      : INT;
    L3      : INT;
    L4      : INT;
    L5      : INT;
END_VAR

varByte := 128;
errID   := F_FormatArgToStr(FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, 20, 8, TYPEFIELD_B, F_BYTE( varByte ), s1 );
errID   := F_FormatArgToStr(FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, 20, 8, TYPEFIELD_B, F_BYTE( varByte ), s2 );
errID   := F_FormatArgToStr(FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, 20, 8, TYPEFIELD_B, F_BYTE( varByte ), s3 );
errID   := F_FormatArgToStr(FALSE, FALSE, FALSE, TRUE, TRUE, TRUE, 20, 8, TYPEFIELD_B, F_BYTE( varByte ), s4 );
L1 := LEN( s1 );
L2 := LEN( s2 );
L3 := LEN( s3 );
L4 := LEN( s4 );

```

Das Ergebnis:

```

s1 = '10000000'
s2 = '      10000000'
s3 = '10000000      '
s4 = '2#10000000      '
L1 = 8
L2 = 20
L3 = 20
L4 = 20

```

Formatieren einer LREAL-Variablen.

```

double := 12345.6789;
errID   := F_FormatArgToStr( FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, 20, 8, TYPEFIELD_F, F_LREAL( double ), s1 );
errID   := F_FormatArgToStr( FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, 20, 8, TYPEFIELD_F, F_LREAL( double ), s2 );
errID   := F_FormatArgToStr( FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, 20, 8, TYPEFIELD_F, F_LREAL( double ), s3 );
errID   := F_FormatArgToStr( FALSE, FALSE, TRUE, FALSE, FALSE, TRUE, 20, 8, TYPEFIELD_F, F_LREAL( double ), s4 );
errID   := F_FormatArgToStr( TRUE, FALSE, FALSE, TRUE, TRUE, TRUE, 20, 8, TYPEFIELD_F, F_LREAL( double ), s5 );
L1 := LEN( s1 );
L2 := LEN( s2 );
L3 := LEN( s3 );
L4 := LEN( s4 );
L5 := LEN( s5 );

```

Das Ergebnis:

```

s1 = '12345.67890000'
s2 = ' 12345.67890000'
s3 = '12345.67890000  '
s4 = '00000012345.67890000'
s5 = '+12345.67890000  '
L1 = 14
L2 = 20
L3 = 20
L4 = 20
L5 = 20

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.36 F_GenerateHashValue

Mit der Funktion kann ein Hash-Wert berechnet werden.

Bei erfolgreicher Berechnung gibt die Funktion TRUE zurück.

FUNCTION F_GenerateHashValue : BOOL

VAR_INPUT

```
VAR_INPUT
    hashMode : E_HashMode;
    pData    : PVOID;
    nData    : UDINT;
    pHash    : PVOID;    // destination buffer for generated hash value
    nHash    : UDINT;    // size of destination buffer in bytes. This needs to match the hash mode.
END_VAR
```

hashMode: Hier wird ein Hash Mode, wie beispielsweise SHA 512, angegeben. Siehe [E_HashMode](#) [► 303].

pData: Hier wird die Adresse der Eingangsdaten angegeben.

nData: Hier wird die Größe der Eingangsdaten in Bytes angegeben.

pHash: Hier wird die Adresse des Puffers angegeben, wohin der Hash-Wert gespeichert werden soll.

nHash: Hier wird die Größe des Puffers für den Hash-Wert in Bytes angegeben. Die Größe ist abhängig vom Hash Mode, siehe auch [E_HashMode](#) [► 303]. Es muss die passende Größe angegeben werden. Andernfalls schlägt die Funktion fehl.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024.29	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.51.0

4.37 F_GetClassIdVersioned

Die Funktion berechnet anhand der Class ID und der Library ID die versionierte Class ID. Anwendung findet dies bei [versionierten C++ Projekten](#).

FUNCTION F_GetClassIdVersioned : BOOL

VAR_INPUT

```
VAR_INPUT
    sLibraryId : STRING(255); // 'vendorName|libraryName|libraryVersion' (e.g. 'C+
+ Module Vendor|IncrementerCpp|0.0.0.1' )
    clsId      : CLSID;
    clsIdVersioned : REFERENCE TO CLSID;
END_VAR
```

sLibraryId: Hier wird die Library ID angegeben.

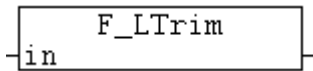
clsId: Hier wird die Class ID angegeben.

clsIdVersioned: Hier wird die errechnete versionierte Class ID ausgegeben.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024.29	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.51.0

4.38 F_LTrim



Entfernt von der Zeichenfolge führende Leerzeichen und gibt die reduzierte Zeichenfolge zurück.

FUNCTION F_LTrim : T_MaxString

VAR_INPUT

```

VAR_INPUT
  in : T_MaxString;
END_VAR
  
```

in: Der zu konvertierende String (Typ: T_MaxString).

Beispiel:

```

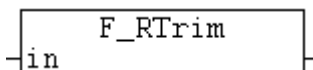
PROGRAM MAIN
VAR
  sLTrim : STRING;
END_VAR

sLTrim := F_LTrim(' <trim ');(* result: '<trim' *)
sLTrim := F_LTrim(' <trim');(* result: '<trim' *)
sLTrim := F_LTrim('<trim');(* result: '<trim' *)
sLTrim := F_LTrim('');(* result: '' *)
  
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.39 F_RTrim



Schneidet vom angegebenen Wert alle Leerzeichen am Ende ab und gibt das Ergebnis zurück.

FUNCTION F_RTrim : T_MaxString

VAR_INPUT

```

VAR_INPUT
  in : T_MaxString;
END_VAR
  
```

in: Der zu konvertierende String (Typ: T_MaxString).

Beispiel:

```

PROGRAM MAIN
VAR
  sRTrim : STRING;
  sLTrim : STRING;
END_VAR

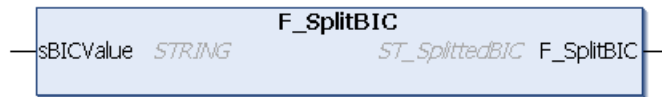
sRTrim := F_RTrim(' trim> ');(* result: ' trim>' *)
sRTrim := F_RTrim('trim> ');(* result: 'trim>' *)
sRTrim := F_RTrim('trim>');(* result: 'trim>' *)
sRTrim := F_RTrim('');(* result: '' *)

sLTrim := F_RTrim( F_LTrim( ' <trim> '));(* result: '<trim>' *)
  
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.40 F_SplitBIC



Die Funktion F_SplitBIC zerlegt den Beckhoff Identification Code (BIC) sBICValue anhand von bekannten Kennungen in seine Bestandteile und liefert die erkannten Teil-Strings in einer Struktur ST_SplittedBIC [► 319] als Rückgabewert. Leerzeichen am Ende der Teilstrings werden automatisch entfernt. Unbenutzte Teilstrings werden als Leerstring zurückgeliefert. Wird eine unbekannte Kennung gefunden, dann wird der Reststring der BIC als sUndefined in der Rückgabestruktur weitergereicht. Die BIC kann z.B. mit den Bausteinen FB_EcCoEReadBIC bzw. FB_EcReadBIC (Tc2_EtherCAT Library) aus EtherCAT Slaves ausgelesen werden.

FUNCTION F_SplitBIC : ST_SplittedBIC

VAR_INPUT

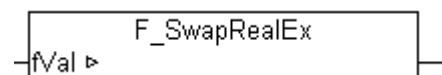
```
VAR_INPUT
  sBICValue   : STRING;
END_VAR
```

Name	Typ	Beschreibung
sBICValue	STRING	Dieser Eingang muss den Beckhoff Identification Code (BIC) enthalten, die mit der Funktion F_SplitBIC in Teilstrings zerlegt wird, z.B. „1P193995SBTN0002agdw1KEL7411 Q1 2P112104020018“.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.41 F_SwapRealEx



Die Speicherdarstellung einer REAL-Zahl auf einem Busklemmen Controller (z.B. BC2000, BC3100, BC9000) unterscheidet sich von der Speicherdarstellung einer REAL-Zahl auf einem x86/x64/ARM-System (IPC oder Embedded PC).

Um eine REAL-Zahl eines Busklemmen Controllers auf einem IPC richtig darstellen zu können müssen die Hi- und Lo-Words der REAL-Zahl vertauscht werden. Die Programmierumgebung macht dies bereits im Online- oder Simulations-Mode. Um die REAL-Daten eines Buscontrollers über das Netzwerk (ADS-Protokoll, ADSDLL, AdsOcx usw.) anzufordern und auf einem x86/x64/ARM-IPC richtig darzustellen, müssen die REAL-Daten in das richtige Format konvertiert werden. Dieses kann auf der Busklemmen Controller- oder IPC-Seite erfolgen.

Mit der Funktion F_SwapRealEx können die REAL-Variablen (die z.B. von einer VB-Applikation eingelesen werden oder mit TwinCAT Scope View aufgezeichnet werden sollen) auf der PC-Seite in das passende Format konvertiert werden. Die Funktion verändert die Speicherdarstellung des übergebenen fVal-Parameters (VAR_IN_OUT).

FUNCTION F_SwapRealEx : BOOL**VAR_IN_OUT**

```
VAR_IN_OUT
  fVal : REAL;
END_VAR
```

fVal: Der zu konvertierende REAL Wert.

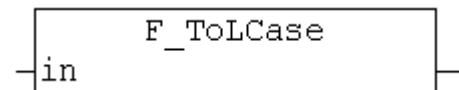
Rückgabeparameter	Bedeutung
TRUE	Kein Fehler
FALSE	Fehler bei der Funktionsausführung

Beispiel:

Siehe: [Beispiel: Kommunikation BC/BX<->PC/CX \(F_SwapRealEx\)](#). [► 331]

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.42 F_ToLCase

Die F_ToLCase-Funktion wandelt alle Großbuchstaben einer Zeichenfolge in Kleinbuchstaben um.

● Zeichensatz



Standardmäßig verwendet die Konvertierungsfunktion den Zeichensatz der Windows Code Page 1252 Latin 1, SBCS (Single Byte Character Set). Ein anderer Zeichensatz kann zur Laufzeit (zurzeit aber nur Windows Code Page 1250 Central European) über die globale Variable GLOBAL_SBCS_TABLE angewählt werden (siehe Beispiel).

FUNCTION F_ToLCase : T_MaxString**VAR_INPUT**

```
VAR_INPUT
  in : T_MaxString;
END_VAR
```

in: Der zu konvertierende String (Typ: T_MaxString).

Beispiel:

```
PROGRAM MAIN
VAR
  sLCase : STRING;
END_VAR

sLCase := F_ToLCase( 'TO LOWER CASE 1234567890 ÄÖÜß' );
```

Das Ergebnis der Konvertierung ist: 'to lower case 1234567890 äöüß'

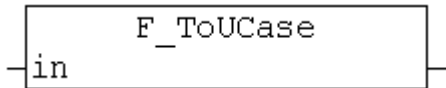
```
GLOBAL_SBCS_TABLE := eSBCS_CentralEuropean;
sLCase := F_ToLCase( 'TO LOWER CASE 1234567890 ĄĘŚĆŹŻŁÓ' );
```

Das Ergebnis der Konvertierung ist: 'to lower case 1234567890 ąęśćźżłó'

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.43 F_ToUCase



Die F_ToUCase-Funktion wandelt alle Kleinbuchstaben einer Zeichenfolge in Großbuchstaben um.

Zeichensatz

i Standardmäßig verwendet die Konvertierungsfunktion den Zeichensatz der Windows Code Page 1252 Latin 1, SBCS (Single Byte Character Set). Ein anderer Zeichensatz kann zur Laufzeit (zurzeit aber nur Windows Code Page 1250 Central European) über die globale Variable GLOBAL_SBCS_TABLE angewählt werden (siehe Beispiel).

FUNCTION F_ToUCase : T_MaxString

VAR_INPUT

```

VAR_INPUT
  in : T_MaxString;
END_VAR
  
```

in: Der zu konvertierende String (Typ: T_MaxString).

Beispiel:

```

PROGRAM MAIN
VAR
  sUCase : STRING;
END_VAR

sUCase := F_ToUCase( 'to upper case 1234567890 äöüß' );
  
```

Das Ergebnis der Konvertierung ist: 'TO UPPER CASE 1234567890 ÄÖÜß'

```

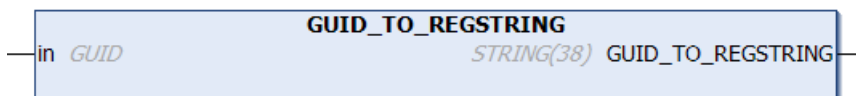
GLOBAL_SBCS_TABLE := eSBCS_CentralEuropean;
sUCase := F_ToUCase( 'to upper case 1234567890 aęśćźżłó' );
  
```

Das Ergebnis der Konvertierung ist: 'TO UPPER CASE 1234567890 AĘŚĆŹŹŁÓ'

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.44 GUID_TO_REGSTRING



Diese Funktion konvertiert eine strukturierte GUID [▶ 309]-Variable in eine Registry-GUID-Stringvariable (eingeschlossen in geschweifte Klammern).

FUNCTION GUID_TO_REGSTRING : STRING(38)

VAR_INPUT

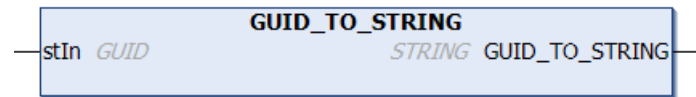
```
VAR_INPUT
  in : GUID;
END_VAR
```

Rückgabewert	Bedeutung
'{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}'	Kein Fehler ('x' ist ein Hexadezimal-Halbbyte)
'{00000000-0000-0000-0000-000000000000}'	Kein Fehler, GUID hat den Initialwert (alle Bytes sind Null)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.45 GUID_TO_STRING



Diese Funktion konvertiert eine strukturierte GUID [► 309]-Variable in eine GUID-Stringvariable (ohne geschweifte Klammern).

FUNCTION GUID_TO_STRING: STRING

VAR_INPUT

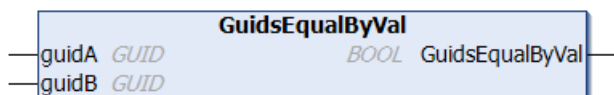
```
VAR_INPUT
  stIn : GUID;
END_VAR
```

Rückgabewert	Bedeutung
'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX'	Kein Fehler ('x' ist ein Hexadezimal-Halbbyte)
00000000-0000-0000-0000-000000000000'	Kein Fehler, GUID hat den Initialwert (alle Bytes sind Null)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.46 GuidEqualByVal



Diese Funktion vergleicht zwei GUID Werte.

FUNCTION GuidsEqualByVal : BOOL

VAR_INPUT

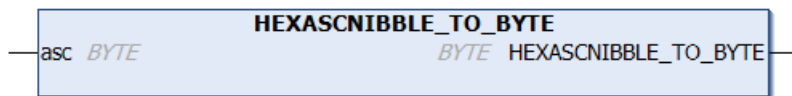
```
VAR_INPUT
    guidA : GUID;
    guidB : GUID;
END_VAR
```

Rückgabewert	Bedeutung
FALSE	guidA <> guidB
TRUE	guidA = guidB

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.47 HEXASCNIBBLE_TO_BYTE



Diese Funktion konvertiert den ASCII-Code eines hexadezimalen Halbbytezeichens in den Dezimalwert.

FUNCTION HEXASCNIBBLE_TO_BYTE : BYTE

VAR_INPUT

```
VAR_INPUT
    asc : BYTE;
END_VAR
```

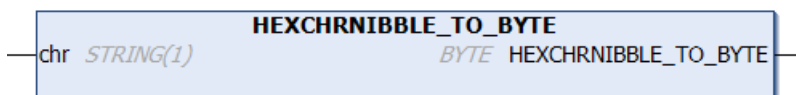
asc: Ascii-Code eines hexadezimalen Halbbytezeichens (Ascii code von: ‚0‘ bis ‚9‘ oder ‚a‘ bis ‚f‘ oder ‚A‘ bis ‚F‘).

Rückgabewert	Bedeutung
0 bis 15	Erfolgreich, kein Fehler.
255	Fehler, fehlerhafter Eingangsparameterwert.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.48 HEXCHRNIBBLE_TO_BYTE



Diese Funktion konvertiert ein hexadezimalen Halbbytezeichen in seinen dezimalen Wert.

FUNCTION HEXCHRNIBBLE_TO_BYTE : BYTE

VAR_INPUT

```
VAR_INPUT
    chr : STRING(1);
END_VAR
```

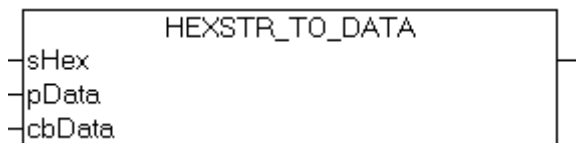
chr: Hexadezimaler Halbbyte-Zeichen (,0' bis ,9' oder ,a' bis ,f' oder ,A' bis ,F').

Rückgabewert	Bedeutung
0 bis 15	Erfolgreich, kein Fehler.
255	Fehler, falscher Eingangsparameterwert.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.49 HEXSTR_TO_DATA



Die Funktion konvertiert einen Hexadezimalstring in Binärdaten und liefert die Anzahl der erfolgreich konvertierten Datenbytes als Ergebnis zurück. Es dürfen nur Leerzeichen als Trennzeichen in dem zu konvertierenden Hexadezimalstring verwendet werden. Bei den Hex-Zeichen sind Klein- und Großbuchstaben zugelassen. Bei einem Fehler oder unerlaubten Zeichen wird die Konvertierung abgebrochen und eine Null-Länge als Ergebnis zurückgeliefert.

FUNCTION HEXSTR_TO_DATA : UDINT

VAR_INPUT

```

VAR_INPUT
    sHex    : T_MaxString;
    pData   : POINTER TO BYTE;
    cbData  : UDINT;
END_VAR
  
```

sHex: Der zu konvertierende Hexadezimalstring (Typ: T_MaxString, z.B.: 'AB CD 01 23').

pData: Anfangsadresse (Pointer) auf den Zielpuffer in den die konvertierten Datenbytes geschrieben werden sollen. Die Adresse kann mit dem ADR-Operator ermittelt werden.

cbData: Max. verfügbare Länge des Zielpuffers. Die Länge kann mit dem SIZEOF-Operator ermittelt werden.

Beispiel:

```

PROGRAM MAIN
VAR
    sH    : STRING := 'AB CD EF 01 23 45 67 89';
    data  : ARRAY[0..10] OF BYTE;
    cbData : UDINT;
END_VAR

cbData := HEXSTR_TO_DATA( sH, ADR( data ), SIZEOF( data ) );
  
```

Das Ergebnis (Online):

TwinCAT_Device.Untitled2.P_TEST2					
Expression	Type	Value	Prepared value	Address	Comment
◆ sH	STRING	'AB CD EF 01 23 45 67 89'			
▣ ◆ data	ARRAY [0..10] OF BYTE				
◆ data[0]	BYTE	16#AB			
◆ data[1]	BYTE	16#CD			
◆ data[2]	BYTE	16#EF			
◆ data[3]	BYTE	16#01			
◆ data[4]	BYTE	16#23			
◆ data[5]	BYTE	16#45			
◆ data[6]	BYTE	16#67			
◆ data[7]	BYTE	16#89			
◆ data[8]	BYTE	16#00			
◆ data[9]	BYTE	16#00			
◆ data[10]	BYTE	16#00			
◆ cbData	UDINT	16#00000008			

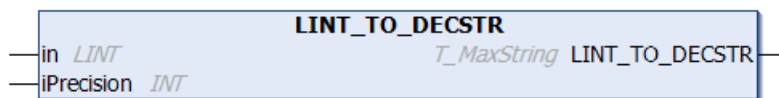

```

1  cbData[16#00000008] := HEXSTR_TO_DATA( sH[ 'AB CD EF 01 23 45 67 89' ], ADR( data ), SIZEOF( data ) );RETURN
    
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.50 LINT_TO_DECSTR



Die Funktion konvertiert eine vorzeichenbehaftete Dezimalzahl in einen Dezimalstring (Basis 10).

FUNCTION LINT_TO_DECSTR : T_MaxString

VAR_INPUT

```

VAR_INPUT
    in          : LINT;
    iPrecision  : INT;
END_VAR
    
```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer ist als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring. Bei negativen Zahlen erscheint im resultierenden String zusätzlich das negative Vorzeichen.

Beispiel:

```

PROGRAM MAIN
VAR
    s1   : STRING;
    s2   : STRING;
    s3   : STRING;
    s4   : STRING;
    iCnt : LINT;
END_VAR
    
```

```

iCnt := -1234;
s1  := LINT_TO_DECSTR( iCnt, 1 );
s2  := LINT_TO_DECSTR( iCnt, 10 );
iCnt := 0;
s3  := LINT_TO_DECSTR( iCnt, 0 );
iCnt := 1234;
s4  := LINT_TO_DECSTR( iCnt, 10 );

```

Das Ergebnis:

s1 = '-1234'

s2 = '-0000001234'

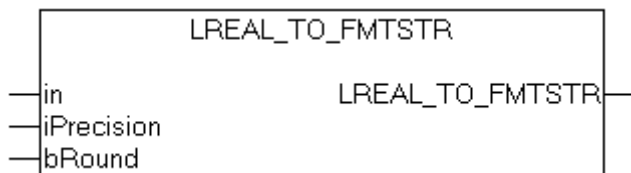
s3 = ''

s4 = '0000001234'

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.51 LREAL_TO_FMTSTR



Die Funktion konvertiert und formatiert eine Fließkommazahl in eine String-Variablen mit dem folgenden Format: [-]**dddd.dddd** (dddd sind Dezimalzahlen). Die Anzahl der Zahlen vor dem Dezimalpunkt hängt von dem Wert der Fließkommazahl ab. Die Anzahl der Zahlen hinter dem Dezimalpunkt hängt von der geforderten Präzision ab. Ein Vorzeichen erscheint nur bei negativen Werten. Bei einem unendlichen positiven Wert wird **'#INF'** und bei unendlichem negativen Wert: **'-#INF'** zurückgeliefert. Wenn die übergebene Variable einen unzulässigen Wert besitzt (NaN, Not-a-Number), dann wird **'#QNAN'** oder **'-#QNAN'** zurückgeliefert. Wenn die Länge des formatierten Strings die maximal zulässige Länge des resultierenden Strings überschreitet, dann wird **'#OVF'** oder **'-#OVF'** zurückgeliefert.

FUNCTION LREAL_TO_FMTSTR : STRING(510)

VAR_INPUT

```

VAR_INPUT
  in      : LREAL;
  iPrecision : INT;
  bRound  : BOOL;
END_VAR

```

in: Fließkommazahl die konvertiert und formatiert werden soll.

iPrecision: Präzision. Der Wert bestimmt die Anzahl der Zahlen hinter dem Dezimalpunkt. Bei dem Minimalwert (Null) erscheinen keine Nachkommastellen, der Maximalwert von *iPrecision* wird durch die Anzahl der Zahlen vor dem Dezimalpunkt und die maximal zulässige Länge des resultierenden Strings begrenzt. Wenn *in* = 0 und *iPrecision* = 0 dann wird String '0' zurückgeliefert.

bRound: Beim gesetzten *bRound*-Parameter wird der formatierte String auf die entsprechenden Anzahl der Nachkommastellen (*iPrecision*) gerundet. Beim Runden gilt folgende Regel: Hat die Dezimalzahl hinter der letzten gewünschten Nachkommastelle den Wert ≥ 5 dann wird aufgerundet sonst nicht.

Beispiel 1:

Die Zahl 0.46523 soll in einen String mit zwei Nachkommastellen konvertiert und gerundet werden.

```
sOut := LREAL_TO_FMTSTR( 0.46523, 2, TRUE );
```

Das Ergebnis ist: '0.47';

Beispiel: 2



Die maximale Anzahl der signifikanten Dezimalstellen bei den LREAL-Variablen ist auf 15-Stellen begrenzt.

Bedingt durch die interne Darstellung der Fließkommazahlen und Rundungsfehler während der Konvertierung kann es vorkommen, dass der resultierende String nicht exakt dem Wert der *in*-Variable entspricht.

```
PROGRAM MAIN
VAR
    double : LREAL;
    s1      : STRING;
    s2      : STRING;
    s3      : STRING;
    s4      : STRING;
END_VAR

double := 0.5;
s1 := LREAL_TO_FMTSTR( double, 25, FALSE );
s2 := LREAL_TO_FMTSTR( double, 2, FALSE );
s3 := LREAL_TO_FMTSTR( double, 0, TRUE );
s4 := LREAL_TO_FMTSTR( double, 2, TRUE );
```

Das Ergebnis ist:

s1 = '0.4999999999999999756000000' Dies ist die interne Darstellung der *double*-Variablen. Diese Zahl wird als Ausgangspunkt für die Rundungsoperation benutzt.

s2 = '0.49'

Durch die Rundung ergeben sich dann folgende Ergebnisse:

s3 = '0'

s4 = '0.50'

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.52 LWORD_TO_BASE36STR

Die Funktion konvertiert eine Dezimalzahl in einen Base36-String (Basis 16). Bei Zahlen mit Basis 16 werden zur Darstellung neben den Ziffern 0-9 ebenso die Buchstaben A-Z verwendet.

FUNCTION LWORD_TO_BASE36STR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : LWORD;
    iPrecision  : INT;
    bLoCase    : BOOL;
END_VAR
```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Haben der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

bLoCase: Dieser Parameter bestimmt, ob Klein- oder Großbuchstaben bei der Konvertierung benutzt werden sollen. FALSE => "ABCDEFXY", TRUE => "abcdefxy".

Beispiel:

```
PROGRAM MAIN
VAR
  s1   : STRING;
  s2   : STRING;
  s3   : STRING;
  s4   : STRING;
  nCnt : LWORD;
END_VAR

nCnt := 43981;
s1   := LWORD_TO_BASE36STR( nCnt, 1, FALSE );
s2   := LWORD_TO_BASE36STR( nCnt, 1, TRUE );
nCnt := 15;
s3   := LWORD_TO_BASE36STR( nCnt, 4, FALSE );
nCnt := 0;
s4   := LWORD_TO_BASE36STR( nCnt, 0, FALSE );
```

Das Ergebnis:

s1 = 'XXP'

s2 = 'xpx'

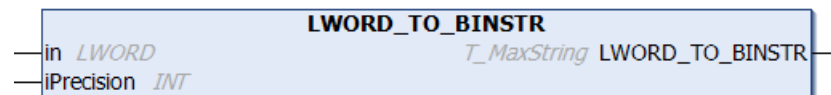
s3 = '000F'

s4 = ''

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.47.0

4.53 LWORD_TO_BINSTR



Die Funktion konvertiert eine Dezimalzahl in einen Binärstring (Basis 2).

FUNCTION LWORD_TO_BINSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
  in       : LWORD;
  iPrecision : INT;
END_VAR
```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Beispiel:

```
PROGRAM MAIN
VAR
  s1   : STRING;
  s2   : STRING;
```

```
s3 : STRING;
nCnt : LWORD;
END_VAR

s1 := LWORD_TO_BINSTR( 16#81, 16 );
nCnt := 15;
s2 := LWORD_TO_BINSTR( nCnt, 1 );
nCnt := 0;
s3 := LWORD_TO_BINSTR( nCnt, 0 );
```

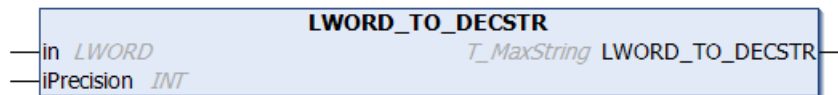
Das Ergebnis:

```
s1 = '0000000010000001'
s2 = '1111'
s3 = ''
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.54 LWORD_TO_DECSTR



Die Funktion konvertiert eine Dezimalzahl in einen Dezimalstring (Basis 10).

FUNCTION LWORD_TO_DECSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
in : LWORD;
iPrecision : INT;
END_VAR
```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Beispiel:

```
PROGRAM MAIN
VAR
s1 : STRING;
s2 : STRING;
s3 : STRING;
nCnt : LWORD;
END_VAR

nCnt := 43981;
s1 := LWORD_TO_DECSTR( nCnt, 1 );
s2 := LWORD_TO_DECSTR( nCnt, 10 );
nCnt := 0;
s3 := LWORD_TO_DECSTR( nCnt, 0 );
```

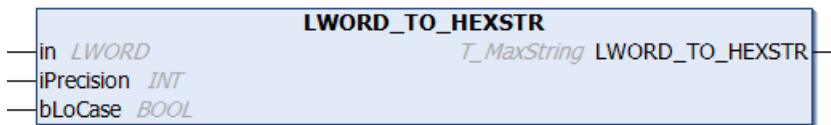
Das Ergebnis:

```
s1 = '43981'
s2 = '0000043981'
s3 = ''
```


Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.55 LWORD_TO_HEXSTR



Die Funktion konvertiert eine Dezimalzahl in einen Hexadezimalstring (Basis 16).

FUNCTION LWORD_TO_HEXSTR : T_MaxString

VAR_INPUT

```

VAR_INPUT
  in      : LWORD;
  iPrecision : INT;
  bLoCase : BOOL;
END_VAR

```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

bLoCase: Dieser Parameter bestimmt ob Klein- oder Großbuchstaben bei der Konvertierung benutzt werden sollen. FALSE => "ABCDEF", TRUE => "abcdef".

Beispiel:

```

PROGRAM MAIN
VAR
  s1 : STRING;
  s2 : STRING;
  s3 : STRING;
  s4 : STRING;
  nCnt : LWORD;
END_VAR

nCnt := 43981;
s1 := LWORD_TO_HEXSTR( nCnt, 1, FALSE );
s2 := LWORD_TO_HEXSTR( nCnt, 1, TRUE );
nCnt := 15;
s3 := LWORD_TO_HEXSTR( nCnt, 4, FALSE );
nCnt := 0;
s4 := LWORD_TO_HEXSTR( nCnt, 0, FALSE );

```

Das Ergebnis:

s1 = 'ABCD'

s2 = 'abcd'

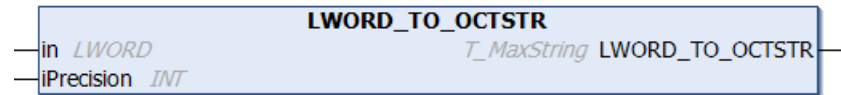
s3 = '000F'

s4 = ''

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.56 LWORD_TO_OCTSTR



Die Funktion konvertiert eine Dezimalzahl in einen Oktalstring (Basis 8).

FUNCTION LWORD_TO_OCTSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in      : LWORD;
    iPrecision : INT;
END_VAR
```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Beispiel:

```
PROGRAM MAIN
VAR
    s1 : STRING;
    s2 : STRING;
    s3 : STRING;
    nCnt : LWORD;
END_VAR

nCnt := 43981;
s1 := LWORD_TO_OCTSTR( nCnt, 1 );
s2 := LWORD_TO_OCTSTR( nCnt, 10 );
nCnt := 0;
s3 := LWORD_TO_OCTSTR( nCnt, 0 );
```

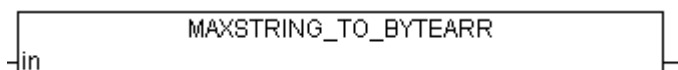
Das Ergebnis:

s1 = '125715'
s2 = '0000125715'
s3 = ''

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.57 MAXSTRING_TO_BYTEARR



Konvertiert einen String in einzelne ASCII-Codes eines Byte-Arrays.

FUNCTION MAXSTRING_TO_BYTEARR: ARRAY[0..MAX_STRING_LENGTH] OF BYTE

VAR_INPUT

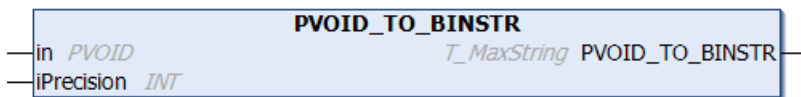
```
VAR_INPUT
  in : T_MaxString;
END_VAR
```

in: String der konvertiert werden soll (Typ: T_MaxString).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.58 PVOID_TO_BINSTR



Die Funktion konvertiert den Wert einer Pointervariablen vom Typ PVOID in einen Binärstring (Basis 2).

FUNCTION PVOID_TO_BINSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
  in          : PVOID;
  iPrecision  : INT;
END_VAR
```

in: Die zu konvertierende Pointervariable.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Beispiel:

```
PROGRAM MAIN
VAR
  s1   : STRING;
  s2   : STRING;
  s3   : STRING;
  s4   : STRING;
  s5   : STRING;
  s6   : STRING;
  nCnt : BYTE;
  pCnt : PVOID := 0;
END_VAR

pCnt := 0;
s1   := PVOID_TO_BINSTR( pCnt, 0 );
s2   := PVOID_TO_BINSTR( pCnt, 1 );
s3   := PVOID_TO_BINSTR( pCnt, 32 );

pCnt := ADR( nCnt );
s4   := PVOID_TO_BINSTR( pCnt, 0 );
s5   := PVOID_TO_BINSTR( pCnt, 1 );
s6   := PVOID_TO_BINSTR( pCnt, 32 );
```

Das Ergebnis:

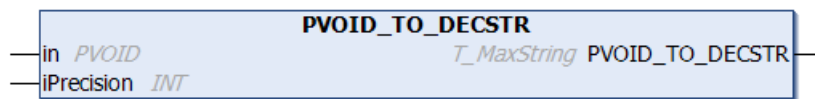
s1 = "

s2 = '0'
 s3 = '00000000000000000000000000000000'
 s4 = '10000111110111100000001001010101' (kann variieren)
 s5 = '10000111110111100000001001010101' (kann variieren)
 s6 = '10000111110111100000001001010101' (kann variieren)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.59 PVOID_TO_DECSTR



Die Funktion konvertiert den Wert einer Pointervariablen vom Typ PVOID in einen Dezimalstring (Basis 10).

FUNCTION PVOID_TO_DECSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in      : PVOID;
    iPrecision : INT;
END_VAR
```

in: Die zu konvertierende Pointervariable.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Beispiel:

```
PROGRAM MAIN
VAR
    s1 : STRING;
    s2 : STRING;
    s3 : STRING;
    s4 : STRING;
    s5 : STRING;
    s6 : STRING;
    nCnt : WORD;
    pCnt : PVOID := 0;
END_VAR

pCnt := 0;
s1 := PVOID_TO_DECSTR( pCnt, 0 );
s2 := PVOID_TO_DECSTR( pCnt, 1 );
s3 := PVOID_TO_DECSTR( pCnt, 16 );

pCnt := ADR( nCnt );
s4 := PVOID_TO_DECSTR( pCnt, 0 );
s5 := PVOID_TO_DECSTR( pCnt, 1 );
s6 := PVOID_TO_DECSTR( pCnt, 16 );
```

Das Ergebnis:

s1 = "
 s2 = '0'

s3 = '0000000000000000'

s4 = '2279473749' (kann variieren)

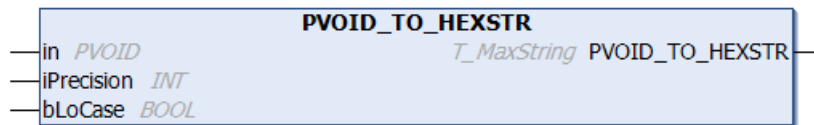
s5 = '2279473749' (kann variieren)

s6 = '0000002279473749' (kann variieren)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.60 PVOID_TO_HEXSTR



Die Funktion konvertiert den Wert einer Pointervariablen vom Typ PVOID in einen Hexadezimalstring (Basis 16).

FUNCTION PVOID_TO_HEXSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : PVOID;
    iPrecision  : INT;
    bLoCase    : BOOL;
END_VAR
```

in: Die zu konvertierende Pointervariable.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

bLoCase: Dieser Parameter bestimmt ob Klein- oder Großbuchstaben bei der Konvertierung benutzt werden sollen. FALSE => "ABCDEF", TRUE => "abcdef".

Beispiel:

```
PROGRAM MAIN
VAR
    s1   : STRING;
    s2   : STRING;
    s3   : STRING;
    s4   : STRING;
    s5   : STRING;
    s6   : STRING;
    s7   : STRING;
    s8   : STRING;
    s9   : STRING;
    s10  : STRING;
    s11  : STRING;
    s12  : STRING;
    nCnt : WORD;
    pCnt : PVOID := 0;
END_VAR

pCnt := 0;
s1   := PVOID_TO_HEXSTR( pCnt, 0, FALSE );
s2   := PVOID_TO_HEXSTR( pCnt, 0, TRUE  );

s3   := PVOID_TO_HEXSTR( pCnt, 1, FALSE );
s4   := PVOID_TO_HEXSTR( pCnt, 1, TRUE  );
```

```
s5 := PVOID_TO_HEXSTR( pCnt, 16, FALSE );
s6 := PVOID_TO_HEXSTR( pCnt, 16, TRUE );

pCnt := ADR( nCnt );
s7 := PVOID_TO_HEXSTR( pCnt, 0, FALSE );
s8 := PVOID_TO_HEXSTR( pCnt, 0, TRUE );

s9 := PVOID_TO_HEXSTR( pCnt, 1, FALSE );
s10 := PVOID_TO_HEXSTR( pCnt, 1, TRUE );

s11 := PVOID_TO_HEXSTR( pCnt, 16, FALSE );
s12 := PVOID_TO_HEXSTR( pCnt, 16, TRUE );
```

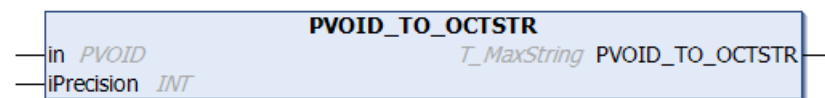
Das Ergebnis:

```
s1 = "
s2 = "
s3 = '0'
s4 = '0'
s5 = '0000000000000000'
s6 = '0000000000000000'
s7 = '87CBC255' (kann variieren)
s8 = '87cbc255' (kann variieren)
s9 = '87CBC255' (kann variieren)
s10 = '87cbc255' (kann variieren)
s11 = '000000087CBC255' (kann variieren)
s12 = '000000087cbc255' (kann variieren)
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.61 PVOID_TO_OCTSTR



Die Funktion konvertiert den Wert einer Pointervariablen vom Typ PVOID in einen Oktalstring (Basis 8).

FUNCTION PVOID_TO_OCTSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : PVOID;
    iPrecision  : INT;
END_VAR
```

in: Die zu konvertierende Pointervariable.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Beispiel:

```
PROGRAM MAIN
VAR
  s1   : STRING;
  s2   : STRING;
  s3   : STRING;
  s4   : STRING;
  s5   : STRING;
  s6   : STRING;
  nCnt : WORD;
  pCnt : PVOID := 0;
END_VAR

pCnt := 0;
s1   := PVOID_TO_OCTSTR( pCnt, 0 );
s2   := PVOID_TO_OCTSTR( pCnt, 1 );
s3   := PVOID_TO_OCTSTR( pCnt, 16 );

pCnt := ADR( nCnt );
s4   := PVOID_TO_OCTSTR( pCnt, 0 );
s5   := PVOID_TO_OCTSTR( pCnt, 1 );
s6   := PVOID_TO_OCTSTR( pCnt, 16 );
```

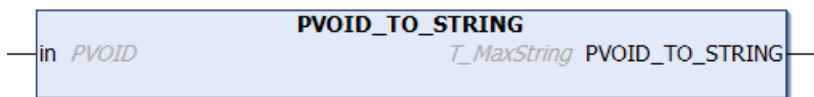
Das Ergebnis:

- s1 = "
- s2 = '0'
- s3 = '0000000000000000'
- s4 = '20767501125' (kann variieren)
- s5 = '20767501125' (kann variieren)
- s6 = '0000020767501125' (kann variieren)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.62 PVOID_TO_STRING



Die Funktion konvertiert den Wert einer Pointervariablen vom Typ PVOID in einen Hexadezimalstring (Basis 16). Der Hexadezimalstring besitzt den SPS-Prefix: '16#'. Die Auflösung ist fix und beträgt 8 Stellen auf einem 32 Bit System und 16 Stellen auf einem 64 Bit System.

FUNCTION PVOID_TO_STRING : T_MaxString

VAR_INPUT

```
VAR_INPUT
  in : PVOID;
END_VAR
```

in: Die zu konvertierende Pointervariable.

Beispiel:

```
PROGRAM MAIN
VAR
  s1 : STRING;
  s2 : STRING;
  nCnt : BYTE;
  p1 : POINTER TO BYTE := 0;
  p2 : POINTER TO BYTE := ADR( nCnt );
END_VAR

s1 := PVOID_TO_STRING( p1 );
s2 := PVOID_TO_STRING( p2 );
```

Das Ergebnis auf einem 32 bit System:

```
s1 = '16#00000000'
s2 = "16#87DE0255" (kann variieren)
```

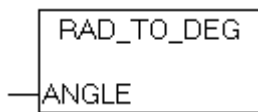
Das Ergebnis auf einem 64 Bit System:

```
s1 = '16#0000000000000000'
s2 = "16#8734651087DE0255" (kann variieren)
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.63 RAD_TO_DEG



Die Funktion konvertiert das Bogenmaß in einen Grad-Winkel.

FUNCTION RAD_TO_DEG : LREAL

VAR_INPUT

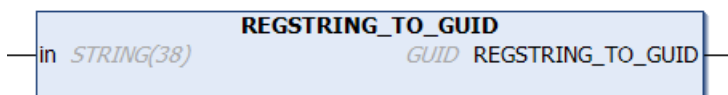
```
VAR_INPUT
  ANGLE : LREAL;
END_VAR
```

ANGLE: Das zu konvertierende Bogenmaß.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.64 REGSTRING_TO_GUID



Diese Funktion konvertiert eine Registry-GUID-Stringvariable (eingeschlossen in geschweifte Klammern) in eine strukturierte [GUID \[► 309\]](#)-Variable.

FUNCTION REGSTRING_TO_GUID : GUID

VAR_INPUT

```
VAR_INPUT
    in : STRING(38);
END_VAR
```

Rückgabewert	Bedeutung
'{xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}'	Kein Fehler (,x' ist ein Hexadezimal-Halbbyte).
'{00000000-0000-0000-0000-000000000000}'	Konvertierung fehlgeschlagen oder GUID hat den Initialwert (alle Bytes sind Null)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.65 ROUTETRANSPORT_TO_STRING



Die Funktion konvertiert die AMS Message Router Transportschichtkennung in einen String.

FUNCTION ROUTETRANSPORT_TO_STRING : STRING

VAR_INPUT

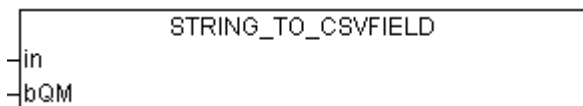
```
VAR_INPUT
    eType : E_RouteTransportType;
END_VAR
```

eType: Die zu konvertierende Transportschichtkennung (Typ: [E_RouteTransportType](#) [[▶ 307](#)]).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.66 STRING_TO_CSVFIELD



Die Funktion konvertiert den Wert einer SPS-Stringvariablen in ein Datenfeld im CSV-Format als String. Einfache Einführungszeichen im Quellstring werden durch doppelte Einführungszeichen ersetzt. Beim gesetzten bQM-Parameter (QM = quotation marks) werden auch die äußeren (das CSV-Datenfeld einschließenden) Einführungszeichen hinzugefügt. Beim Erfolg liefert die Funktion den konvertierten String als Ergebnis zurück. Die Funktion liefert einen Leerstring wenn bei der Konvertierung ein Fehler aufgetreten ist aber nur dann wenn der Quellstring auch kein Leerstring war.

Die Funktion wird normalerweise zusammen mit dem Funktionsbaustein [FB_CSVMemBufferWriter](#) [[▶ 50](#)] verwendet um Datensätze im SPS-Speicher im CSV-Format zu erzeugen. Im nächsten Schritt kann der Speicherinhalt dann in die Datei geschrieben werden.

Der Quellstring darf keine Binärdaten enthalten. Binärdaten mit dem Wert Null würden den String an der falschen Stelle terminieren und abschneiden! Um Binärdaten zu konvertieren verwenden Sie bitte die Funktion: [ARG_TO_CSVFIELD \[► 247\]](#).

FUNCTION STRING_TO_CSVFIELD : T_MaxString

VAR_INPUT

```
VAR_INPUT
  in : T_MaxString;
  bQM : BOOL;
END_VAR
```

in: Quellstring dessen Wert in ein Datenfeld im CSV-Format konvertiert werden soll (Typ: T_MaxString).

bQM: Bei TRUE an diesem Eingang werden dem Ergebnisstring auch die einschließenden Einführungszeichen hinzugefügt.

bQM	Beschreibung	Quellstring	Ergebnisstring	CSV-Konform
FALSE	Quellstring, der nicht mit den einschließenden Einführungszeichen versehen wird, sollte möglichst nur aus Buchstaben und Zahlen bestehen. In diesem Fall darf der Quellstring keine nicht-druckbaren Steuerzeichen, Einführungszeichen, Semikolon, Komma (US-CSV-Format) oder Binärdaten enthalten.	'Module_XA5'	'Module_XA5'	Ja
		'123456'	'123456'	Ja
		"	"	Ja
		'A"'B'	'A""""B'	Nein
		'A"B'	'A"'B'	Nein
		','	','	Nein
		'\$R\$N'	'\$R\$N'	Nein
		'AB\$00CD'	'AB' (String wurde abgeschnitten)	Nein
TRUE	Quellstring, der mit den einschließenden Einführungszeichen versehen wird, darf auch die nicht-druckbaren Steuerzeichen, Einführungszeichen, Semikolon oder Komma (US-CSV-Format) enthalten. Binärdaten sind nicht zulässig.	'Module_XA5'	""Module_XA5""	Ja
		'123456'	""123456""	Ja
		"	""""	Ja
		'A"'B'	""A""""B""	Ja
		'A"B'	""A"'B""	Ja
		','	""','""	Ja
		'\$R\$N'	""\$R\$N""	Ja
		'AB\$00CD'	""AB"" (String wurde abgeschnitten)	Nein

Beispiel:

```
PROGRAM MAIN
VAR
  s1 : STRING;
  s2 : STRING;
END_VAR

s1 := STRING_TO_CSVFIELD( 'Module_"ALFA_$05"_6', TRUE );
s2 := STRING_TO_CSVFIELD( 'Module_50', FALSE );
```

Das Ergebnis:

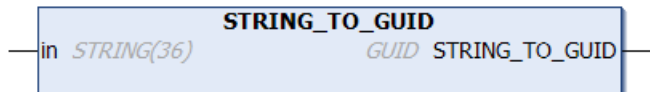
```
s1 = ""Module_""ALFA_$05""_6""
s2 = 'Module_50'
```

Weitere Informationen finden Sie hier: [Beispiel: Schreiben/lesen einer CSV-Datei \[► 351\]](#).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.67 STRING_TO_GUID



Diese Funktion konvertiert eine GUID-Stringvariable (ohne geschweifte Klammern) in eine Strukturierte GUID [► 309]-Variable.

FUNCTION STRING_TO_GUID : GUID

VAR_INPUT

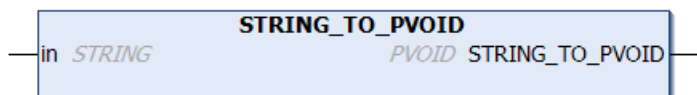
```
VAR_INPUT
    in : STRING(36);
END_VAR
```

Rückgabewert	Bedeutung
'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'	Kein Fehler (,x' ist ein Hexadezimal-Halbbyte)
'00000000-0000-0000-0000-000000000000'	Konvertierung fehlgeschlagen oder GUID hat den Initialwert (alle Bytes sind Null)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.68 STRING_TO_PVOID



Die Funktion konvertiert eine Stringvariable in eine Pointervariable vom Typ PVOID. Die Funktion liefert den Rückgabewert Null, wenn der Eingangsstring fehlerhafte Zeichen enthält und nicht als Adresse interpretiert werden kann.

FUNCTION STRING_TO_PVOID : PVOID

VAR_INPUT

```
VAR_INPUT
    in : STRING;
END_VAR
```

in: Die zu konvertierende Stringvariable.

Beispiel:

```
PROGRAM MAIN
VAR
    sP1 : STRING := '16#89345678';
    sP2 : STRING := '8#21115053170';
    sP3 : STRING := '2#10001001001101000101011001111000';
    sP4 : STRING := '2301908600';
    sP5 : STRING := '';
    pP1 : PVOID := 0;
```

```

pP2 : PVOID := 0;
pP3 : PVOID := 0;
pP4 : PVOID := 0;
pP5 : PVOID := 0;
END_VAR

pP1 := STRING_TO_PVOID( sP1 );
pP2 := STRING_TO_PVOID( sP2 );
pP3 := STRING_TO_PVOID( sP3 );
pP4 := STRING_TO_PVOID( sP4 );
pP5 := STRING_TO_PVOID( sP5 );

```

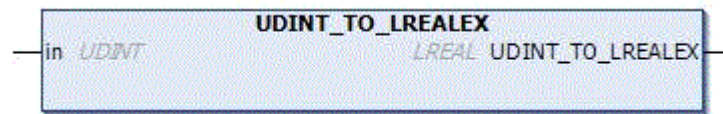
Das Ergebnis:

pP1 = 2301908600
pP2 = 2301908600
pP3 = 2301908600
pP4 = 2301908600
pP5 = 0

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.69 UDINT_TO_LREALX



In TwinCAT 2 auf der ARM-Plattform wird die Konvertierung der vorzeichenlosen Zahlen in Fließkommazahlen von Typ: LREAL nicht unterstützt. Vorzeichenlose Zahlen mit einem gesetztem höchstwertigen Bit werden möglicherweise implizit in negative Fließkommazahlen konvertiert. Die hier beschriebene Funktion erlaubt in TwinCAT 2 eine explizite Konvertierung vom Typ UDINT in eine positive Fließkommazahl vom Typ LREAL (auch wenn das höchstwertige Bit gesetzt wurde und ohne Compiler-Warnung). Diese Funktion benötigen Sie nur um konvertierte TwinCAT 2 Projekte ohne Änderungen in TwinCAT 3 übersetzen zu können.

In TwinCAT 3 werden vorzeichenlose Zahlen von Typ UDINT immer (implizit und explizit) in eine positive Fließkommazahl Zahl konvertiert. Aus diesem Grund kann auf die Verwendung dieser Funktion verzichtet werden.

FUNCTION UDINT_TO_LREALX : LREAL

VAR_INPUT

```

VAR_INPUT
  in : UDINT;
END_VAR

```

Beispiel:

```

PROGRAM MAIN
VAR
  nUdint : UDINT := 16#FFFFFFFF;
  fLreal : LREAL := 0.0;
END_VAR

```

fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := nUdint	-*1, Warning 1105**	4294967295	4294967295

fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := UDINT_TO_LREAL(nUdint)	-*1, Warning 1105**	4294967295	4294967295
fLreal := UDINT#16#FFFFFFFF	-*1, Warning 1105**	4294967295	4294967295
fLreal := 16#FFFFFFF	4294967295	4294967295	4294967295
fLreal := UDINT_TO_LREALEX(nUdint)	4294967295	4294967295	4294967295
fLreal := UDINT_TO_LREALEX(UDINT#16#FFFFFFFF)	4294967295	4294967295	4294967295
fLreal := UDINT_TO_LREALEX(16#FFFFFFF)	4294967295	4294967295	4294967295

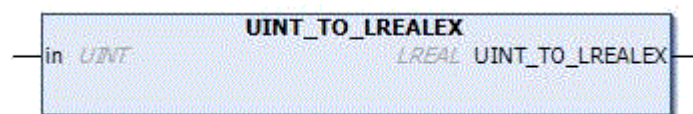
*not supported

**Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.70 UINT_TO_LREALEX



In TwinCAT 2 auf der ARM-Plattform wird die Konvertierung der vorzeichenlosen Zahlen in Fließkommazahlen von Typ: LREAL nicht unterstützt. Vorzeichenlose Zahlen mit einem gesetztem höchstwertigen Bit werden möglicherweise implizit in negative Fließkommazahlen konvertiert. Die hier beschriebene Funktion erlaubt in TwinCAT 2 eine explizite Konvertierung vom Typ UINT in eine positive Fließkommazahl vom Typ LREAL (auch wenn das höchstwertige Bit gesetzt wurde und ohne Compiler-Warnung). Diese Funktion benötigen Sie nur um konvertierte TwinCAT 2 Projekte ohne Änderungen in TwinCAT 3 übersetzen zu können.

In TwinCAT 3 werden vorzeichenlose Zahlen von Typ UINT immer (implizit und explizit) in eine positive Fließkommazahl Zahl konvertiert. Aus diesem Grund kann auf die Verwendung dieser Funktion verzichtet werden.

FUNCTION UINT_TO_LREALEX : LREAL

VAR_INPUT

```
VAR_INPUT
  in : UINT;
END_VAR
```

Beispiel:

```
PROGRAM MAIN
VAR
  nUdint : UINT := 16#FFFF;
  fLreal : LREAL := 0.0;
END_VAR
```

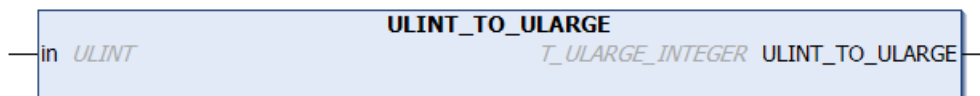
fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := nUInt	+65535, Warning 1105*	+65535	+65535
fLreal := UINT_TO_LREAL(nUInt)	+65535, Warning 1105*	+65535	+65535
fLreal := UINT#16#FFFF	+65535, Warning 1105*	+65535	+65535
fLreal := 16#FFFF	+65535	+65535	+65535
fLreal := UINT_TO_LREALEX(nUInt)	+65535	+65535	+65535
fLreal := UINT_TO_LREALEX(UINT#16#FFFF)	+65535	+65535	+65535
fLreal := UINT_TO_LREALEX(16#FFFF)	+65535	+65535	+65535

*Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.71 ULINT_TO_ULARGE



Die Funktion konvertiert eine TwinCAT 3 vorzeichenlose 64 bit Zahl („native“-Typ) in eine TwinCAT 2 vorzeichenlose 64 bit Zahl („legacy“-Typ: T_ULARGE_INTEGER [► 327]).

FUNCTION ULINT_TO_ULARGE : T_ULARGE_INTEGER

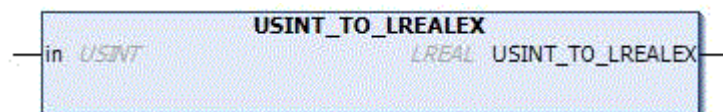
VAR_INPUT

```
VAR_INPUT
  in : ULINT;
END_VAR
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.72 USINT_TO_LREALEX



In TwinCAT 2 auf der ARM-Plattform wird die Konvertierung der vorzeichenlosen Zahlen in Fließkommazahlen von Typ: LREAL nicht unterstützt. Vorzeichenlose Zahlen mit einem gesetztem höchstwertigen Bit werden möglicherweise implizit in negative Fließkommazahlen konvertiert. Die hier beschriebene Funktion erlaubt in TwinCAT 2 eine explizite Konvertierung vom Typ USINT in eine positive

Fließkommazahl vom Typ LREAL (auch wenn das höchstwertige Bit gesetzt wurde und ohne Compiler-Warnung). Diese Funktion benötigen Sie nur um konvertierte TwinCAT 2 Projekte ohne Änderungen in TwinCAT 3 übersetzen zu können.

In TwinCAT 3 werden vorzeichenlose Zahlen von Typ USINT immer (implizit und explizit) in eine positive Fließkommazahl Zahl konvertiert. Aus diesem Grund kann auf die Verwendung dieser Funktion verzichtet werden.

FUNCTION USINT_TO_LREALEX : LREAL

VAR_INPUT

```
VAR_INPUT
  in : USINT;
END_VAR
```

Beispiel:

```
PROGRAM MAIN
VAR
  nUsint : USINT := 16#FF;
  fLreal : LREAL := 0.0;
END_VAR
```

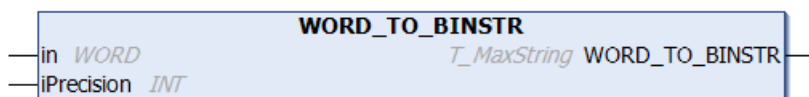
fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := nUsint	+255, Warning 1105*	+255	+255
fLreal := USINT_TO_LREAL(nUsint)	+255, Warning 1105*	+255	+255
fLreal := USINT#16#FF	+255, Warning 1105*	+255	+255
fLreal := 16#FF	+255	+255	+255
fLreal := USINT_TO_LREALEX(nUsint)	+255	+255	+255
fLreal := USINT_TO_LREALEX(USINT#16#FF)	+255	+255	+255
fLreal := USINT_TO_LREALEX(16#FF)	+255	+255	+255

*Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.73 WORD_TO_BINSTR



Die Funktion konvertiert eine Dezimalzahl in einen Binärstring (Basis 2).

FUNCTION WORD_TO_BINSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
  in : WORD;
  iPrecision : INT;
END_VAR
```

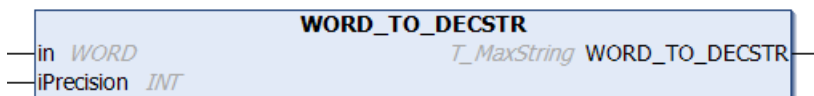
in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.74 WORD_TO_DECSTR



Die Funktion konvertiert eine Dezimalzahl in einen Dezimalstring (Basis 10).

FUNCTION WORD_TO_DECSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in      : WORD;
    iPrecision : INT;
END_VAR
```

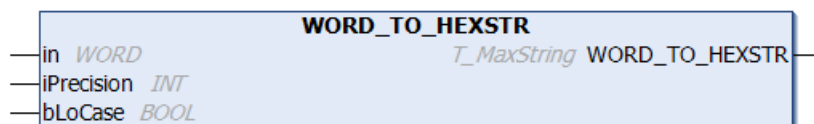
in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

4.75 WORD_TO_HEXSTR



Die Funktion konvertiert eine Dezimalzahl in einen Hexadezimalstring (Basis 16).

FUNCTION WORD_TO_HEXSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in      : WORD;
    iPrecision : INT;
    bLoCase  : BOOL := FALSE;
END_VAR
```


in: Die zu konvertierende Dezimalzahl.

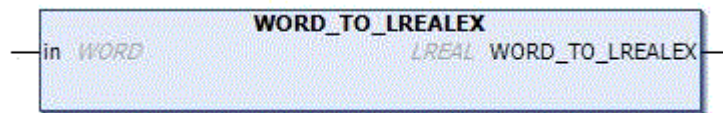
iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

bLoCase: Dieser Parameter bestimmt ob Klein- oder Großbuchstaben bei der Konvertierung benutzt werden sollen. FALSE => "ABCDEF", TRUE => "abcdef".

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.76 WORD_TO_LREAL



In TwinCAT 2 auf der ARM-Plattform wird die Konvertierung der vorzeichenlosen Zahlen in Fließkommazahlen von Typ: LREAL nicht unterstützt. Vorzeichenlose Zahlen mit einem gesetzten höchstwertigen Bit werden möglicherweise implizit in negative Fließkommazahlen konvertiert. Die hier beschriebene Funktion erlaubt in TwinCAT 2 eine explizite Konvertierung vom Typ WORD in eine positive Fließkommazahl vom Typ LREAL (auch wenn das höchstwertige Bit gesetzt wurde und ohne Compiler-Warnung). Diese Funktion benötigen Sie nur um konvertierte TwinCAT 2 Projekte ohne Änderungen in TwinCAT 3 übersetzen zu können.

In TwinCAT 3 werden vorzeichenlose Zahlen von Typ WORD immer (implizit und explizit) in eine positive Fließkommazahl Zahl konvertiert. Aus diesem Grund kann auf die Verwendung dieser Funktion verzichtet werden.

FUNCTION WORD_TO_LREAL : LREAL

VAR_INPUT

```
VAR_INPUT
    in : WORD;
END_VAR
```

Beispiel:

```
PROGRAM MAIN
VAR
    nWord : WORD := 16#FFFF;
    fLreal : LREAL := 0.0;
END_VAR
```

fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := nWord	+65535, Warning 110 5*	+65535	+65535
fLreal := WORD_TO_LREAL(nWord)	+65535, Warning 110 5*	+65535	+65535
fLreal := WORD#16#FFFF	+65535, Warning 110 5*	+65535	+65535
fLreal := 16#FFFF	+65535	+65535	+65535
fLreal := WORD_TO_LREAL(nWord)	+65535	+65535	+65535

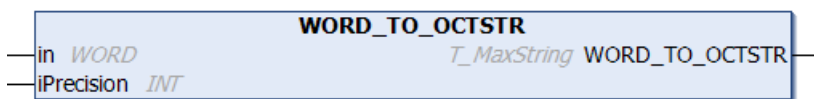
fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := WORD_TO_L REALEX(WORD#16#FFF F)	+65535	+65535	+65535
fLreal := WORD_TO_L REALEX(16#FFFF)	+65535	+65535	+65535

*Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

4.77 WORD_TO_OCTSTR



Die Funktion konvertiert eine Dezimalzahl in einen Oktalstring (Basis 8).

FUNCTION WORD_TO_OCTSTR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : WORD;
    iPrecision : INT;
END_VAR
```

in: Die zu konvertierende Dezimalzahl.

iPrecision: Minimale Anzahl der erscheinenden Stellen (digits). Ist die tatsächliche Anzahl der signifikanten Stellen kleiner als der *iPrecision*-Parameter, dann wird der resultierende String links mit Nullen aufgefüllt. Ist die Anzahl der signifikanten Stellen größer als der *iPrecision*-Parameter, dann wird der resultierende String nicht abgeschnitten! Hat der *iPrecision*-Parameter und der *in*-Parameter den Wert Null, dann ist der resultierende String ein Leerstring.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5 Datentypen

5.1 ADSDATATYPEID

ADS Datentypbezeichnung. Dieser Datentyp wird z.B. von dem Funktionsbaustein [FB_ReadSymInfoByNameEx \[► 132\]](#) verwendet.

```

TYPE ADSDATATYPEID :
(
  ADST_VOID      := 0,
  ADST_INT8      := 16,
  ADST_UINT8     := 17,
  ADST_INT16     := 2,
  ADST_UINT16    := 18,
  ADST_INT32     := 3,
  ADST_UINT32    := 19,
  ADST_INT64     := 20,
  ADST_UINT64    := 21,
  ADST_REAL32    := 4,
  ADST_REAL64    := 5,
  ADST_BIGTYPE   := 65,
  ADST_STRING    := 30,
  ADST_WSTRING   := 31,
  ADST_REAL80    := 32,
  ADST_BIT       := 33,
  ADST_MAXTYPES
);
END_TYPE

```

Wert	Bedeutung
ADST_VOID	Reserviert
ADST_INT8	Signed 8 bit integer
ADST_UINT8	Unsigned 8 bit integer
ADST_INT16	Signed 16 bit integer
ADST_UINT16	Unsigned 16 bit integer
ADST_INT32	Signed 32 bit integer
ADST_UINT32	Unsigned 32 bit integer
ADST_INT64	Signed 64 bit integer
ADST_UINT64	Unsigned 64 bit integer
ADST_REAL32	32 bit floating point number
ADST_REAL64	64 bit floating point number
ADST_BIGTYPE	Structured type
ADST_STRING	String type
ADST_WSTRING	Wide character type
ADST_REAL80	Reserviert
ADST_BIT	Bit type
ADST_MAXTYPES	Max. available type

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.2 E_AmsLoggerMode

AMS-Logger Steuermodi. Dieser Datentyp wird von dem Funktionsbaustein: [FB_AmsLogger \[► 43\]](#) verwendet.

```

TYPE E_AmsLoggerMode :
(
    AMSLOGGER_RUN      := 1,
    AMSLOGGER_STOP    := 2
);
END_TYPE
    
```

Wert	Bedeutung
AMSLOGGER_RUN	Startet den AMS Logger
AMSLOGGER_STOP	Stoppt den AMS Logger

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.3 E_ArgType

Interne Argument-Typbezeichnung. Dieser Typ wird von String-Format Funktionen/Funktionsbausteinen verwendet.

```

TYPE E_ArgType :
(
    ARGTYPE_UNKNOWN    := 0,
    ARGTYPE_BYTE,
    ARGTYPE_WORD,
    ARGTYPE_DWORD,
    ARGTYPE_REAL,
    ARGTYPE_LREAL,
    ARGTYPE_SINT,
    ARGTYPE_INT,
    ARGTYPE_DINT,
    ARGTYPE_USINT,
    ARGTYPE_UINT,
    ARGTYPE_UDINT,
    ARGTYPE_STRING,
    ARGTYPE_BOOL,
    ARGTYPE_BIGTYPE,
    ARGTYPE_ULONG,
    ARGTYPE_UHUGE,
    ARGTYPE_LARGE,
    ARGTYPE_HUGE,
    ARGTYPE_LWORD
);
END_TYPE
    
```

Wert	Entsprechender SPS-Datentyp
ARGTYPE_UNKNOWN	Typ ist unbekannt oder nicht initialisiert
ARGTYPE_BYTE	BYTE (8 bits)
ARGTYPE_WORD	WORD (16 bits)
ARGTYPE_DWORD	DWORD (32 bits)
ARGTYPE_REAL	REAL
ARGTYPE_LREAL	LREAL
ARGTYPE_SINT	SINT
ARGTYPE_INT	INT
ARGTYPE_DINT	DINT
ARGTYPE_USINT	USINT
ARGTYPE_UINT	UINT
ARGTYPE_UDINT	UDINT
ARGTYPE_STRING	String vom Typ: T_MaxString
ARGTYPE_BOOL	BOOL
ARGTYPE_BIGTYPE	Beliebige Datenstruktur oder Bytepuffer

Wert	Entsprechender SPS-Datentyp
ARGTYPE_ULARGE	T_ULARGE_INTEGER oder ULINT (unsigned 64 bit integer)
ARGTYPE_UHUGE	T_UHUGE_INTEGER (unsigned 128 bit integer)
ARGTYPE_LARGE	T_LARGE_INTEGER oder LINT (signed 64 bit integer)
ARGTYPE_HUGE	T_HUGE_INTEGER (signed 128 bit integer)
ARGTYPE_LWORD	LWORD (64 bits)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.4 E_DbgContext

Dieser Variablentyp kann von Protokollbausteinen verwendet werden und bestimmt den Kontext der Debugausgabe.

```

TYPE E_DbgContext :
(
  eDbgContext_NONE := 0, (* Not used *)
  eDbgContext_USER := 1, (* Service user *)
  eDbgContext_PROV := 2 (* Service provider *)
);
END_TYPE

```

Wert	Bedeutung
eDbgContext_NONE	Parameter wird nicht benutzt
eDbgContext_USER	Die Debugausgabe wurde ausgelöst durch den Service-User
eDbgContext_PROV	Die Debugausgabe wurde ausgelöst durch den Service-Provider

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.5 E_DbgDirection

Dieser Variablentyp kann von Pufferbausteinen oder Protokollbausteinen zur Konfiguration der Debugausgabe verwendet werden.

```

TYPE E_DbgDirection :
(
  eDbgDirection_OFF := 0, (* Disabled (no debug oputput) *)
  eDbgDirection_IN := 1, (* Enabled only for incoming data *)
  eDbgDirection_OUT := 2, (* Enabled only for outgoing data *)
  eDbgDirection_ALL := 3 (* Enabled for incoming and outgoing data *)
);
END_TYPE

```

Wert	Bedeutung
eDbgDirection_OFF	Deaktiviert die Debugausgabe
eDbgDirection_IN	Aktiviert die Ausgabe der eingehenden Telegramme
eDbgDirection_OUT	Aktiviert die Ausgabe der ausgehenden Telegramme
eDbgDirection_ALL	Aktiviert die Ausgabe der eingehenden und ausgehenden Telegramme

Beispiel:

Die Debugausgabe selbst kann z.B. mit Hilfe der ADSLOGSTR-Funktion realisiert werden.

In einem Ringpuffer könnte z.B. die Debugausgabe über die Variable auf folgende Weise gesteuert werden:

- Beim Wert `eDbgDirection_IN` oder `eDbgDirection_ALL` erfolgt die Debugausgabe wenn dem Puffer ein neuer Wert hinzugefügt wird;
- Beim Wert `eDbgDirection_OUT` oder `eDbgDirection_ALL` erfolgt die Debugausgabe wenn ein Wert aus dem Puffer entfernt wird;

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.6 E_EnumCmdType

Steuerparameter für Aufzählungsbausteine. Nicht alle Parameter werden von jedem Aufzählungsbaustein benutzt!

```

TYPE E_EnumCmdType :
(
    eEnumCmd_First := 0,
    eEnumCmd_Next,
    eEnumCmd_Abort
);
END_TYPE
    
```

Wert	Bedeutung
<code>eEnumCmd_First</code>	Listet das erste Element auf
<code>eEnumCmd_Next</code>	Listet das nächste Element auf
<code>eEnumCmd_Abort</code>	Bricht die Auflistung ab (schließt geöffnete handles)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.7 E_HashMode

```

{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_HashMode :
(
    HASH_MD5 := 1,           // generates 16 bytes hash value
    HASH_SHA1,              // generates 20 bytes hash value
    HASH_SHA256,            // generates 32 bytes hash value
    HASH_SHA384,            // generates 48 bytes hash value
    HASH_SHA512             // generates 64 bytes hash value
) DINT;
END_TYPE
    
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024.29	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= 3.3.51.0

5.8 E_LDevType

```

TYPE E_LDevType :
(
  eLDT_Unknown      : UDINT,
  eLDT_Beckhoff     : UDINT,
  eLDT_GenericPC    : UDINT,
  eLDT_TerminalDongle : UDINT,
  eLDT_UsbDongle    : UDINT
);
END_TYPE

```

Wert	Bedeutung
eLDT_Unknown	Gerätetyp ist unbekannt.
eLDT_Beckhoff	Beckhoff-Gerät (z. B. IPC, CX, ...)
eLDT_GenericPC	Fremd-PC
eLDT_TerminalDongle	License-Key-Klemme, z. B. EL6070
eLDT_UsbDongle	License-Key-USB-Stick, z. B. C9900-L100

5.9 E_LDongleStatus

```

TYPE E_LDongleStatus :
(
  eLDT_Unknown      : UDINT,
  eLDT_OK           : UDINT,
  eLDT_Pending      : UDINT,
  eLDT_Invalid      : UDINT,
  eLDT_NoConnection : UDINT
);
END_TYPE

```

Wert	Bedeutung
eLDT_Unknown	Lizenz-Dongle-Status unbekannt
eLDT_OK	Lizenz-Dongle wurde erfolgreich validiert
eLDT_Pending	Validierung des Lizenz-Dongle läuft
eLDT_Invalid	Lizenz-Dongle ist ungültig
eLDT_NoConnection	Keine Verbindung zum Lizenz-Dongle

5.10 E_LicenseHResult

```

TYPE E_LicenseHResult :
(
  //success
  E_LHR_LicenseOK      : DINT := 0,
  E_LHR_LicenseOK_Pending : DINT := 16#203,
  E_LHR_LicenseOK_Demo  : DINT := 16#254,
  E_LHR_LicenseOK_OEM   : DINT := 16#255,
  //error
  E_LHR_LicenseNotFound : DINT := DWORD_TO_DINT(16#98110700+16#24),
  E_LHR_LicenseExpired  : DINT := DWORD_TO_DINT(16#98110700+16#25),
  E_LHR_LicenseExceeded : DINT := DWORD_TO_DINT(16#98110700+16#26),
  E_LHR_LicenseInvalid  : DINT := DWORD_TO_DINT(16#98110700+16#27),
  E_LHR_LicenseSystemIdInvalid : DINT := DWORD_TO_DINT(16#98110700+16#28),
  E_LHR_LicenseNoTimeLimit : DINT := DWORD_TO_DINT(16#98110700+16#29),
  E_LHR_LicenseTimeInFuture : DINT := DWORD_TO_DINT(16#98110700+16#2A),
  E_LHR_LicenseTimePeriodToLong : DINT := DWORD_TO_DINT(16#98110700+16#2B),
  E_LHR_DeviceException : DINT := DWORD_TO_DINT(16#98110700+16#2C),
  E_LHR_LicenseDuplicated : DINT := DWORD_TO_DINT(16#98110700+16#2D),
  E_LHR_SignatureInvalid : DINT := DWORD_TO_DINT(16#98110700+16#2E),
  E_LHR_CertificateInvalid : DINT := DWORD_TO_DINT(16#98110700+16#2F),
  E_LHR_LicenseOemNotFound : DINT := DWORD_TO_DINT(16#98110700+16#30),
  E_LHR_LicenseRestricted : DINT := DWORD_TO_DINT(16#98110700+16#31),
  E_LHR_LicenseDemoDenied : DINT := DWORD_TO_DINT(16#98110700+16#32),
  E_LHR_LicensePlatformLevelInv : DINT := DWORD_TO_DINT(16#98110700+16#33)
) DINT;
END_TYPE

```

Wert	Bedeutung
E_LHR_LicenseOK	Lizenz ist gültig
E_LHR_LicenseOK_Pending	Validierung des Lizenzierungsgeräts (z. B. Lizenzklemme) notwendig
E_LHR_LicenseOK_Demo	Testlizenz ist gültig
E_LHR_LicenseOK_OEM	OEM-Lizenz ist gültig
E_LHR_LicenseNotFound	Fehlende Lizenz
E_LHR_LicenseExpired	Lizenz abgelaufen
E_LHR_LicenseExceeded	Lizenz hat zu wenig Instanzen
E_LHR_LicenseInvalid	Lizenz ist ungültig
E_LHR_LicenseSystemIdInvalid	Falsche System-ID für die Lizenz
E_LHR_LicenseNoTimeLimit	Lizenz nicht zeitlich begrenzt
E_LHR_LicenseTimeInFuture	Lizenzproblem: Ausstellungszeitpunkt in der Zukunft
E_LHR_LicenseTimePeriodTooLong	Lizenz-Zeitraum zu lang
E_LHR_DeviceException	Exception beim Systemstart
E_LHR_LicenseDuplicated	Lizenzdaten mehrfach gelesen
E_LHR_SignatureInvalid	Ungültige Signatur
E_LHR_CertificateInvalid	Ungültiges Zertifikat
E_LHR_LicenseOemNotFound	OEM-Lizenz für unbekanntes OEM
E_LHR_LicenseRestricted	Lizenz für das System ungültig
E_LHR_LicenseDemoDenied	Testlizenz nicht erlaubt
E_LHR_LicensePlatformLevelInv	Ungültiger Plattform-Level für die Lizenz

5.11 E_MIB_IF_Type

Management Information Base Interface Typ.

```
TYPE E_MIB_IF_Type :
```

```
(
  MIB_IF_TYPE_OTHER      := 1,
  MIB_IF_TYPE_ETHERNET  := 6,
  MIB_IF_TYPE_TOKENRING := 9,
  MIB_IF_TYPE_FDDI      := 15,
  MIB_IF_TYPE_PPP       := 23,
  MIB_IF_TYPE_LOOPBACK  := 24,
  MIB_IF_TYPE_SLIP      := 28
);
END_TYPE
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.12 E_NumGroupTypes

Numerische Zahlengruppen. Dieser Datentyp wird z.B. von dem Funktionsbaustein: FB_EnumStringNumbers [► 56] verwendet.

```
TYPE E_NumGroupTypes :
```

```
(
  eNumGroup_Float,
  eNumGroup_Unsigned,
  eNumGroup_Signed
);
END_TYPE
```

Wert	Bedeutung
eNumGroup_Float	Fließkommazahlen

Wert	Bedeutung
eNumGroup_Unsigned	Vorzeichenlose Zahlen
eNumGroup_Signed	Vorzeichenbehaftete Zahlen

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.13 E_PersistentMode

Modus in dem die persistenten Daten geschrieben werden sollen. Dieser Datentyp wird von dem Funktionsbaustein: [FB_WritePersistentData](#) [► 119] verwendet.

```
TYPE E_PersistentMode :
(
  SPDM_2PASS      := 0,
  SPDM_VAR_BOOST := 1
);
END_TYPE
```

Wert	Bedeutung
SPDM_2PASS	Alle Daten sind aus einem Zyklus
SPDM_VAR_BOOST	Daten einzelner persistenten Variablen sind aus einem Zyklus

Beispiel:

Siehe auch: [Schreiben der pers. Daten: Systemverhalten](#) [► 356].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.14 E_RegValueType

Typbezeichnungen für Registrywerte.

```
TYPE E_RegValueType :
(
  REG_NONE := 0,
  REG_SZ,
  REG_EXPAND_SZ,
  REG_BINARY,
  REG_DWORD,
  REG_DWORD_BIG_ENDIAN,
  REG_LINK,
  REG_MULTI_SZ,
  REG_RESOURCE_LIST,
  REG_FULL_RESOURCE_DESCRIPTOR,
  REG_RESOURCE_REQUIREMENTS_LIST,
  REG_QWORD
);
END_TYPE
```

Wert	Bedeutung
REG_NONE	No value TYPE
REG_SZ	Unicode null terminated STRING
REG_EXPAND_SZ	Unicode null terminated STRING (with environment variable references)

Wert	Bedeutung
REG_BINARY	Free form binary
REG_DWORD	32 bit number and REG_DWORD_LITTLE_ENDIAN (same as REG_DWORD)
REG_DWORD_BIG_ENDIAN	32 bit number
REG_LINK	Symbolic Link (unicode)
REG_MULTI_SZ	Multiple Unicode strings
REG_RESOURCE_LIST	Resource list in the resource map
REG_FULL_RESOURCE_DESCRIPTOR	Resource list in the hardware description
REG_RESOURCE_REQUIREMENTS_LIST	-
REG_QWORD	64 bit number and REG_QWORD_LITTLE_ENDIAN (same as REG_QWORD)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.15 E_RouteTransportType

Die Transportschicht mit der AMS Nachrichten befördert werden. Zur Zeit wird nur TCP/IP als Transportschicht unterstützt.

```

TYPE E_RouteTransportType :
(
    eRouteTransport_None           := 0,
    eRouteTransport_TCP_IP        := 1,
    eRouteTransport_IIO_LIGHTBUS  := 2,
    eRouteTransport_PROFIBUS_DP   := 3,
    eRouteTransport_PCI_ISA_BUS   := 4,
    eRouteTransport_ADS_UDP       := 5,
    eRouteTransport_FATP_UDP      := 6,
    eRouteTransport_COM_PORT      := 7,
    eRouteTransport_USB           := 8,
    eRouteTransport_CAN_OPEN     := 9,
    eRouteTransport_DEVICE_NET    := 10,
    eRouteTransport_SSB          := 11,
    eRouteTransport_SOAP         := 12
);
END_TYPE
    
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.16 E_SBCSType

Windows SBCS (Single Byte Character Set) code page types.

```

TYPE E_SBCSType :
(
    eSBCS_WesternEuropean := 1,
    eSBCS_CentralEuropean := 2
);
END_TYPE
    
```

Wert	Bedeutung
eSBCS_WesternEuropean	Windows 1252 (default) code page
eSBCS_CentralEuropean	Windows 1251 code page

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.17 E_ScopeServerState

```

TYPE E_ScopeServerState
(
  SCOPE_SERVER_IDLE,
  SCOPE_SERVER_CONNECT,
  SCOPE_SERVER_START,
  SCOPE_SERVER_STOP,
  SCOPE_SERVER_SAVE,
  SCOPE_SERVER_DISCONNECT,
  SCOPE_SERVER_RESET
);

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.18 E_TimeZoneID

Zusätzliche Informationen zur konfigurierten Zeitzone des Betriebssystems.

```

TYPE E_TimeZoneID :
(
  eTimeZoneID_Invalid := -1,
  eTimeZoneID_Unknown := 0,
  eTimeZoneID_Standard := 1,
  eTimeZoneID_Daylight := 2
);
END_TYPE

```

Wert	Bedeutung
eTimeZoneID_Invalid	Die Zeitzone-Konfiguration konnte nicht gelesen werden
eTimeZoneID_Unknown	Die Zeitzone-Konfiguration konnte zwar gelesen werden, die Normal-/Sommerzeit-Information ist aber unbekannt
eTimeZoneID_Standard	Es wird aktuell die Normalzeit benutzt
eTimeZoneID_Daylight	Es wird aktuell die Sommerzeit benutzt

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.19 E_TypeFieldParam

Stringformat Typfeld.

```

TYPE E_TypeFieldParam :
(
  TYPEFIELD_UNKNOWN := 0,
  TYPEFIELD_B, (* b or B: binary number *)
  TYPEFIELD_O, (* o or O: octal number *)
  TYPEFIELD_U, (* u or U: unsigned decimal number *)
  TYPEFIELD_C, (* c or C: one ASCII character *)
  TYPEFIELD_F, (* f or F: float number ( normalized format )*)
);

```

```

TYPEFIELD_D, (* d or D: signed decimal number *)
TYPEFIELD_S, (* s or S: string *)
TYPEFIELD_XU, (* X: hexadecimal number (upper case characters) *)
TYPEFIELD_XL, (* x: hexadecimal number (lower case characters) *)
TYPEFIELD_EU, (* E: float number (scientific format) *)
TYPEFIELD_EL (* e: float number (scientific format) *)
);
END_TYPE

```

Wert	Bedeutung
TYPEFIELD_UNKNOWN	Unbekannt oder nicht initialisiert
TYPEFIELD_B	b oder B: Binärzahl
TYPEFIELD_O	o oder O: Oktalzahl
TYPEFIELD_U	u oder U: Vorzeichenlose Dezimalzahl
TYPEFIELD_C	c oder C: Ein ASCII Zeichen
TYPEFIELD_F	f oder F: Fließkommazahl (normalisierte Darstellung)
TYPEFIELD_D	d oder D: Vorzeichenbehaftete Dezimalzahl
TYPEFIELD_S	s oder S: Zeichenkette
TYPEFIELD_XU	X: Hexadezimalzahl (upper case characters)
TYPEFIELD_XL	x: Hexadezimalzahl (lower case characters)
TYPEFIELD_EU	E: Fließkommazahl (wissenschaftliche Darstellung)
TYPEFIELD_EL	e: Fließkommazahl (wissenschaftliche Darstellung)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.20 FLOAT

LREAL alias Typ.

```

TYPE FLOAT :LREAL;
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.21 GUID

System ID.

```

TYPE GUID :
STRUCT
    Data1 : DWORD;
    Data2 : WORD;
    Data3 : WORD;
    Data4 : ARRAY[0..7] OF BYTE;
END_STRUCT
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.22 OTSTRUCT

Zeitformat für einen Betriebsstundenzähler.

```

TYPE OTSTRUCT :
STRUCT
  wWeek      : WORD;
  wDay       : WORD;
  wHour      : WORD;
  wMinute    : WORD;
  wSecond    : WORD;
  wMilliseconds : WORD;
END_STRUCT
END_TYPE

```

wWeek: Anzahl der Wochen: 0 ~ 65535;

wDay: Anzahl der Tage: 0 ~ 7;

wHour: Anzahl der Stunden: 0 ~ 23;

wMinute: Anzahl der Minuten: 0 ~ 59;

wSecond: Anzahl der Sekunden: 0 ~ 59;

wMilliseconds: Anzahl der Millisekunden: 0 ~ 999;

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.23 PROFILERSTRUCT

Stautsinformationen des Profiler-Funktionsbausteins.

```

TYPE PROFILERSTRUCT :
STRUCT
  LastExecTime    : DWORD;
  MinExecTime     : DWORD;
  MaxExecTime     : DWORD;
  AverageExecTime : DWORD;
  MeasureCycle    : DWORD;
END_STRUCT
END_TYPE

```

LastExecTime: Der letzte gemessene Wert der Ausführungszeit in [μ s].

MinExecTime: Die minimale Ausführungszeit in [μ s].

MaxExecTime: Die maximale Ausführungszeit in [μ s].

AverageExecTime: Die mittlere Ausführungszeit der 10 letzten Messungen in [μ s].

MeasureCycle: Anzahl der bereits durchgeführten Messungen.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.24 REMOTEPC

Remote-PC Konfigurationseintrag.

```

TYPE REMOTEPC :
STRUCT
    NetId : T_AmsNetId;
    Name : STRING(31);
END_STRUCT
ND_TYPE
    
```

NetId: Die Netzwerkadresse des Remote-PC's (Typ: T_AmsNetID);

Name: Die Remote-PC-Bezeichnung;

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.25 REMOTEPCINFOSTRUCT

Eine Liste mit mehreren Remote-PC Konfigurationseinträgen (Typ: REMOTEPC [▶ 310]).

```

TYPE REMOTEPCINFOSTRUCT : ARRAY[0..99] OF REMOTEPC;
END_TYPE
    
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.26 ST_AmsRouteEntry

Dieser Datentyp enthält Informationen über die Konfiguration einer TwinCAT Remoteverbindung.

```

TYPE ST_AmsRouteEntry :
STRUCT
    sName : STRING(MAX_ROUTE_NAME_LEN);
    sNetID : T_AmsNetId;
    sAddress : STRING(MAX_ROUTE_ADDR_LEN);
    eTransport : E_RouteTransportType;
    tTimeout : TIME;
    dwFlags : DWORD;
END_STRUCT
END_TYPE
    
```

sName: Symbolischer Name des entfernten TwinCAT Systems. Dieser Name kann frei vergeben werden. Die maximale Stringlänge ist durch eine Konstante (default: 31 Zeichen) begrenzt.

sNetID: Netzwerkadresse des entfernten TwinCAT Systems (Typ: T_AmsNetID).

sAddress: Systemadresse bezogen auf die jeweilige Transportschicht. Bei TCP/IP als Transportschicht wird hier die IP-Adresse angegeben. Die maximale Stringlänge ist durch eine Konstante (default: 79 Zeichen) begrenzt.

eTransport: Die Transportschicht mit der AMS Nachrichten befördert werden (Typ: E_RouteTransportType [▶ 307]). Zur Zeit wird nur die Transportschicht TCP/IP unterstützt.

tTimeout: Timeoutzeit. (zur Zeit reserviert und nicht benutzt).

dwFlags: Zusätzliche Optionen (zur Zeit reserviert und nicht benutzt).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.27 ST_AmsRouteEntryEx

Dieser Datentyp enthält Informationen über die Konfiguration einer TwinCAT Remoteverbindung.

```

TYPE ST_AmsRouteEntryEx :
STRUCT
  sName          : STRING (MAX_ROUTE_NAME_LEN);
  sNetID         : T_AmsNetId;
  sAddress       : STRING (MAX_ROUTE_ADDR_LEN);
  sVirtualNetID : T_AmsNetId;
  eTransport     : E_RouteTransportType;
  tTimeout      : TIME;
  dwFlags       : DWORD;
END_STRUCT
END_TYPE

```

sName: Symbolischer Name des entfernten TwinCAT Systems. Dieser Name kann frei vergeben werden. Die maximale Stringlänge ist durch eine Konstante (default: 31 Zeichen) begrenzt.

sNetID: Netzwerkadresse des entfernten TwinCAT Systems (Typ: T_AmsNetID).

sAddress: Systemadresse bezogen auf die jeweilige Transportschicht. Bei TCP/IP als Transportschicht wird hier die IP-Adresse angegeben. Die maximale Stringlänge ist durch eine Konstante (default: 79 Zeichen) begrenzt.

sVirtualNetID: Virtuelle Netzwerkadresse (Typ: T_AmsNetID) Siehe auch die Beschreibung der AmsNAT-Funktionalität.

eTransport: Die Transportschicht mit der AMS Nachrichten befördert werden (Typ: E_RouteTransportType [► 307]). Zur Zeit wird nur die Transportschicht TCP/IP unterstützt.

tTimeout: Timeoutzeit. (zur Zeit reserviert und nicht benutzt).

dwFlags: Zusätzliche Optionen (zur Zeit reserviert und nicht benutzt).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System) >= v3.3.41.0

5.28 ST_CheckLicense

Struktur mit Lizenzinformationen

```

TYPE ST_CheckLicense :
STRUCT
  stLicenseId    : GUID;
  tExpirationTime : TIMESTRUCT;
  sExpirationTime : STRING (80);
  eResult        : E_LicenseHResult;
  nCount         : UDINT;
END_STRUCT
END_TYPE

```

Name	Beschreibung
stLicenseId	Lizenz-ID
tExpirationTime	Verfallsdatum
sExpirationTime	Verfallsdatum
eResult	Lizenzstatus (siehe E_LicenseHResult [► 304])
nCount	Anzahl der Instanzen für diese Lizenz (0=unbegrenzt)

5.29 ST_DeviceIdentification

```

TYPE ST_DeviceIdentification :
STRUCT
  strTargetType      : STRING(30);
  strHardwareModel   : STRING(10);
  strHardwareSerialNo : STRING(12);
  strHardwareVersion : STRING(8);
  strHardwareDate    : STRING(12);
  strHardwareCPU     : STRING(20);
  strImageDevice     : STRING(48);
  strImageVersion    : STRING(32);
  strImageLevel      : STRING(32);
  strImageOsName     : STRING(48);
  strImageOsVersion  : STRING(8);
  strTwinCATVersion  : STRING(4);
  strTwinCATRevision : STRING(4);
  strTwinCATBuild    : STRING(8);
  strTwinCATLevel    : STRING(20);
  strAmsNetId       : T_AmsNetId;
END_STRUCT
END_TYPE
    
```

strTargetType : Typ des Zielsystems, z.B. 'CX1000-CE',

strHardwareModel : Hardware-Modell, z.B. '1001'.

strHardwareSerialNo : Hardware-Seriennummer, z.B. '123'.

strHardwareVersion : Hardware-Version, z.B. '1.7'.

strHardwareDate : Hardware-Produktionsdatum, z.B. '18.8.06'.

strHardwareCPU : Hardware-CPU-Architektur, z.B. 'INTELx86', 'ARM', 'UNKNOWN' oder " (leerer String).

strImageDevice : Software-Plattform, z.B. 'CX1000',

strImageVersion : Version der Software-Plattform, z.B. '2.15'.

strImageLevel : Level der Software-Plattform, z.B. 'HMI'.

strImageOsName : Name des Betriebssystems, z.B. 'Windows CE'.

strImageOsVersion : Version des Betriebssystems, z.B. '5.0'.

strTwinCATVersion : TwinCAT Version, z.B. bei TwinCAT 2.10.1307: '2'.

strTwinCATRevision : TwinCAT Reversion, z.B. bei TwinCAT 2.10.1307: '10'.

strTwinCATBuild : TwinCAT Build, z.B. bei TwinCAT 2.10.1307: '1307'.

strTwinCATLevel : Registrierter TwinCAT Level, z.B. 'PLC', 'NC-PTP', 'NC-I',

strAmsNetId : TwinCAT AMS-NetID, z.B. '5.0.252.31.1.1'.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.30 ST_DeviceIdentificationEx

```

TYPE ST_DeviceIdentificationEx :
STRUCT
  strTargetType      : STRING(30);
  strHardwareModel   : STRING(16);
  strHardwareSerialNo : STRING(16);
  strHardwareVersion : STRING(8);
  strHardwareDate    : STRING(12);
  strHardwareCPU     : STRING(20);
  strImageDevice     : STRING(48);
    
```



```

    strImageVersion      : STRING(32);
    strImageLevel       : STRING(32);
    strImageOsName      : STRING(48);
    strImageOsVersion   : STRING(8);
    strTwinCATVersion   : STRING(4);
    strTwinCATRevision  : STRING(4);
    strTwinCATBuild     : STRING(8);
    strTwinCATLevel     : STRING(20);
    strAmsNetId        : T_AmsNetId;
END_STRUCT
END_TYPE

```

strTargetType : Typ des Zielsystems, z.B. 'CX1000-CE',

strHardwareModel : Hardware-Modell, z.B. '1001'.

strHardwareSerialNo : Hardware-Seriennummer, z.B. '123'.

strHardwareVersion : Hardware-Version, z.B. '1.7'.

strHardwareDate : Hardware-Produktionsdatum, z.B. '18.8.06'.

strHardwareCPU : Hardware-CPU-Architektur, z.B. 'INTELx86', 'ARM', 'UNKNOWN' oder "" (leerer String).

strImageDevice : Software-Plattform, z.B. 'CX1000',

strImageVersion : Version der Software-Plattform, z.B. '2.15'.

strImageLevel : Level der Software-Plattform, z.B. 'HMI'.

strImageOsName : Name des Betriebssystems, z.B. 'Windows CE'.

strImageOsVersion : Version des Betriebssystems, z.B. '5.0'.

strTwinCATVersion : TwinCAT Version, z.B. bei TwinCAT 2.10.1307: '2'.

strTwinCATRevision : TwinCAT Reversion, z.B. bei TwinCAT 2.10.1307: '10'.

strTwinCATBuild : TwinCAT Build, z.B. bei TwinCAT 2.10.1307: '1307'.

strTwinCATLevel : Registrierter TwinCAT Level, z.B. 'PLC', 'NC-PTP', 'NC-I',

strAmsNetId : TwinCAT AMS-NetID, z.B. '5.0.252.31.1.1'.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.31 ST_FileAttributes

Datei- oder Verzeichnis-Attribute.

```

TYPE ST_FileAttributes :
STRUCT
    bReadOnly          : BOOL; (* FILE_ATTRIBUTE_READONLY *)
    bHidden            : BOOL; (* FILE_ATTRIBUTE_HIDDEN *)
    bSystem            : BOOL; (* FILE_ATTRIBUTE_SYSTEM *)
    bDirectory         : BOOL; (* FILE_ATTRIBUTE_DIRECTORY *)
    bArchive           : BOOL; (* FILE_ATTRIBUTE_ARCHIVE *)
    bDevice            : BOOL;
(* FILE_ATTRIBUTE_DEVICE. Under CE: FILE_ATTRIBUTE_INROM or FILE_ATTRIBUTE_ENCRYPTED *)
    bNormal            : BOOL; (* FILE_ATTRIBUTE_NORMAL *)
    bTemporary         : BOOL; (* FILE_ATTRIBUTE_TEMPORARY *)
    bSparseFile        : BOOL; (* FILE_ATTRIBUTE_SPARSE_FILE *)
    bReparsePoint      : BOOL; (* FILE_ATTRIBUTE_REPARSE_POINT *)
    bCompressed         : BOOL; (* FILE_ATTRIBUTE_COMPRESSED *)
    bOffline           : BOOL; (* FILE_ATTRIBUTE_OFFLINE. Under CE: FILE_ATTRIBUTE_ROMSTATICREF *)
    bNotContentIndexed : BOOL;
(* FILE_ATTRIBUTE_NOT_CONTENT_INDEXED. Under CE: FILE_ATTRIBUTE_ROMMODULE *)

```

```
bEncrypted      : BOOL; (* FILE_ATTRIBUTE_ENCRYPTED *)
END_STRUCT
END_TYPE
```

bReadOnly: Die Datei oder Verzeichnis hat nur einen Lesezugriff. Die Datei kann von Applikationen gelesen werden, sie kann aber nicht beschrieben oder gelöscht werden. Im Falle eines Verzeichnisses können die Applikationen das Verzeichnis nicht löschen.

bHidden: Die Datei oder Verzeichnis ist versteckt und wird nicht angezeigt in einer Standardauflistung.

bSystem: Die Datei oder Verzeichnis gehört zum Teil des Betriebssystems oder wird exklusiv vom Betriebssystem benutzt.

bDirectory: Über dieses Attribut wird ein Verzeichnis identifiziert.

bArchive: Die Datei oder Verzeichnis gehören zum Archiv. Applikationen benutzen dieses Attribut um die Dateien für ein Backup oder für das Entfernen zu markieren.

bDevice: Reserviert.

bNormal: Die Datei oder Verzeichnis hat keine anderen Attribute gesetzt. Dieses Attribut ist nur dann gültig wenn es allein benutzt wird.

bTemporary: Die Datei wird nur kurzzeitig für die Aufbewahrung der Daten benutzt.

bSparseFile: Die Datei ist eine abgespeckte Datei.

bReparsePoint: Mit der Datei oder Verzeichnis wurde ein "reparse point" assoziiert.

bCompressed: Die Datei oder Verzeichnis ist komprimiert. Bei der Datei sind die Daten komprimiert und beim Verzeichnis ist die Komprimierung per Default aktiv für neu erstellte Dateien oder Unterverzeichnisse.

bOffline: Die Datei ist nicht immer verfügbar.

bNotContentIndexed: Die Datei ist nicht indiziert beim Indizierservice.

bEncrypted: Die Datei oder das Verzeichnis ist verschlüsselt.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.32 ST_FileRBufferHead

Ringpufferdatei-Header-Status. Diese Struktur wird von dem Funktionsbaustein `FB_FileRingBuffer` [► 59] verwendet. Beim Öffnen der Ringpufferdatei wird diese Struktur gelesen und beim Schließen in die Ringpufferdatei gespeichert. Beim Lesen/Schreiben der Datensätze wird diese Struktur immer aktualisiert.

```
TYPE ST_FileRBufferHead :
STRUCT
    status      : DWORD := 0; (* buffer status flags Bit 0 = 1 => Opened, Bit 0 = 0 => Closed, Bit 1 = 1
file corrupted, all other bits are reserved *)
    access      : UDINT := 0; (* access counter, increments every time the buffer is reopened *)
    nID         : UDINT := 0; (* user defined value *)
    cbBuffer    : UDINT := 16#100000; (* max. buffer size (1MB) *)
    nCount      : UDINT := 0; (* number of fifo entries *)
    cbSize      : UDINT := 0; (* current (used) file buffer data byte length *)
    ptrFirst    : UDINT := 0; (* seek pointer start position of first (oldest) buffer entry *)
    ptrLast     : UDINT := 0; (* seek pointer end position of last (newest) buffer entry *)
    rsrv0       : UDINT := 0; (* reserved *)
    rsrv1       : UDINT := 0; (* reserved *)
    rsrv2       : UDINT := 0; (* reserved *)
    rsrv3       : UDINT := 0; (* reserved *)
END_STRUCT
END_TYPE
```

status: Status-Flags. Bit 0 = 1 => Datei ist geöffnet, Bit 0 = 0 => Datei ist geschlossen. Bit 1 = 1 => Datei ist korrupt (wurde vorher nicht richtig geschlossen oder die maximale Puffergröße passt nicht).

access: Zugriffszähler. Beim jedem erneuten Öffnen der Datei wird dieser Zähler inkrementiert.

nID: Benutzerdefinierter 32 Bit Wert.

cbBuffer: Die max. Ringpuffer-Dateigröße.

nCount: Aktuelle Anzahl der gespeicherten Datensätze.

cbSize: Aktuelle Anzahl der gespeicherten Datenbytes.

ptrFirst: Dateizeiger-Position des ältesten Datensatzes.

ptrLast: Dateizeiger-Position des neuesten Datensatzes.

rsrv0..rsrv3: Reserviert.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.33 ST_FindFileEntry

Dieser Datentyp wird von den Dateisuche-Funktionsbausteinen: [FB_EnumFindFileEntry \[► 51\]](#) und [FB_EnumFindFileList \[► 53\]](#) verwendet.

```

TYPE ST_FindFileEntry :
STRUCT
  sFileName           : T_MaxString;
  sAlternateFileName : STRING(13);
  fileAttributes     : ST_FileAttributes;
  fileSize           : T_ULARGE_INTEGER;
  creationTime       : T_FILETIME;
  lastAccessTime     : T_FILETIME;
  lastWriteTime      : T_FILETIME;
END_STRUCT
END_TYPE

```

sFileName: Nullterminierter String mit dem Namen der Datei oder des Verzeichnisses (Typ: [T_MaxString](#)).

sAlternateFileName: Nullterminierter String mit dem Alternativnamen der Datei oder des Verzeichnisses im klassischen 8.3 Format (filename.ext).

fileAttributes: Struktur mit Datei-/Verzeichnis-Attributen (Typ: [ST_FileAttributes \[► 314\]](#)).

fileSize: Bytegröße der Datei (64 bit Zahl, Typ: [T_ULARGE_INTEGER \[► 327\]](#)).

creationTime: Die Strukturvariable gibt an wann die Datei oder das Verzeichnis erstellt wurde (Typ: [T_FILETIME \[► 323\]](#)).

lastAccessTime: Bei einer Datei gibt die Struktur an wann die Datei zuletzt gelesen oder geschrieben wurde (Typ: [T_FILETIME \[► 323\]](#)). Beim Verzeichnis gibt die Struktur an wann das Verzeichnis erstellt wurde.

lastWriteTime: Bei einer Datei gibt die Struktur an wann in die Datei zuletzt geschrieben wurde (Typ: [T_FILETIME \[► 323\]](#)). Beim Verzeichnis gibt die Struktur an wann das Verzeichnis erstellt wurde.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.34 ST_IPAdapterHwAddr

Physikalische Adresse (MAC).

```
TYPE ST_IPAdapterHwAddr :
STRUCT
    length : UDINT := 0;
    b      : ARRAY[0..MAX_ADAPTER_ADDRESS_LENGTH] OF BYTE;
END_STRUCT
END_TYPE
```

length: Bytelänge der physikalischen Hardwareadresse.

b: MAC-Adressbytes.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.35 ST_IPAdapterInfo

Netzwerkadapterinformationen.

```
TYPE ST_IPAdapterInfo :
STRUCT
    bDefault      : BOOL;
    sAdapterName  : STRING(MAX_ADAPTER_NAME_LENGTH) := '';
    sDescription  : STRING(MAX_ADAPTER_DESCRIPTION_LENGTH) := '';
    physAddr     : ST_IPAdapterHwAddr;
    dwIndex      : DWORD;
    eType        : E_MIB_IF_Type;
    sIpAddr      : T_IPv4Addr;
    sSubNet      : T_IPv4Addr;
    sDefGateway  : T_IPv4Addr;
    bDhcpEnabled : BOOL;
    sDhcpSrv     : T_IPv4Addr;
    bHaveWins    : BOOL;
    sPrimWinsSrv : T_IPv4Addr;
    sSecWinsSrv  : T_IPv4Addr;
    tLeaseObt    : DT;
    tLeaseExp    : DT;
END_STRUCT
END_TYPE
```

bDefault: Dieser Variable wird zur Zeit **nur unter Windows CE** verwendet! Beim TRUE wird der Netzwerkadapter als Default-Adapter von TwinCAT verwendet.

sAdapterName: Adaptername als String.

sDescription: Adapterbeschreibung als String.

physAddr: Physikalische Hardwareadresse. (Typ: [ST_IPAdapterHwAddr](#) [► 317])

dwIndex: Interner Adapter-Systemindex.

eType: Adapter-Typ (Typ: [E_MIB_IF_Type](#) [► 305]).

sIpAddr: IP-Adresse (Typ: [T_Ipv4Addr](#)).

sSubNet: IP-Netzmaske (Typ: [T_Ipv4Addr](#)).

sDefGateway: IP-Adresse des Default-Gateways (Typ: [T_Ipv4Addr](#)).

bDhcpEnabled: Gibt an ob DHCP für diesen Adapter aktiviert wurde oder nicht.

sDhcpSrv: IP-Adresse des DHCP-Servers (Typ: [T_Ipv4Addr](#)).

bHaveWins: Gibt an ob Windows Internet Name Service (WINS) verwendet wird oder nicht.

sPrimWinsSrv: IP-Adresse des primären WINS-Servers (Typ: T_Ipv4Addr).

sSecWinsSrv: IP-Adresse des sekundären WINS-Servers (Typ: T_Ipv4Addr).

tLeaseObt: Gibt an wann die IP-Adresse vom DHCP-Server „gemietet“ wurde (UTC).

tLeaseExp: Gibt an wie lange die IP-Adresse vom DHCP-Server „vermietet“ werden darf bevor eine „Verlängerung“ vom DHCP-Server beantragt werden muss (UTC).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.36 ST_LicenseDongle

Struktur mit Identifikationsdaten aller angeschlossenen Lizenz-Dongles

```

TYPE ST_LicenseDongle :
STRUCT
  stAmsAddr      : AMSADDR;
  eDevType       : E_LDevType;
  nFlags         : UDINT;
  eDongleStatus  : E_LDongleStatus;
  nSerialNo      : UDINT;
  nReserved1     : UDINT;
  nReserved2     : UDINT;
END_STRUCT
END_TYPE

```

Name	Beschreibung
stAmsAddr	Netzwerkennung (AmsNetId und Port) des Lizenz-Dongles
eDevType	Siehe E_LDevType [► 304]
nFlags	0: statisch konfiguriert 1: dynamischer Dongle
eDongleStatus	Validierungsstatus des Lizenz-Dongles (siehe E_LDongleStatus [► 304])
nSerialNo	ID-Nummer des Dongles / der Lizenzklemme
nReserved1	Reserviert für zukünftige Verwendung
nReserved2	Reserviert für zukünftige Verwendung

5.37 ST_ReadEvent

Informationen zu einer mittels [FB_AdsReadEvents \[► 40\]](#) gelesenen Meldung.

```

TYPE ST_ReadEvent :
STRUCT
  nSourceId      : UDINT;
  nEventId       : UDINT;
  nClass         : DWORD;
  nConfirmState  : DWORD;
  nResetState    : DWORD;
  sSource        : STRING(255);
  sDate          : STRING(23);
  sTime          : STRING(23);
  sComputer      : STRING;
  sMessageText   : STRING(255);
  bQuitMessage   : BOOL;
  bConfirmable   : BOOL;
END_STRUCT
END_TYPE

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.38 ST_SplittedBIC

Die Struktur ST_SplittedBIC kann über die Funktion F_SplitBIC [► 271] der Tc2_Utilities anhand vom Beckhoff Identification Conde (BIC) gefüllt werden. Die Kennungen und Datengrößen der Teilstrings, mit denen die BIC zerlegt wird, stehen als Kommentar an den Strukturelementen.

```

TYPE ST_SplittedBIC :
STRUCT
  sItemNo : STRING(6); //1, '1P' 8(2+6)
  sBTN : STRING(8); //2, 'SBTN' 12(4+8)
  sDescription : STRING(30); //3, '1K' 32(2+30)
  sQuantity : STRING(5); //4, 'Q' 6(1+5)
  sBatchNo : STRING(12); //5, '2P' 14(2+12)
  sIdSerialNo : STRING(9); //6, '51S' 12(3+9)
  sVariantNo : STRING(9); //7, '30P' 12(3+9)
  sDataCode : STRING(6); //8, '9D' 8(2+6)
  sOrderBatchNo : STRING(12); //9, '1T' 14(2+12)
  sSerialNo : STRING(20); //10, '52S' 23(3+20)
  sPackUnitQuantity : STRING(5); //11, '4QPU' 9(4+5)
  sDataElement : STRING(11); //12, '6D' 13(2+11)
  sMoistSensLevel : STRING(4); //13, 'Z' 5(1+4)
  sPlatingMaterial : STRING(2); //14, 'E' 3(1+2)
  sManufacturer : STRING(9); //15, '12V' 12(3+9)
  sCountryOfOrigin : STRING(2); //16, '4L' 4(2+2)
  sCustomSpecItemNo : STRING(16); //17, 'P' 17(1+16)
  sUndefined : STRING(819); //remaining undefined BIC string (1023 + /0)
END_STRUCT
END_TYPE
    
```

Name	Beschreibung
sItemNo	Artikelnummer, Länge max. 6 Zeichen, z.B. „072222“
sBTN	Beckhoff Traceability Number (BTN), Länge max. 8 Zeichen, z.B. „k4p562d7“
sDescription	Artikelbezeichnung, Länge max. 30 Zeichen, z.B. „KL9010“
sQuantity	Menge, Länge max. 5 Zeichen, z.B. „1“
sBatchNo	Chargennummer, Länge max. 12 Zeichen, z.B. „401503180016“
sIdSerialNo	ID-/Seriennummer, Länge max. 9 Zeichen, z.B. „678294104“
sVariantNo	Variante, Länge max. 9 Zeichen, z.B. „000048443“
sDataCode	Lieferantenspezifischer Data Code, Länge max. 6 Zeichen, z.B. „G0118“
sOrderBatchNo	Lieferantenspezifische Auftrags-/Chargennummer, Länge max. 12 Zeichen, z.B. „FA12345678“
sSerialNo	Lieferantenspezifische Seriennummer, Länge max. 20 Zeichen, z.B. „2304853-1-004“
sPackUnitQuantity	VPE Menge, Länge max. 5 Zeichen, z.B. „1000“
sDataElement	Date Code YYYYMMDD, spezifiziert, Länge max. 11 Zeichen, z.B. „20190315146“
sMoistSensLevel	MS-Trockenlevel, Länge max. 4 Zeichen, z.B. „MSL3“
sPlatingMaterial	Beschichtungsmaterial (gemäß JEDEC J-STD-609), Länge max. 2 Zeichen, z.B. „e1“
sManufacturer	Hersteller (DUNS-Nummer), Länge max. 9 Zeichen, z.B. „313291892“
sCountryOfOrigin	Ursprungsland, Länge max. 2 Zeichen, z.B. „CN“
sCustomSpecItemNo	Kundenspezifische Artikelnummer/Teilenummer, Länge max. 16 Zeichen, z.B. „0000000107353052“
sUndefined	Nicht erkannter Reststring der BIC (Länge max. 819 Zeichen), z.B. nach Erweiterung der BIC um zusätzliche Elemente, ggf. prüfen, ob neue Version der Tc2_Utilities Bibliothek verfügbar ist

5.39 ST_TcOnlineLicenseInfoDataEx

```

TYPE ST_TcOnlineLicenseInfoDataEx :
STRUCT
  stLicenseId      : GUID;
  stLicenseName    : STRING(80);
  tExpirationTime  : TIMESTRUCT;
  sExpirtaionTime  : STRING(80);
  nMaxCount        : UDINT;
  nUsedCount       : UDINT;
  eResult          : E_LicenseHResult;
  nVolumeNo       : UDINT;
  nOptInfo         : WORD;
  nRestriction     : WORD;
  bOemLicense      : BOOL;
  bBeckhoffLicense : BOOL;
  bBeckhoffPC     : BOOL;
  bEtherCATDongle  : BOOL;
  bUSB Dongle     : BOOL;
  bGenDevTypeLic  : BOOL;
END_STRUCT
END_TYPE

```

Name	Beschreibung
stLicenseId	Lizenz-ID
stLicenseName	Lizenzname
tExpirationTime	Verfallsdatum
sExpirationTime	Verfallsdatum
nMaxCount	Maximale Anzahl nutzbarer Instanzen für diese Lizenz (0=unbegrenzt)
nUsedCount	Anzahl genutzter Instanzen dieser Lizenz
eResult	Lizenzstatus (siehe E_LicenseHResult [▶ 304])
nVolumeNo	(Nur bei Volumenlizenzen): Volume-Bundle-Nummer 0: keine Bundle-Nummer
nOptInfo	(Intern)
nRestriction	Lizenz hat Einschränkungen für die Verwendung
bOemLicense	OEM-Lizenz
bBeckhoffLicense	Beckhoff-Lizenz
bBeckhoffPC	Lizenz für Beckhoff-Hardware (IPC, CX usw.)
bEtherCATDongle	Lizenz ist für EtherCAT-Lizenzklemme, z. B. EL6070
bUSB Dongle	Lizenz ist für USB-Lizenz-Dongle, z. B. C9900-L100
bGenDevTypeLic	Lizenz ist für den Gerätetyp

5.40 ST_TcOnlineLicensesInfoData

Lizenz-Informationen.

```

TYPE ST_TimeZoneInformation :
STRUCT
  stLicenseId      : GUID;
  sLicenseName     : STRING(80);
  tExpirationTime  : TIMESTRUCT;
  sExpirationTime  : STRING(80);
  nMaxCount        : UDINT;
  nUsedCount       : UDINT;
  eResult          : E_LicenseHResult;
END_STRUCT
END_TYPE

```

stLicenseId: Definiert die Lizenz ID als [GUID](#) [[▶ 309](#)].

sLicenseName: Bezeichnung der Lizenz als String.

tExpirationTime: Gibt die zeitliche Gültigkeit der Lizenz an (Typ: [TIMESTRUCT](#) [[▶ 327](#)]).

sExpirationTime: Gibt die zeitliche Gültigkeit der Lizenz als String an.

nMaxCount: Gibt die maximale Anzahl von erlaubten Instanzen an, sofern es sich um eine Lizenz handelt, welche eine Limitierung von Instanzen beinhaltet (Bsp: TC3 NC PTP Axes Pack 25).

nUsedCount: Gibt die Anzahl von verwendeten Instanzen an, sofern es sich um eine Lizenz handelt, welche eine Limitierung von Instanzen beinhaltet.

eResult: Gibt für diese Lizenz einen Fehlercode als HRESULT-Enumeration aus. (Hierbei wird ein Fehler durch einen negativen Wert repräsentiert.)

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4018	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) v3.3.9.0 oder höher

5.41 ST_TcRouterStatusInfo

TwinCAT-Router-Statusinformation.

```

TYPE ST_TcRouterStatusInfo :
STRUCT
    maxMem      : DWORD; (* Max. router memory byte size *)
    maxMemAvail : DWORD; (* Available router memory byte size *)
    regPorts    : DWORD; (* Number of registered ports *)
    regDrivers  : DWORD; (* Number of registered TwinCAT server ports *)
    amsDebugLog : BOOL; (* TRUE = Ams logging/debugging enabled, FALSE = Ams logging/
debugging disabled *)
END_STRUCT
END_TYPE
    
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.42 ST_TimeZoneInformation

Zeitzone-Informationen. Die Normalzeit wird auch als Winterzeit bezeichnet. Die bias-Parameter können auch negative Werte annehmen.

```

TYPE ST_TimeZoneInformation :
STRUCT
    bias          : DINT
    standardName  : STRING(31);
    standardDate  : TIMESTRUCT;
    standardBias  : DINT;
    daylightName  : STRING(31);
    daylightDate  : TIMESTRUCT;
    daylightBias  : DINT;
END_STRUCT
END_TYPE
    
```

bias: Definiert die aktuelle Differenz der Lokalzeit zur UTC-Zeit in Minuten. UTC = local time + bias.

standardName: Bezeichnung der Normalzeit als String.

standardDate: Diese Struktur beinhaltet Informationen zum Übergang von Sommerzeit zur Normalzeit (Typ: TIMESTRUCT [▶ 327]). Der Strukturparameter *wMonth* ist Null wenn dieser Wert nicht benutzt wird. Wenn dieser Parameter benutzt wird, dann muss auch der *daylightDate*-Parameter benutzt werden. Um *standardDate* konfigurieren zu können setzen Sie den *wYear*-Parameter gleich Null, bei *wDayOfWeek* wählen Sie den gewünschten Wochentag und bei *wDay* einen Wert zwischen 1 und 5 (Woche im Monat, 5 entspricht der letzten Woche).

standardBias: Zeitdifferenz in Minuten für Berechnungen der Lokalzeit während der Normalzeit. Dieser Wert ist meistens Null.

daylightName: Bezeichnung der Sommerzeit als String.

daylightDate: Diese Struktur beinhaltet Informationen zum Übergang von Normalzeit zur Sommerzeit (Typ: [TIMESTRUCT](#) [[▶ 327](#)]). Der Strukturparameter *wMonth* ist Null wenn dieser Wert nicht benutzt wird. Wenn dieser Parameter benutzt wird, dann muss auch der *standardDate*-Parameter benutzt werden. Um *daylightDate* konfigurieren zu können setzen Sie den *wYear*-Parameter gleich Null, bei *wDayOfWeek* wählen Sie den gewünschten Wochentag und bei *wDay* einen Wert zwischen 1 und 5 (Woche im Monat, 5 entspricht der letzten Woche).

daylightBias: Zeitdifferenz in Minuten für Berechnungen der Lokalzeit während der Sommerzeit.

Beispiel:

Siehe: [FB_SetTimeZoneInformation](#) [[▶ 111](#)].

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.43 SYMINFOSTRUCT

TwinCAT SPS Symbolinformation.

```

TYPE SYMINFOSTRUCT :
STRUCT
    symEntryLen    : UDINT;
    idxGroup       : UDINT;
    idxOffset      : UDINT;
    byteSize       : UDINT;
    adsDataType    : ADSDATATYPEID;
    symDataType    : T_MaxString;
    symComment     : T_MaxString;
END_STRUCT
END_TYPE

```

symEntryLen: Die tatsächliche Bytelänge des Symboleintrags in der Symboltabelle. Die Symbole werden in einer Symboltabelle abgelegt. Die Länge der einzelnen Einträge ist variabel und abhängig von der Länge des Symbolnamens, der Typbezeichnung und des Kommentars.

idxGroup: Der Index-Group der Symbolvariablen;

idxOffset: Der Index-Offset der Symbolvariablen;

byteSize: Die tatsächliche Speicherlänge, die der Wert der Symbolvariablen belegt in Byte. Eine boolsche SPS-Variable belegt z.B. einen Byte und ein String mit 20 Zeichen belegt tatsächlich 21 Byte (20 Byte für Zeichen + ein Byte für die abschließende Null);

adsDataType: Die ADS-Datentyp-Id. (Typ: [ADSDATATYPEID](#) [[▶ 300](#)]) Diese Typ-Bezeichnung wird beim ADS-Zugriff auf symbolische Variablen benutzt. Alle SPS-Strukturen und Arrays (selbstdefinierte Datentypen) besitzen die Ads-Datentypbezeichnung: ADST_BIGTYPE und können über diese Datentypkonstante nicht identifiziert werden. Um die selbstdefinierten Datentypen identifizieren zu können, benutzen Sie die *symDataType*-Variable oder lesen Sie die Basistypen der einzelnen Variablen der Struktur.

symDataType: Die Datentypbezeichnung der Symbolvariablen als String. Z.B. der Typ-Name einer vom Benutzer definierten SPS-Datenstruktur (Typ: T_MaxString, max. 255 Zeichen).

symComment: Der Kommentar zu der Symbolvariablen, den der Benutzer in der Zeile der SPS-Variablendefinition hinzugefügt hat (Typ: T_MaxString, max. 255 Zeichen).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.44 T_Arg

Argumenttyp für Stringformat-Funktionen/Funktionsbausteine.

```

TYPE T_Arg :
STRUCT
  eType : E_ArgType := ARGTYPE_UNKNOWN;
  cbLen : UDINT     := 0;
  pData : PVOID     := 0;
END_STRUCT
END_TYPE
    
```

eType : Datentypkennung (Typ: [E_ArgType](#) [[▶ 301](#)]).

cbLen : Anzahl der Bytes, die im Speicher belegt werden.

pData : Adresspointer.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.45 T_FILETIME

Variablen von diesem Typ sind 64 bit Zahlen. Der Wert entspricht der Anzahl der 100-Nanosekunden-Intervalle seit dem 1 Januar 1601 (UTC).

```

TYPE T_FILETIME :
STRUCT
  dwLowDateTime : DWORD;
  dwHighDateTime : DWORD;
END_STRUCT
END_TYPE
    
```

dwLowDateTime: Unteren 32 bit.

dwHighDateTime: Oberen 32 bit.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.46 T_FILETIME64

Variablen von diesem Typ sind 64 bit-Zahlen. Der Wert entspricht der Anzahl der 100-Nanosekunden-Intervalle seit dem 1 Januar 1601 (typischerweise UTC).

```

TYPE T_FILETIME64 : ULINT; END_TYPE
    
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.4024	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.44.0

5.47 T_FIX16

Variablen von diesem Typ repräsentieren eine 16 Bit vorzeichenbehaftete Festkommazahl. Dieser Datentyp wird oft bei Systemen benutzt die keine FPU-Einheit besitzen (z.B.: Mikrokontroller oder Geräte aus dem Bereich der Fernwirktechnik). Wenn z.B. über die serielle Schnittstelle Daten im Festkommazahlenformat ausgetauscht werden sollten, dann müssen diese Daten in das passende Format konvertiert werden.

Die Anzahl der Nachkommastellen wird passend zum benötigten Zahlenbereich und Auflösung gewählt. Bei 15 Nachkommastellen lassen sich z.B. Festkommazahlen im Bereich: $-1..1 \cdot 2^{15}$ darstellen. Dies entspricht etwa dem Fließkommazahlenbereich: $-1..0.999969482421875$.

Im Gegensatz zu Fließkommazahlen ist die Auflösung der Festkommazahlen im gesamten Zahlenbereich konstant. Leider haben die Festkommazahlen einen kleineren darstellbaren Zahlenbereich. Vorsicht ist geboten bei mathematischen Operationen die einen positiven oder negativen Überlauf generieren können.

```
TYPE T_FIX16 :
STRUCT
  value : INT := 0;
  n      : WORD(0..15);
  status : DWORD := 0;
END_STRUCT
END_TYPE
```

value: Diese Membervariable beinhaltet den eigentlichen Wert der Festkommazahl (16 Bit Vor- und Nachkommastellen).

n: Anzahl der Nachkommastellen. Zulässiger Bereich: 0..15. Das höherwertigste Bit ist für das Vorzeichenbit reserviert.

status: Status-Flags (reserviert, zur Zeit nicht benutzt).

Beispiel 1:

Ein A/D-C liefert Messwerte als vorzeichenbehaftete 16 Bit Festkommazahlen mit 15 Nachkommastellen. Diese Messwerte wurden in die SPS eingelesen und sollen in LREAL-Datentyp konvertiert werden.

```
PROGRAM FIX_TO_FLOAT
VAR
  adc_0      : WORD := 2#1010000000000000; (* = -0.75 (Q0.15) *)
  adc_1      : WORD := 2#0111000000000000; (* = +0.875 (Q0.15) *)
  fix_0, fix_1 : T_FIX16;
  dbl_0, dbl_1 : LREAL;
END_VAR

fix_0 := WORD_TO_FIX16( adc_0, 15 );
fix_1 := WORD_TO_FIX16( adc_1, 15 );
dbl_0 := FIX16_TO_LREAL( fix_0 );
dbl_1 := FIX16_TO_LREAL( fix_1 );
```

Beispiel 2:

Die Parameter eines Micro-Controllers sind vorzeichenbehaftete 16 Bit Festkommazahlen mit 8 Nachkommastellen. Die LREAL-Parameter in der SPS sollen in diesen Format konvertiert werden.

```
PROGRAM FLOAT_TO_FIX
VAR
  dbl_0      : LREAL := +3.5;
  dbl_1      : LREAL := -3.5;
  fix_0, fix_1 : T_FIX16;
  ctrl_0, ctrl_1 : WORD;
END_VAR

fix_0 := LREAL_TO_FIX16( dbl_0, 8 );
fix_1 := LREAL_TO_FIX16( dbl_1, 8 );
ctrl_0 := FIX16_TO_WORD( fix_0 );
ctrl_1 := FIX16_TO_WORD( fix_1 );
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

5.48 T_HashTableEntry

Ein Hash-Tabellen-Eintrag/Element.

```

TYPE T_HashTableEntry :
STRUCT
    key    : DWORD := 0; (* Entry key *)
    value  : PVOID := 0; (* Entry value *)
END_STRUCT
END_TYPE
    
```

key: Schlüssel (32 Bit vorzeichenlose Zahl oder auch 32 Bit Pointer).

value: Wert (32/64 Bit vorzeichenlose Zahl oder auch Pointer sind möglich).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.49 T_HHASHTABLE

Ein Hash-Tabellen-Handle. Das Hash-Tabellen-Handle wird von dem Funktionsbaustein: [FB_HashTableCtrl](#) [[▶ 80](#)] verwendet.

```

TYPE T_HHASHTABLE :
STRUCT
    nCount : UDINT := 0;
    nFree  : UDINT := 0;
END_STRUCT
END_TYPE
    
```

nCount: Anzahl der belegten Elemente.

nFree: Anzahl der freien Elemente.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.50 T_HLINKEDLIST

Ein Linked-List-Handle. Das Linked-List-Handle wird von dem Funktionsbaustein: [FB_LinkedListCtrl](#) [[▶ 88](#)] verwendet.

```

TYPE T_HLINKEDLIST :
STRUCT
    nCount : UDINT := 0;
    nFree  : UDINT := 0;
END_STRUCT
END_TYPE
    
```

nCount: Anzahl der belegten Elemente.

nFree: Anzahl der freien Elemente.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.51 T_HUGE_INTEGER

Variablen von diesem Typ repräsentieren eine TwinCAT 2 vorzeichenbehaftete 128 bit Zahl („legacy“-Typ).

```
TYPE T_HUGE_INTEGER :
STRUCT
  qwLowPart  : T_ULONG_INTEGER;
  qwHighPart : T_ULONG_INTEGER;
END_STRUCT
END_TYPE
```

qwLowPart: Unteren 64 bit.

qwHighPart: Oberen 64 bit.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.52 T_LARGE_INTEGER

Variablen von diesem Typ repräsentieren eine TwinCAT 2 vorzeichenbehaftete 64 bit Zahl („legacy“-Typ).

```
TYPE T_LARGE_INTEGER :
STRUCT
  dwLowPart  : DWORD;
  dwHighPart : DWORD;
END_STRUCT
END_TYPE
```

dwLowPart: Unteren 32 bit.

dwHighPart: Oberen 32 bit.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.53 T_LinkedListEntry

Variablen von diesem Typ repräsentieren ein Knoten/Element einer verketteten Liste.

```
TYPE T_LinkedListEntry :
STRUCT
  value : PVOID := 0;
END_STRUCT
END_TYPE
```

value: Wert (32/64 Bit, vorzeichenlose Zahl oder auch Pointer sind möglich).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.54 T_UHUGE_INTEGER

Variablen von diesem Typ repräsentieren eine TwinCAT 2 vorzeichenlose 128 bit Zahl („legacy“-Typ).

```

TYPE T_UHUGE_INTEGER :
STRUCT
    qwLowPart : T_ULONG_INTEGER;
    qwHighPart : T_ULONG_INTEGER;
END_STRUCT
END_TYPE
    
```

qwLowPart: Unteren 64 bit.

qwHighPart: Oberen 64 bit.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.55 T_ULONG_INTEGER

Variablen von diesem Typ repräsentieren eine TwinCAT 2 vorzeichenlose 64 bit Zahl („legacy“-Typ).

```

TYPE T_ULONG_INTEGER :
STRUCT
    dwLowPart : DWORD;
    dwHighPart : DWORD;
END_STRUCT
END_TYPE
    
```

dwLowPart: Unteren 32 bit.

dwHighPart: Oberen 32 bit.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

5.56 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_Utilities (System)

6 Globale Konstanten

6.1 Bibliotheksversion

Alle Bibliotheken haben eine bestimmte Version. Diese Version ist u. a. im SPS-Bibliotheks-Repository zu sehen. Eine globale Konstante enthält die Information über die Bibliotheksversion:

Global_Version

```
VAR_GLOBAL CONSTANT
    stLibVersion_Tc2_Uilities : ST_LibVersion;
END_VAR
```

stLibVersion_Tc2_Uilities: Versionsnummer der Tc2_Uilities-Bibliothek (Typ: ST_LibVersion).

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



Alle anderen Möglichkeiten Bibliotheksversionen zu vergleichen, die Sie von TwinCAT 2 kennen, sind veraltet.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

7 Globale Variablen

VAR_GLOBAL

Name	Typ	Wert	Verwendung	Bedeutung
MAX_AVERAGE_MEASURES	INT	10	Profiler [▶ 136]	Die Anzahl der gemittelten Messwerte. Mögliche Werte: 2..100
GLOBAL_FORMAT_HASH_PREFIX_TYPE	E_HashPrefixTypes	HASHPREFIX_IEC	FB FormatString [▶ 64], F FormatArgToStr [▶ 266]	Standard IEC prefix für Binär-, Oktal- oder Hexadezimal-Formatierung
GLOBAL_SBCS_TABLE	E_SBCSType [▶ 307]	eSBCS_WesternEuropean	F ToLCase [▶ 272], F ToUCase [▶ 273]	Windows SBCS (Single Byte Character Set) Code Page Table
GLOBAL_DCF77_PULSE_SPLIT	TIME	T#140ms	DCF77 TIME [▶ 32]	Pulselänge. 0 == Pulse < 140ms, 1 == Pulse > 140
GLOBAL_DCF77_SEQUENCE_CHECK	BOOL	FALSE	DCF77 TIME [▶ 32]	Plausibilitätsprüfung von zwei aufeinanderfolgenden Telegrammen: TRUE = Aktiviert, FALSE = Deaktiviert.
DEFAULT_CSV_FIELD_SEPARATOR	BYTE	16#2C	FB CSVMemBufferWriter [▶ 50], FB CSVmemBufferReader [▶ 49]	Datenfeldtrennzeichen. Semikolon= 16#3B => german field separator, Komma = 16#2C => US field separator

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

8 Beispiele

8.1 Beispiel: Kommunikation BC/BX<->PC/CX (F_SwapRealEx)

Das Beispiel zeigt die Verwendung der `F_SwapRealEx` [► 271]-Funktion. Dieses Beispiel beinhaltet zwei Komponenten: TwinCAT 2.xx BC/BX (Busklemmen Controller)-Applikation und TwinCAT 3.xx PC/CX (x86)-Applikation. Die PC/CX Applikation liest/schreibt eine Strukturvariable aus/in den Merkerbereich des BC's/ BX's. Die Strukturvariable beinhaltet REAL-Elemente. Diese müssen vor der Verwendung auf dem PC/CX oder vor dem Versand zum BC/BX in das richtige Format konvertiert werden.

Hier können Sie die kompletten Sourcen entpacken:

TwinCAT 2.xx - BC/BX (Busklemmen Controller) Applikation/Projektdatei: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_Uutilities/Resources/803333131.zip

TwinCAT 3.xx - PC/CX (x86, x64, ARM) Applikation/Archivdatei: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_Uutilities/Resources/803336971.zip

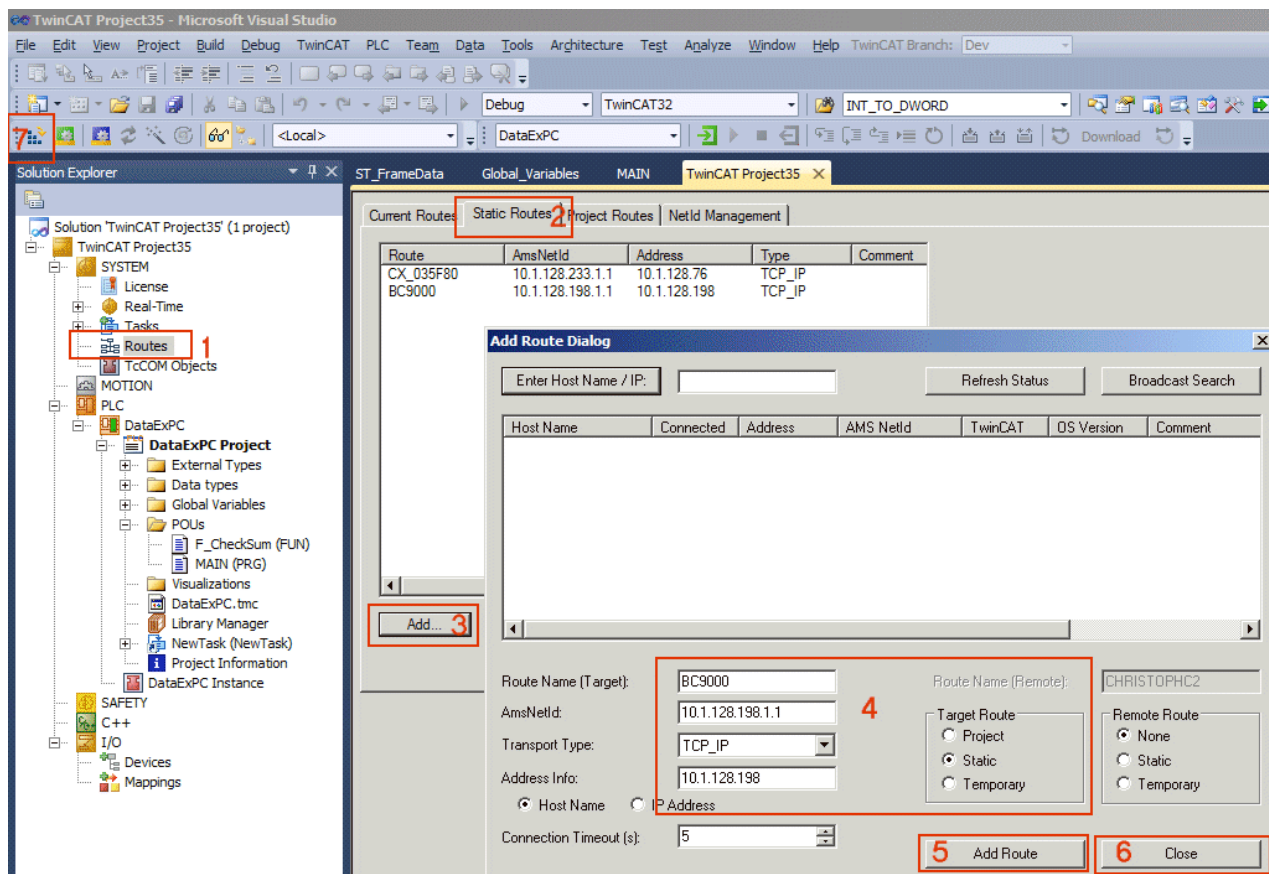
Systemanforderungen:

- TwinCAT 2.xx PLC (benötigt für den Download der BC/BX-Applikation) + BC/BX Hardware (z.B. BC9000);
- TwinCAT 3.xx Engineering und Laufzeitsystem (benötigt für den Download der PC/CX-Applikation);

Projekt herunterladen

Benutzen Sie die TwinCAT 2.xx PLC um die BC/BX-Applikation in das Laufzeitsystem eines Busklemmen Controllers runterzuladen (z.B. BC9000). Erstellen Sie ein Bootprojekt und starten Sie die SPS. Im nächsten Schritt benutzen Sie TwinCAT 3.xx und erstellen ein neues XAE-Projekt. Importieren Sie die Archivdatei in TwinCAT XAE mit einem rechten Mausklick auf den PLC-Knoten und dann auf *Add existing item..*

Damit auf den BC/BX (Busklemmen Controller) über ADS zugegriffen werden kann muss dieser als Gerät in die Liste der TwinCAT AMS Route-Verbindungen (routes) eingetragen werden. Erstellen Sie bitte eine neue statische Route (folgen Sie den Schritten im Bild). Die AmsNetId und IP Adresse des BC/BX muss passend konfiguriert werden (vergessen Sie bitte nicht die AmsNetID im SPS-Programcode).



Wichtiger Hinweis!

Der BC/BX (Busklemmen Controller) und PC/CX (x86, x64, ARM) hat eine unterschiedliche Speicherausrichtung (data alignment). Definieren Sie bitte Strukturen mit 8-Byte Speicherausrichtung wenn Sie diese für den Datenaustausch BC/BX <-> PC/CX verwenden wollen.

- TwinCAT 2.xx + PC/CX (x86) Plattform => Datenstrukturen haben 1 Byte Speicherausrichtung;
- TwinCAT 2.xx + CX (ARM) Plattform => Datenstrukturen haben 4 Byte (DWORD) Speicherausrichtung;
- TwinCAT 2.xx + BC/BX (Busklemmen Controller) Plattform => Datenstrukturen haben 2 Byte (WORD) Speicherausrichtung;
- TwinCAT 3.xx + PC/CX (x86, x64, ARM) Plattform => Datenstrukturen haben 8 Byte Speicherausrichtung;

Die verwendete Strukturvariablendefinition auf beiden Systemen:

```
(* 8 byte aligned structure, byte size := 152 byte *)
TYPE ST_FrameData :
STRUCT
  nFrameSize: DWORD>(*Frame byte size, member byte size := 4 byte*)
  nTxFrames : DWORD>(*Tx frame number, member byte size := 4 byte*)
  nRxFrames : DWORD>(*Rx frame number, member byte size := 4 byte*)
  nCounter  : DWORD>(*Number value, member byte size := 4 byte*)
  fU       : REAL>(*Floating point number, member byte size := 4 byte*)
  fV       : REAL>(*Floating point number, member byte size := 4 byte*)
  fW       : REAL>(*Floating point number, member byte size := 4 byte*)
  aFloats  : ARRAY[0..9] OF REAL>(* Array of floating point numbers, array byte size := 40 byte*)
  sMsg     : STRING>(*String variable, member byte size := 81 byte incl. String null delimiter*)
  bEnable  : BOOL>(*Boolean flag, member byte size := 1 byte*)
  nRsv0   : BYTE>(*Reserved byte to meet the 8 byte alignment, member byte size := 1 byte*)
  nCRC    : BYTE>(*CRC checksum byte, member byte size := 1 byte*)
END_STRUCT
END_TYPE
```

BC/BX (Busklemmen Controller) Applikation

Nach jedem Schreibzugriff vom PC/CX wird die Datenlänge und Checksumme überprüft. Danach werden neue Zufallswerte für den Lesezugriff generiert die zusätzlich mit einer einfachen Checksumme versehen werden.

```

PROGRAM MAIN
VAR
  stRxFrame AT%MB500 : ST_FrameData; (* Data transported from PC/CX (x86) to BC/
  BX (Bus Terminal Controller) *)
  stTxFrame AT%MB0 : ST_FrameData; (* Data transported from BC/
  BX (Bus Terminal Controller) to PC/CX (x86) *)
  nReceivedFrame : UDINT;
  i : INT;
  nRxErrors : UDINT;
END_VAR

(* New frame from PC/CX received? *)
IF stRxFrame.nTxFrames <> nReceivedFrame THEN
  (* Frame length OK? *)
  IF stRxFrame.nFrameSize = SIZEOF( stRxFrame) THEN
    (* Checksum OK? *)
    IF stRxFrame.nCRC = F_CheckSum( ADR( stRxFrame), SIZEOF( stRxFrame) - 1 ) THEN (* => OK *)
      (* Create/modify the tx data *)
      stTxFrame.nFrameSize := SIZEOF( stTxFrame); (* Set frame byte size *)
      stTxFrame.nTxFrames := stTxFrame.nTxFrames + 1; (* Increment the send frame number *)
      stTxFrame.nRxFrames := stRxFrame.nTxFrames; (* Report the received frame number *)
      stTxFrame.bEnable := NOT stRxFrame.bEnable; (* Toggle bool flag *)
      stTxFrame.nCounter := stTxFrame.nCounter + 1; (* Send some counter value *)
      stTxFrame.sMsg := CONCAT( 'Message from BC/
  BX, counter:', DWORD_TO_STRING( stTxFrame.nCounter ) ); (* Create any string message *)
      stTxFrame.fU := stRxFrame.fU + 10.0; (* Modify some floating point values *)
      stTxFrame.fV := stRxFrame.fV + 100.0;
      stTxFrame.fW := stRxFrame.fW + 1000.0;
      FOR i:= 0 TO 9 DO
        stTxFrame.aFloats[i] := stTxFrame.aFloats[i] + i + 3.141592;
      END_FOR
      stTxFrame.nCRC := F_CheckSum( ADR( stTxFrame), SIZEOF( stTxFrame) - 1 );
    (* Create checksum *)
    ELSE (* => Checksum error *)
      nRxErrors := nRxErrors + 1;
    END_IF
    ELSE (* => Invalid frame length *)
      nRxErrors := nRxErrors + 1;
    END_IF
    nReceivedFrame := stRxFrame.nTxFrames;
  END_IF

```

PC/CX (x86, x64, ARM) Applikation

Eine steigende Flanke am bWrite startet den Schreibvorgang. Die REAL elemente werden vor dem Schreibvorgang in den BC/BX-Format konvertiert. Die Datenlänge und Checksumme wird ermittelt und gesetzt. Eine steigende Flanke am bRead startet den Lesevorgang. Nach erfolgreichem Lesevorgang wird zuerst die Länge der Daten dann die eine einfache Checksumme überprüft. Danach werden die REAL-Elemente in das PC/CX-Format konvertiert.

```

PROGRAM MAIN
VAR
  bWrite : BOOL; (* Rising edge at this variable writes data to the BC/
  BX (Bus Terminal Controller) *)
  bRead : BOOL; (* Rising edge at this variable reads data from BC/
  BX (Bus Terminal Controller) *)
  stTxFrame : ST_FrameData; (* Data transported from PC/CX (x86) to BC/
  BX (Bus Terminal Controller) *)
  stRxFrame : ST_FrameData; (* Data transported from BC/BX (Bus Terminal Controller) to PC/
  CX (x86) *)
  fbWrite : ADWRITE := ( NETID := '172.17.61.50.1.1', PORT := 800, IDXGRP := 16#4020, IDXO
  FFS := 500, TMOUT := DEFAULT_ADS_TIMEOUT );
  fbRead : ADSREAD := ( NETID := '172.17.61.50.1.1', PORT := 800, IDXGRP := 16#4020, IDXOF
  FS := 0, TMOUT := DEFAULT_ADS_TIMEOUT );
  (* Temporary used variables *)
  stTxToBC : ST_FrameData;
  stRxFromBC : ST_FrameData;
  i : INT;
  nTxState : UDINT;
  nRxState : UDINT;
  nTxErrors : UDINT;
  nRxErrors : UDINT;
END_VAR

```

```

(******)
CASE nTxState OF
  0:
    IF bWrite THEN(* Write BC/BX data *)
      bWrite := FALSE;

      (* Prepare/modify tx data *)
      stTxFrame.nFrameSize := SIZEOF( stTxFrame );(* Set frame byte size *)
      stTxFrame.nTxFrames := stTxFrame.nTxFrames + 1;(* Increment the send frame number *)
      stTxFrame.nRxFrames := stRxFrame.nRxFrames;(* Report the received frame number *)
      stTxFrame.bEnable := NOT stTxFrame.bEnable;(* Toggle bool flag *)
      stTxFrame.nCounter := stTxFrame.nCounter + 1;(* Increment counter value *)
      stTxFrame.sMsg := CONCAT( 'Message from PC/
CX, counter: ', DWORD_TO_STRING( stTxFrame.nCounter ) );(* Create some string message *)
      stTxFrame.fU := stTxFrame.fU + 1.2;(* Modify some floating point values *)
      stTxFrame.fV := stTxFrame.fV + 3.4;
      stTxFrame.fW := stTxFrame.fW + 5.6;
      FOR i:= 0 TO 9 DO
        stTxFrame.aFloats[i] := stTxFrame.aFloats[i] + i;
      END_FOR
      stTxFrame.nCRC := 0;

      (* Create temporary copy of tx data *)
      stTxToBC := stTxFrame;

      (* Swap REAL variables to BC/BX (Bus Terminal Controller) format *)
      F_SwapRealEx( stTxToBC .fU );
      F_SwapRealEx( stTxToBC .fV );
      F_SwapRealEx( stTxToBC .fW );
      FOR i:= 0 TO 9 DO
        F_SwapRealEx( stTxToBC .aFloats[i] );
      END_FOR

      (* Create CRC check number *)
      stTxToBC .nCRC := F_CheckSum( ADR( stTxToBC ), SIZEOF( stTxToBC ) - 1 );

      (* Send *)
      fbWrite( WRITE := FALSE );
      fbWrite( LEN := SIZEOF( stTxToBC ), SRCADDR := ADR( stTxToBC ), WRITE := TRUE );
      nTxState := 1;
    END_IF

  1:(* Wait until ads write command not busy *)
    fbWrite( WRITE := FALSE );
    IF NOT fbWrite.BUSY THEN
      IF NOT fbWrite.ERR THEN
        nTxState := 0;
      ELSE(* Ads error *)
        nTxState := 100;
      END_IF
    END_IF

  100:(* TODO: Error state, add error handling *)
    nTxErrors := nTxErrors + 1;
    nTxState := 0;
END_CASE

(******)
CASE nRxState OF
  0:
    IF bRead THEN(* Read BC/BX data *)
      bRead := FALSE;

      fbRead( READ := FALSE );
      fbRead( LEN := SIZEOF( stRxFromBC ), DESTADDR := ADR( stRxFromBC ), READ := TRUE );
      nRxState := 1;
    END_IF

  1:(* Wait until ads read command not busy *)
    fbRead( READ := FALSE );
    IF NOT fbRead.BUSY THEN
      IF NOT fbRead.ERR THEN
        (* Perform simple frame length check *)
        IF stRxFromBC.nFrameSize = SIZEOF( stRxFromBC ) THEN (* Check frame length *)
          (* Perform simple CRC check *)
          IF stRxFromBC.nCRC = F_CheckSum( ADR( stRxFromBC ), SIZEOF( stRxFromBC ) -
1 ) THEN

          (* Swap REAL variables to PC/CX (x86) format *)

```

```

F_SwapRealEx( stRxFromBC.fU );
F_SwapRealEx( stRxFromBC.fV );
F_SwapRealEx( stRxFromBC.fW );
FOR i:= 0 TO 9 DO
    F_SwapRealEx( stRxFromBC.aFloats[i] );
END_FOR

stRxFrame := stRxFromBC;
nRxState  := 0;

ELSE(* => Checksum error *)
    nRxState := 100;
END_IF
ELSE(* => Invalid frame length *)
    nRxState := 100;
END_IF
ELSE(* => Ads error *)
    nRxState := 100;
END_IF
END_IF

100: (* TODO: Error state, add error handling *)
    nRxErrors := nRxErrors + 1;
    nRxState := 0;

END_CASE

```

Applikationstest

Öffnen Sie die PC/CX-Applikation und schreiben TRUE in die bWrite-Variable. Im nächsten Schritt schreiben Sie TRUE in die bRead-Variable.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

8.2 Beispiel: Dateisuche (FB_EnumFindFileEntry, FB_EnumFindFileList)

Hier können Sie die kompletten Sourcen entpacken: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_Utilities/Resources/803340811.zip

Beispiel: FB_EnumFindFileEntry (ST)

Auf dem lokalen TwinCAT System sollen alle Dateien im Verzeichnis C:\Windows\System32\ aufgelistet werden. Die Dateinamen sollen als Meldungen ins TwinCAT XAE Error Liste geschrieben werden. Es soll möglich sein diesen Vorgang abubrechen. Bei einer steigenden Flanke an der *bEnum*-Variablen beginnt die Auflistung der gefundenen Dateien. Bei einer steigenden Flanke an der *bAbort*-Variablen wird der Vorgang abgebrochen.

```

PROGRAM P_TestEnumEntry
VAR
    fbEnum: FB_EnumFindFileEntry := ( sNetID := '', tTimeout := T#5s, sPathName := 'C:\Windows\System32\*.*' );
    bEnum : BOOL;
    bAbort: BOOL;
    nState: BYTE;
END_VAR

CASE nState OF
    0:
        IF bEnum THEN
            (* flag set ? *)
            bEnum := FALSE; (* reset flag *)
            fbEnum.eCmd := eEnumCmd_First; (* enum first entry *)
            nState := 1;
        END_IF
    1: (* enum one entry *)

```

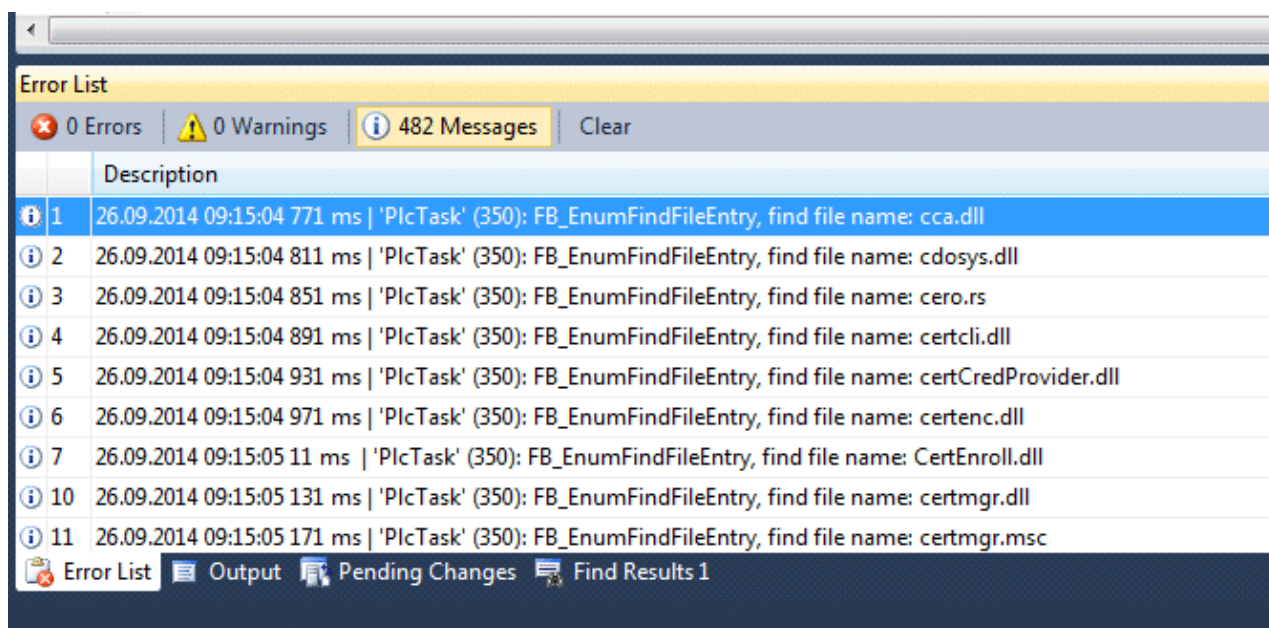
```

IF bAbort THEN
    bAbort      := FALSE;
    fbEnum.eCmd := eEnumCmd_Abort;
END_IF
fbEnum( bExecute := FALSE );
fbEnum( bExecute := TRUE );
nState      := 2;

2:    (* wait until function block not busy *)
fbEnum( bExecute := FALSE );
IF NOT fbEnum.bBusy THEN
    IF NOT fbEnum.bError THEN
        IF NOT fbEnum.bEOE THEN
            ADSLOGSTR( ADSLOG_MSGTYPE_HINT OR ADSLOG_MSGTYPE_LOG, 'FB_EnumFindFileEntry, find
d file name: %s', fbEnum.stFindFile.sFileName );
            fbEnum.eCmd := eEnumCmd_Next; (* enum next entry *)
            nState      := 1;
        ELSE (* no more entries *)
            nState      := 0;
        END_IF
    ELSE (* log error *)
        ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'FB_EnumFindFileEntry error:
%s', DWORD_TO_HEXSTR( fbEnum.nErrID, 0, FALSE ) );
        nState      := 0;
    END_IF
END_IF
END_CASE

```

Die geschriebenen Logmeldungen in der TwinCAT XAE Error Liste:



Beispiel: FB_EnumFindFileList (FUP)

Bei einer steigenden Flanke an der *bFirst*-Variablen wird der Funktionsbaustein aktiviert. Beim Erfolg werden die Dateinamen in die *fileList*-Arrayvariable eingetragen.

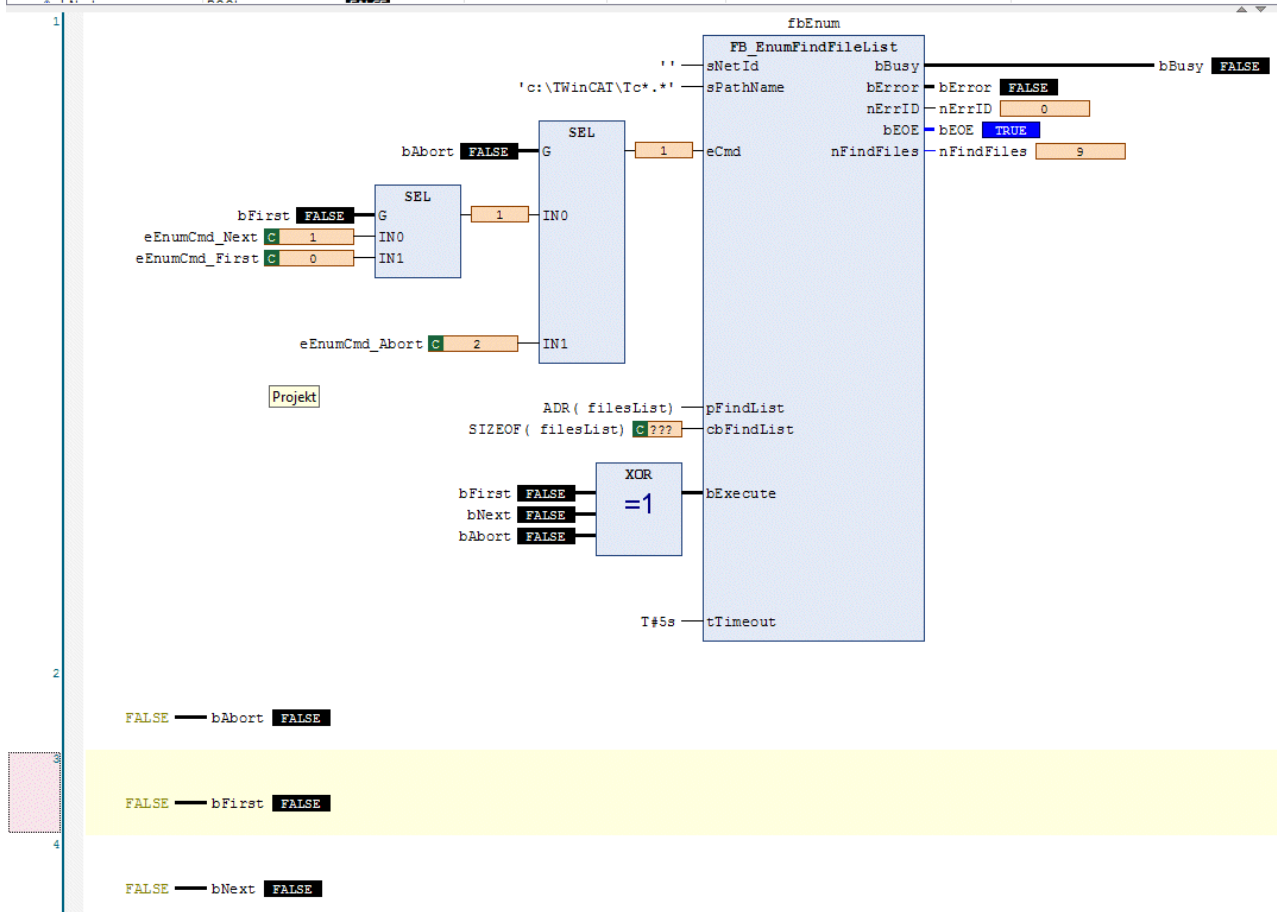
```

PROGRAM P_TestEnumList
VAR
    fbEnum      : FB_EnumFindFileList;
    fileList    : ARRAY[1..10] OF ST_FindFileEntry;
    bFirst      : BOOL;
    bNext       : BOOL;
    bAbort      : BOOL;
    bBusy       : BOOL;
    bError      : BOOL;
    nErrID      : UDINT;
    bEOE        : BOOL;
    nFindFiles  : UDINT;
END_VAR

```

Onlineansicht:

Ausdruck	Datentyp	Wert	Vorbereiteter Wert	Adresse	Kommentar
fbEnum	FB_EnumFindFileList				
filesList	ARRAY [1..10] OF S...				
filesList[1]	ST_FindFileEntry				
sFileName	STRING(255)	TcAmsSerial.dll'			
sAlternateFile...	STRING(13)	'TCAMSS~1.DLL'			
fileAttributes	ST_FileAttributes				
fileSize	T_ULARGE_INTEGER				
creationTime	T_FILETIME				
lastAccessTime	T_FILETIME				
lastWriteTime	T_FILETIME				
filesList[2]	ST_FindFileEntry				
filesList[3]	ST_FindFileEntry				
filesList[4]	ST_FindFileEntry				
filesList[5]	ST_FindFileEntry				
filesList[6]	ST_FindFileEntry				
filesList[7]	ST_FindFileEntry				
filesList[8]	ST_FindFileEntry				
filesList[9]	ST_FindFileEntry				
filesList[10]	ST_FindFileEntry				
bFirst	BOOL	FALSE			



Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

8.3 Beispiel: Datei-Ring-FiFo (FB_FileRingBuffer)

Die kompletten Quellen finden Sie hier: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_Utilities/Resources/803395851.zip

Folgendes Beispiel zeigt eine einfache Verwendung des Funktionsbausteins. Die steigende Flanke am *bOpen* öffnet eine existierende Ringpufferdatei. Wenn die Datei nicht existiert, wird eine neue erzeugt. Die steigende Flanke am *bClose* schließt eine geöffnete Datei. Die steigende Flanke am *bCreate* erzeugt eine neue Datei. Wenn Sie *bAdd* = TRUE setzen wird ein neuer Datensatz in die Ringpufferdatei geschrieben und beim *bRemove*=TRUE wird der älteste Datensatz entfernt.

```

PROGRAM MAIN
VAR
  bOpen      : BOOL;
  bClose     : BOOL;
  bCreate    : BOOL;
  bAdd       : BOOL;
  bRemove    : BOOL;
  bGet       : BOOL;
  bReset     : BOOL;

  fbFileBuffer : FB_FileRingBuffer := ( sNetId      := '',
                                        sPathName   := 'c:\temp\Data.dat',
                                        ePath       := PATH_GENERIC,
                                        nID         := 1,
                                        cbBuffer    := 100, (*cbBuffer := 16#80000000, 2GB*)
                                        bOverwrite  := TRUE,
                                        pWriteBuff  := 0,
                                        cbWriteLen  := 0,
                                        pReadBuff   := 0,
                                        cbReadLen   := 0,
                                        tTimeout    := t#5s );

  storeData  : ARRAY[1..10] OF BYTE :=[10(0)];
  cbStore    : UDINT;
  loadData   : ARRAY[1..10] OF BYTE :=[10(0)];
  cbLoad     : UDINT;
  i          : INT;
END_VAR

fbFileBuffer( cbReturn => cbLoad );

IF NOT fbFileBuffer.bBusy THEN

  IF bOpen THEN
    bOpen := FALSE;
    fbFileBuffer.A_Open();
  END_IF

  IF bClose THEN
    bClose := FALSE;
    fbFileBuffer.A_Close();
  END_IF

  IF bCreate THEN
    bCreate := FALSE;
    fbFileBuffer.A_Create();
  END_IF

  IF bAdd THEN
    bAdd := FALSE;

    (* modify data *)
    FOR i:=1 TO 10 BY 1 DO
      storeData[i] := storeData[i] + 1;
    END_FOR

    cbStore := SEL( cbStore > 1, SIZEOF(storeData), cbStore -
1 ); (* modify the data chunk length *)
    fbFileBuffer.A_AddTail( pWriteBuff := ADR(storeData), cbWriteLen := cbStore,
                           pReadBuff := 0, cbReadLen:=0 );
  END_IF

  IF bRemove THEN
    bRemove := FALSE;
    fbFileBuffer.A_RemoveHead( pWriteBuff := 0, cbWriteLen := 0,
                               pReadBuff := ADR(loadData), cbReadLen := SIZEOF(loadData));
  END_IF

  IF bGet THEN
    bGet := FALSE;
    fbFileBuffer.A_GetHead( pWriteBuff := 0, cbWriteLen := 0,
                            pReadBuff := ADR(loadData), cbReadLen := SIZEOF(loadData));
  END_IF

```

```

END_IF

IF bReset THEN
    bReset := FALSE;
    fbFileBuffer.A_Reset();
END_IF

END_IF

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

8.4 Beispiel: Memory-Ring-FiFo (FB_MemRingBuffer)

Die kompletten Sourcen finden Sie hier: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_Utilities/Resources/803399691.zip

Folgendes Beispiel zeigt eine einfache Verwendung des Funktionsbausteins. Es sollen Datensätze mit der gleichen Länge gepuffert werden (dies ist aber nicht zwingend notwendig). Die Datensätze haben folgende Struktur:

```

TYPE ST_DataSetEntry :
STRUCT
    bFlag : BOOL;
    nValue : BYTE;
    sMsg : STRING(20) := 'Unknown';
END_STRUCT
END_TYPE

```

Die Schnittstelle des FB_DataSetFifo-Funktionsbausteins:

Der im Beispielprojekt verwendete applikationsspezifische Funktionsbaustein FB_DataSetFifo benutzt intern den FB_MemRingBuffer-Funktionsbaustein. Dieser Baustein vereinfacht das Hinzufügen/Entfernen der Datensätze. Außerdem liefert der neue Funktionsbaustein den aktuellen prozentualen Füllstatus des Puffers und eine Overwrite-Option. Wenn *bOverwrite*-Eingang gesetzt ist und der Puffer bereits voll ist, dann wird der älteste Eintrag aus dem Puffer entfernt und mit dem neuen überschrieben.

```

VAR_GLOBAL CONSTANT
    MAX_BUFFER_SIZE : UDINT := 1000;
END_VAR

FUNCTION_BLOCK FB_DataSetFifo
VAR_INPUT
    bOverwrite : BOOL;
    in : ST_DataSetEntry;
END_VAR
VAR_OUTPUT
    bOk : BOOL;
    nCount : UDINT;
    nLoad : UDINT;
    out : ST_DataSetEntry;
END_VAR
VAR
    arrBuffer : ARRAY[0..MAX_BUFFER_SIZE] OF BYTE; (* Buffer memory used by FB_MemRingBuffer function block *)
    fbBuffer : FB_MemRingBuffer;
END_VAR

```

Das Hauptprogramm:

Die steigende Flanke am *bReset* löscht alle Puffer Einträge. Wenn Sie *bAdd* = TRUE setzen wird ein neuer Datensatz in den Ringpuffer geschrieben und beim *bRemove*=TRUE wird der älteste Datensatz entfernt. Bei einer steigenden Flanke am *bGet* wird der älteste Datensatz gelesen aber nicht entfernt.

```

PROGRAM MAIN
VAR
    fbFifo : FB_DataSetFifo := ( bOverwrite := TRUE );
    newEntry : ST_DataSetEntry;
    oldEntry : ST_DataSetEntry;

```

```

    bSuccess      : BOOL;
    nCount       : UDINT;
    nLoad        : UDINT;

    bReset       : BOOL := TRUE;
    bAdd         : BOOL := TRUE;
    bGet         : BOOL := TRUE;
    bRemove      : BOOL := TRUE;
END_VAR

IF bReset THEN
    bReset := FALSE;
    (* reset fifo (clear all entries) *)
    fbFifo.A_Reset( in := newEntry, bOk=>bSuccess, nCount=> nCount, nLoad => nLoad );
END_IF

IF bAdd THEN
    bAdd := FALSE;

    (* create new or modify data set entry *)
    newEntry.bFlag := NOT newEntry.bFlag;
    newEntry.nValue := newEntry.nValue + 1;
    newEntry.sMsg := BYTE_TO_STRING(newEntry.nValue);

    (* add new entry to the fifo *)
    fbFifo.A_Add( in := newEntry, bOk=>bSuccess, nCount=> nCount, nLoad => nLoad );
END_IF

IF bGet THEN
    bGet := FALSE;
    (* get (but not delete) oldest entry *)
    fbFifo.A_Get( out => oldEntry, bOk => bSuccess, nCount => nCount, nLoad => nLoad );
END_IF

IF bRemove THEN
    bRemove:= FALSE;
    (* remove oldest entry *)
    fbFifo.A_Remove( out => oldEntry, bOk => bSuccess, nCount => nCount, nLoad => nLoad );
END_IF

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

8.5 Beispiel: Memory-Ring-FiFo (FB_MemRingBufferEx)

Hier können Sie die kompletten Sourcen entpacken: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_Uilities/Resources/803404811.zip

Bei einer steigenden Flanke am bAdd werden neue Datenelemente (pubObj-Array) in den Ringpuffer abgelegt. Dach kann über eine steigende Flanke am bGet das älteste Datenelement in die getObj-Variable kopiert werden.

Die nicht benötigten Datenelemente werden über eine steigende Flanke am bRelease aus dem Puffer entfernt.

```

PROGRAM MAIN
VAR
    bReset      : BOOL := TRUE;
    bAdd, bGet, bRelease, bGetFree : BOOL;
    putObj      : ARRAY[0..3] OF BYTE :=[ 16#00, 16#AA, 16#BB, 16#CC];
    getObj      : ARRAY[0..3] OF BYTE :=[ 4(0)];
    bOk         : BOOL;
    nCount      : UDINT;
    cbSize      : UDINT;
    cbFree      : UDINT;
    fbBuffer: FB MemRingBufferEx;
    buffer      : ARRAY[0..30] OF BYTE;
END_VAR

IF bReset THEN
    bReset := FALSE;
    fbBuffer.A_Reset( pBuffer := ADR( buffer ), cbBuffer := SIZEOF( buffer ),

```

```

                bOk=>bOk,nCount=>nCount, cbSize=>cbSize, cbFree=>cbFree );
END_IF

IF bAdd THEN
    bAdd := FALSE;
    putObj[0] := putObj[0] + 1;(* modify data *)

    fbBuffer.A_AddTail( pBuffer := ADR( buffer ), cbBuffer := SIZEOF( buffer ),
                        pWrite := ADR( putObj ), cbWrite := SIZEOF( putObj ),
                        bOk=>bOk, nCount=>nCount, cbSize=>cbSize, cbFree=>cbFree );
    IF fbBuffer.bOk THEN
        ;(* Success *)
    ELSE
        ;(* Buffer overflow *)
    END_IF
END_IF

IF bGet THEN
    bGet := FALSE;
    fbBuffer.A_GetHead( pBuffer := ADR( buffer ), cbBuffer := SIZEOF( buffer ),
                       bOk=>bOk, nCount=>nCount, cbSize=>cbSize, cbFree=>cbFree );
    IF fbBuffer.bOk THEN
        (* Success *)
        MEMCPY( ADR( getObj ), fbBuffer.pRead, MIN( SIZEOF( getObj ), fbBuffer.cbRead ) );
    ELSE
        ;(* Buffer empty *)
    END_IF
END_IF

IF bRelease THEN
    bRelease := FALSE;
    fbBuffer.A_FreeHead( pBuffer := ADR( buffer ), cbBuffer := SIZEOF( buffer ),
                        bOk=>bOk, nCount=>nCount, cbSize=>cbSize, cbFree=>cbFree );
    IF fbBuffer.bOk THEN
        ;(* Success *)
    ELSE
        ;(* Buffer empty *)
    END_IF
END_IF

IF bGetFree THEN
    bGetFree := FALSE;
    fbBuffer.A_GetFreeSize( pBuffer := ADR( buffer ), cbBuffer := SIZEOF( buffer ),
                           bOk=>bOk, nCount=>nCount, cbSize=>cbSize, cbFree=>cbFree );
END_IF

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uutilities (System)

8.6 Beispiel: Hash-Tabelle (FB_HashTableCtrl)

Hier können Sie die kompletten Sourcen entpacken: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_Uutilities/Resources/803421707.zip

Das Beispielprojekt besitzt zwei Programmteile:

- P_TABLE_OF_UDINT ist ein einfaches Beispielprogramm welches nur 32 bit Werte in der Hash-Tabelle bearbeitet.
- Im P_TABLE_OF_STRUCTDATA wird gezeigt wie andere Datentypen (z.B. strukturierte Datentypen) in der Hash-Tabelle bearbeitet werden können.

Die maximale Anzahl der Tabellenelemente kann zur Laufzeit nicht verändert werden und wird in dem Beispielprojekt durch die MAX_DATA_ELEMENTS begrenzt. Wenn Sie mehr Elemente benötigen, dann müssen Sie das table-Array entsprechend vergrößern (d.h. den Wert der Konstante erhöhen).

```

VAR_GLOBAL CONSTANT
  MAX_DATA_ELEMENTS : UDINT := 100; (* Max. number of elements in the list *)
  MAX_NAME_LENGTH   : UDINT := 30; (* Max. length of article name *)
END_VAR

```

PROGRAM P_TABLE_OF_UDINT

Im ersten SPS-Zyklus werden in der Tabelle als Beispiel Informationen über Artikelnummer und Artikelname abgelegt. Die Artikelnummer dient als Schlüssel und der Array-Index des Artikelnamens als Wert. Über eine steigende Flanke am bLookup kann der Artikelname anhand der Artikelnummer gefunden werden.

```

PROGRAM P_TABLE_OF_UDINT
VAR
  sInfo      : T_MaxString := '';
  bAdd       : BOOL := TRUE;
  bLookup    : BOOL := TRUE;
  bRemove    : BOOL := TRUE;
  bEnum      : BOOL := TRUE;
  bCount     : BOOL := TRUE;

  search     : UDINT := 11111; (* article number *)

  fbTable    : FB_HashTableCtrl; (* basic hash table control function block *)
  hTable     : T_HHASHTABLE; (* hash table handle *)
  table      : ARRAY[0..MAX_DATA_ELEMENTS] OF T_HashTableEntry;
  (* Max. number of hash table entries. The value of hash table entry = 32 bit integer *)
  names      : ARRAY[0..MAX_DATA_ELEMENTS] OF STRING(MAX_NAME_LENGTH);
  bInit      : BOOL := TRUE;
END_VAR

IF bInit THEN
  bInit := FALSE;
  F_CreateHashTableHnd( ADR( table ), SIZEOF( table ), hTable ); (* Intialize table handle *)
END_IF

IF bAdd THEN
  bAdd := FALSE;

  (* Fill table. Article number is the key. Array index number is the value (article name) *)
  names[0] := 'Chair';
  fbTable.A_Add( key := 12345, putValue := 0 (* array index*), hTable := hTable );
  IF NOT fbTable.bOk THEN
    ; (* Table overflow *)
  END_IF

  names[1] := 'Table';
  fbTable.A_Add( key := 67890, putValue := 1, hTable := hTable );
  IF NOT fbTable.bOk THEN
    ; (* Table overflow *)
  END_IF

  names[2] := 'Couch';
  fbTable.A_Add( key := 11111, putValue := 2, hTable := hTable );
  IF NOT fbTable.bOk THEN
    ; (* Table overflow *)
  END_IF

  names[3] := 'TV set';
  fbTable.A_Add( key := 22222, putValue := 3, hTable := hTable );
  IF NOT fbTable.bOk THEN
    ; (* Table overflow *)
  END_IF
END_IF

IF bLookup THEN (* search for the article name by article number *)
  bLookup := FALSE;
  sInfo := '';
  fbTable.A_Lookup( key := search, hTable := hTable );
  IF fbTable.bOk THEN
    sInfo := names[fbTable.getValue];
  ELSE
    ; (* Entry not found *)
  END_IF
END_IF

IF bRemove THEN (* remove one entry from the table *)
  bRemove := FALSE;
  sInfo := '';

```

```

fbTable.A_Remove( key := search, hTable := hTable );
IF fbTable.bOk THEN
    sInfo := names[fbTable.getValue];
ELSE
    ;(* Entry not found *)
END_IF
END_IF

IF bEnum THEN(* enumerate table entries *)
    bEnum := FALSE;
    sInfo := '';

    fbTable.A_GetFirst( putPosPtr := 0, hTable := hTable );
    IF fbTable.bOk THEN
        sInfo := names[fbTable.getValue];

        REPEAT
            fbTable.A_GetNext( putPosPtr := fbTable.getPosPtr , hTable := hTable );
            IF fbTable.bOk THEN
                sInfo := names[fbTable.getValue];
            END_IF
        UNTIL NOT fbTable.bOk
        END_REPEAT

    END_IF
END_IF

IF bCount THEN(* count entries in the table *)
    bCount := FALSE;
    sInfo := UDINT_TO_STRING( hTable.nCount );
END_IF

```

PROGRAM P_TABLE_OF_STRUCTDATA

Dieses Programmteil zeigt wie strukturierte Datensätze statt einfacher 32-Zahlen in der Tabelle bearbeitet werden können. Dabei wird der 32 bit Elementwert nur als Referenzpointer auf den tatsächlichen Elementwert verwendet. Der Referenzpointer kann dann auch auf Instanzen von strukturierten Variablen oder anderen Datentypen zeigen. Die Funktionalität wurde in einem Funktionsblock gekapselt. Dieser Funktionsblock *FB_SpecialHashTableCtrl* kann als eine spezialisierte Version des *FB_HashTableCtrl*-Funktionsbausteins bezeichnet werden. Der *FB_HashTableCtrl*-Baustein wird auch intern von dem spezialisierten FB verwendet.

Die Funktion *DATAELEMENT_TO_STRING* wird nur verwendet um eine visuelle Ausgabe der Knotenwerte zu ermöglichen.

Als Beispiel wird die Strukturierte Variable vom Typ: *ST_DataElement* verwendet. Der Clou: Sie können die Datentypdeklaration von *ST_DataElement* um weitere Membervariablen erweitern, ohne dass Sie an dem Programm oder dem *FB_SpecialHashTableCtrl*-Funktionsbaustein Veränderungen durchführen müssen.

Die Typdeklaration von *ST_DataElement*:

```

TYPE ST_DataElement :(* Structured application data entry *)
STRUCT
    (* Adapt this structure to match your application needs *)
    number      : UDINT := 0;
    name        : STRING(MAX_NAME_LENGTH) := '';
    price       : REAL := 0.0;
END_STRUCT
END_TYPE

```

Wie werden die 32 bit Elementwerte zu Referenzpointern auf die ST_DataElement-Array-Instanzen?

Die max. Größe der Tabelle ist durch *MAX_DATA_ELEMENTS*-Konstante begrenzt. Folglich können in der Tabelle nur *MAX_DATA_ELEMENTS*-Referenzpointer gespeichert werden. Der *FB_SpecialHashTableCtrl*-Baustein besitzt intern eine *ST_DataElement*-Array-Variable mit derselben Array-Größe wie die *T_HashTableEntry*-Array-Variable. Zur Vereinfachung sind die Array-Indizes in beiden Arrays gleich!

Jedes *T_HashTableEntry*-Array-Element kann nur einmal in die Tabelle "reingehängt" werden. Dabei sucht der *FB_HashTableCtrl*-Funktionsbaustein nach einem freien/unbenutzten *T_HashTableEntry*-Array-Element und fügt es bei Erfolg der Tabelle zu. Mit Hilfe der Aktion *A_GetIndexAtPosPtr* kann der verwendete Index

des T_HashTableEntry-Arrays ermittelt werden. Im nächsten Schritt wird dem zuletzt hinzugefügten 32 bit Knotenwert die Adresse desselben Arrayelements vom ST_DataElement-Array zugewiesen. Im Beispielprojekt durch den zweiten Aktionsaufruf: *A_Add*.

nodes[index].value := ADR(dataPool[index])

Die Zuweisung wird z.B. in der *FB_SpecialHashTableCtrl->A_Add*-Aktion durchgeführt:

```
(* Adds entry to the table *)
MEMSET( ADR( getValue ), 0, SIZEOF( getValue ) );
getPosPtr := 0;

fbTable.A_Add( hTable := hTable, key := key, putValue := 16#00000000(* we will set this value later
*), getPosPtr=>getPosPtr, bOk=>bOk );
(* Add new element to the table, getPosPtr points to the new entry *)
IF fbTable.bOk THEN(* Success *)
  fbTable.A_GetIndexAtPosPtr( hTable := hTable, putPosPtr := getPosPtr, getValue =>indexOfElem, bO
k=>bOk );(* Get array index of getPosPtr entry *)
  IF fbTable.bOk THEN(* Success *)
    pRefPtr      := ADR( dataPool[indexOfElem] );(* Get pointer to the data element *)

    pRefPtr^ := putValue;(* copy application value *)

    fbTable.A_Add( hTable := hTable, key := key, putValue := pRefPtr, bOk=>bOk );
(* Assign the entry value = pointer to the data element *)
  IF fbTable.bOk THEN(* Success *)
    getValue := putValue;
  END_IF
END_IF

PROGRAM P_TABLE_OF_STRUCTDATA
VAR
  sInfo      : T_MaxString := '';
  bAdd       : BOOL := TRUE;
  bLookup    : BOOL := TRUE;
  bRemove    : BOOL := TRUE;
  bEnum      : BOOL := TRUE;
  bCount     : BOOL := TRUE;

  search     : UDINT := 11111;(* article number *)

  fbTable    : FB_SpecialHashTableCtrl;(* Specialized hash table control function block *)
  putValue   : ST_DataElement;
  getValue   : ST_DataElement;
  getPosPtr  : POINTER TO T_HashTableEntry := 0;
  bInit      : BOOL := TRUE;
END_VAR

IF bInit THEN
  bInit      := FALSE;
  fbTable.A_Reset();(* reset / initialize table *)
END_IF

IF bAdd THEN
  bAdd := FALSE;

  (* Fill table. Article number is the key and data structure is the value *)
  putValue.number := 12345;
  putValue.name := 'Chair';
  putValue.price := 44.98;
  fbTable.A_Add( key := 12345, putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
  IF NOT fbTable.bOk THEN
    ;(* Table overflow *)
  END_IF

  putValue.number := 67890;
  putValue.name := 'Table';
  putValue.price := 99.98;
  fbTable.A_Add( key := 67890, putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
  IF NOT fbTable.bOk THEN
    ;(* Table overflow *)
  END_IF

  putValue.number := 11111;
  putValue.name := 'Couch';
  putValue.price := 99.98;
  fbTable.A_Add( key := 11111, putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
  IF NOT fbTable.bOk THEN
```

```

        ;(* Table overflow *)
    END_IF

    putValue.number := 22222;
    putValue.name := 'TV set';
    putValue.price := 99.98;
    fbTable.A_Add( key := 22222, putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbTable.bOk THEN
        ;(* Table overflow *)
    END_IF

END_IF

IF bLookup THEN(* search for the article name by article number *)
    bLookup := FALSE;
    sInfo := '';
    fbTable.A_Lookup( key := search, getPosPtr=>getPosPtr, getValue=>getValue );
    IF fbTable.bOk THEN
        sInfo := DATAELEMENT_TO_STRING( getValue );
    ELSE
        ;(* Entry not found *)
    END_IF
END_IF

IF bRemove THEN(* remove one entry from the table *)
    bRemove := FALSE;
    sInfo := '';
    fbTable.A_Remove( key := search, getPosPtr=>getPosPtr, getValue=>getValue );
    IF fbTable.bOk THEN
        sInfo := DATAELEMENT_TO_STRING( getValue );
    ELSE
        ;(* Entry not found *)
    END_IF
END_IF

IF bEnum THEN(* enumerate table entries *)
    bEnum := FALSE;
    sInfo := '';

    fbTable.A_GetFirst( putPosPtr := 0, getPosPtr=>getPosPtr, getValue=>getValue );
    IF fbTable.bOk THEN
        sInfo := DATAELEMENT_TO_STRING( getValue );

        REPEAT
            fbTable.A_GetNext( putPosPtr := fbTable.getPosPtr , getPosPtr=>getPosPtr, getValue=>getV
            alue );
            IF fbTable.bOk THEN
                sInfo := DATAELEMENT_TO_STRING( getValue );
            END_IF
        UNTIL NOT fbTable.bOk
        END_REPEAT

    END_IF
END_IF

IF bCount THEN(* count entries in the table *)
    bCount := FALSE;
    fbTable.A_Count();
    IF fbTable.bOk THEN
        sInfo := UDINT_TO_STRING( fbTable.nCount );
    END_IF
END_IF

```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

8.7 Beispiel: Verkettete-Liste (FB_LinkedListCtrl)

Hier können Sie die kompletten Quellen entpacken: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_Utilities/Resources/803425547.zip

Das Beispielprojekt besitzt zwei Programmteile:

- P_LIST_OF_UDINT ist ein einfaches Beispielprogramm welches nur 32 bit Werte in der verketteten Liste bearbeitet.
- Im P_LIST_OF_STRUCTDATA wird gezeigt wie andere Datentypen (z.B. strukturierte Datentypen) in der verketteten Liste bearbeitet werden können.

Die maximale Anzahl der Knotenelemente kann zur Laufzeit nicht verändert werden und wird in dem Beispielprojekt durch die MAX_DATA_ELEMENTS begrenzt. Wenn Sie mehr Knoten benötigen, dann müssen Sie das nodes-Array entsprechend vergrößern (d.h. den Wert der Konstante erhöhen).

```
VAR_GLOBAL CONSTANT
    MAX_DATA_ELEMENTS : UDINT := 100; (* Max. number of elements in the list *)
    MAX_NAME_LENGTH   : UDINT := 30; (* Max. length of article name *)
END_VAR
```

PROGRAM P_LIST_OF_UDINT

Im ersten SPS-Zyklus wird das Handle der verketteten Liste initialisiert. Dieses Handle wird dann bei den Zugriffen auf die Liste als VAR_IN_OUT-Variable an den FB_LinkedListCtrl-Funktionsbaustein übergeben. Die verkettete Liste wird über die Aktionsaufrufe des FBs manipuliert. Auf diese Weise können Knotenelemente hinzugefügt/entfernt/durchsucht werden. Bei einer steigenden Flanke an der entsprechenden booleschen Variable wird die gewünschte Aktion ausgeführt. Wenn Sie das Programm starten werden alle Operationen einmalig ausgeführt.

```
PROGRAM P_LIST_OF_UDINT
VAR
    sInfo           : T_MaxString := '';
    bAddTailValue   : BOOL := TRUE;
    bAddHeadValue   : BOOL := TRUE;
    bGetTail        : BOOL := TRUE;
    bGetHead        : BOOL := TRUE;
    bFind           : BOOL := TRUE;
    bRemoveHeadValue : BOOL := TRUE;
    bRemoveTailValue : BOOL := TRUE;
    bCount          : BOOL := TRUE;

    search          : UDINT := 12345;

    fbList          : FB_LinkedListCtrl; (* basic linked list control function block *)
    hList           : T_HLINKEDLIST; (* linked list handle *)
    nodes           : ARRAY[0..MAX_DATA_ELEMENTS] OF T_LinkedListEntry;
    (* Max. number of linked list nodes. The value of list node = 32 bit integer *)
    putValue        : PVOID; (* Pointer or integer value (x86=>32bit, x64=>64bit) *)
    getValue        : PVOID; (* Pointer or integer value (x86=>32bit, x64=>64bit) *)
    getPosPtr       : POINTER TO T_LinkedListEntry := 0;
    bInit           : BOOL := TRUE;
END_VAR

IF bInit THEN
    bInit := FALSE;
    F_CreateLinkedListHnd( ADR( nodes ), SIZEOF( nodes ), hList );
END_IF

IF bAddTailValue THEN(* add some nodes to the list *)
    bAddTailValue := FALSE;

    putValue := 22222;
    fbList.A_AddTailValue( hList := hList, putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    putValue := 11111;
    fbList.A_AddTailValue( hList := hList, putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    putValue := 12345;
    fbList.A_AddTailValue( hList := hList, putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    putValue := 67890;
    fbList.A_AddTailValue( hList := hList, putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
END_IF

IF bAddHeadValue THEN
    bAddHeadValue := FALSE;

    putValue := 33333;
    fbList.A_AddHeadValue( hList := hList, putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    putValue := 44444;
```

```

    fbList.A_AddHeadValue( hList := hList, putValue := putValue, getPosPtr=>getPosPtr, getValue=>get
Value );
END_IF

IF bGetTail THEN(* enumerate all nodes in list (start at tail node) *)
    bGetTail := FALSE;
    sInfo := '';

    fbList.A_GetTail( hList := hList, getValue=>getValue, getPosPtr=>getPosPtr );
    IF fbList.bOk THEN
        sInfo := PVOID_TO_STRING( getValue );

        REPEAT
            fbList.A_GetPrev( hList := hList, putPosPtr := getPosPtr, getValue=>getValue, getPosPtr=
>getPosPtr );
            IF fbList.bOk THEN
                sInfo := PVOID_TO_STRING( getValue );
            ELSE
                EXIT;
            END_IF
            UNTIL NOT fbList.bOk
        END_REPEAT
    END_IF
END_IF

IF bGetHead THEN(* enumerate all nodes in list (start at head node) *)
    bGetHead := FALSE;
    sInfo := '';

    fbList.A_GetHead( hList := hList, getValue=>getValue, getPosPtr=>getPosPtr );
    IF fbList.bOk THEN
        sInfo := PVOID_TO_STRING( getValue );

        REPEAT
            fbList.A_GetNext( hList := hList, putPosPtr := getPosPtr, getValue=>getValue, getPosPtr=
>getPosPtr );
            IF fbList.bOk THEN
                sInfo := PVOID_TO_STRING( getValue );
            ELSE
                EXIT;
            END_IF
            UNTIL NOT fbList.bOk
        END_REPEAT
    END_IF
END_IF

IF bFind THEN(* search for node in the list by node value*)
    bFind := FALSE;
    getPosPtr := 0;(* start from first node element *)
    sInfo := '';

    REPEAT
        fbList.A_FindNext( hList := hList, putPosPtr := getPosPtr, putValue := search, getValue=>get
Value, getPosPtr=>getPosPtr );
        IF fbList.bOk THEN
            sInfo := PVOID_TO_STRING( getValue );
        ELSE
            EXIT;
        END_IF
        UNTIL NOT fbList.bOk
    END_REPEAT
END_IF

IF bRemoveTailValue THEN(* remove tail node from node list *)
    bRemoveTailValue := FALSE;
    sInfo := '';
    fbList.A_RemoveTailValue( hList := hList, getValue=>getValue, getPosPtr=>getPosPtr );
    IF fbList.bOk THEN
        sInfo := PVOID_TO_STRING( getValue );
    END_IF
END_IF

IF bRemoveHeadValue THEN(* remove head node from node list *)
    bRemoveHeadValue := FALSE;
    sInfo := '';

```

```

fbList.A_RemoveHeadValue( hList := hList, getValue=>getValue, getPosPtr=>getPosPtr );
IF fbList.bOk THEN
    sInfo := PVOID_TO_STRING( getValue );
END_IF
END_IF

IF bCount THEN(* count nodes in list *)
    bCount := FALSE;
    sInfo := UDINT_TO_STRING( hList.nCount );
END_IF

```

PROGRAM P_LIST_OF_STRUCTDATA

Dieses Programmteil zeigt wie strukturierte Datensätze statt einfacher 32-Zahlen in der Liste bearbeitet werden können. Dabei wird der 32 bit Knotenwert nur als Referenzpointer auf den tatsächlichen Knotenwert verwendet. Der Referenzpointer kann dann auch auf Instanzen von strukturierten Variablen oder anderen Datentypen zeigen. Die Funktionalität wurde in einem Funktionsblock gekapselt. Dieser Funktionsblock *FB_SpecialLinkedListCtrl* kann als eine spezialisierte Version des *FB_LinkedListCtrl*-Funktionsbausteins bezeichnet werden. Der *FB_LinkedListCtrl*-Baustein wird auch intern von dem spezialisierten FB verwendet.

Die Funktion *DATAELEMENT_TO_STRING* wird nur verwendet um eine visuelle Ausgabe der Knotenwerte zu ermöglichen.

Als Beispiel wird die Strukturierte Variable vom Typ: *ST_DataElement* verwendet. Der Clou: Sie können die Datentypdeklaration von *ST_DataElement* um weitere Membervariablen erweitern, ohne dass Sie an dem Programm oder dem *FB_SpecialLinkedListCtrl*-Funktionsbaustein Veränderungen durchführen müssen.

Die Typdeklaration von *ST_DataElement*:

```

(* Structured application data entry *)
TYPE ST_DataElement :
STRUCT
    (* Adapt this structure to match your application needs *)
    number    : UDINT := 0;
    name      : STRING(MAX_NAME_LENGTH) := '';
    price     : REAL := 0.0;
END_STRUCT
END_TYPE

```

Es wurde eine einfache Suchfunktion implementiert. Sie können nach Knoten mit einem bestimmten *name*, *number* oder *price* suchen.

Wie werden die 32 bit Knotenwerte zu Referenzpointern auf die ST_DataElement-Array-Instanzen?

Die max. Größe der Liste ist durch *MAX_DATA_ELEMENTS*-Konstante begrenzt. Folglich können in der Liste nur *MAX_DATA_ELEMENTS*-Referenzpointer gespeichert werden. Der *FB_SpecialLinkedListCtrl*-Baustein besitzt intern eine *ST_DataElement*-Array-Variablen mit derselben Größe wie die *T_LinkedListEntry*-Array-Variable. Zur Vereinfachung sind die Array-Indizes in beiden Arrays gleich!

Jedes *T_LinkedListEntry*-Array-Element kann nur einmal in die Liste "reingehängt" werden. Dabei sucht der *FB_LinkedListCtrl*-Funktionsbaustein nach einem freien/unbenutzten *T_LinkedListEntry*-Array-Element und fügt es bei Erfolg der Liste zu. Mit Hilfe der Aktion *A_GetIndexAtPosPtr* kann der verwendete Index des *T_LinkedListEntry*-Arrays ermittelt werden. Im nächsten Schritt wird dem zuletzt hinzugefügten 32 bit Knotenwert die Adresse desselben Arrayelements vom *ST_DataElement*-Array zugewiesen. Im Beispielprojekt durch den Aktionsaufruf: *A_SetValueAtPosPtr*.

nodes[index].value := ADR(dataPool[index])

Die Zuweisung wird z.B. in der *FB_SpecialLinkedListCtrl*->*A_AddHeadValue*-Aktion durchgeführt:

```

(* Adds head to the node list *)
MEMSET( ADR( getValue ), 0, SIZEOF( getValue ) );
getPosPtr := 0;

fbList.A_AddHeadValue( hList := hList, putValue := 16#00000000(* we will set this value later *), ge
tPosPtr=>getPosPtr, bOk=>bOk );
(* Add new element to the list, getPosPtr points to the new list node *)
IF fbList.bOk THEN(* Success *)
    fbList.A_GetIndexAtPosPtr( hList := hList, putPosPtr := getPosPtr, getValue =>indexOfElem, bOk=>
bOk );(* Get array index of getPosPtr *)
    IF fbList.bOk THEN(* Success *)
        pRefPtr := ADR( dataPool[indexOfElem] );(* Get pointer to the data element *)
        pRefPtr^ := putValue;(* set element value *)
    END_IF
END_IF

```

```

        fbList.A_SetValueAtPosPtr( hList := hList, putPosPtr := getPosPtr, putValue := pRefPtr, bOk=
>bOk ); (* Assign the node value = pointer to the data element *)
        IF fbList.bOk THEN(* Success *)
            getValue := putValue;
        END_IF
    END_IF
END_IF

PROGRAM P_LIST_OF_STRUCTDATA
VAR
    sInfo          : T_MaxString := '';
    bAddTailValue  : BOOL := TRUE;
    bAddHeadValue  : BOOL := TRUE;
    bGetTail       : BOOL := TRUE;
    bGetHead       : BOOL := TRUE;
    bFind          : BOOL := TRUE;
    bRemoveHeadValue : BOOL := TRUE;
    bRemoveTailValue : BOOL := TRUE;
    bCount         : BOOL := TRUE;

    search         : ST_DataElement := ( name := 'Couch', price := 99.98, number := 12345 );
(* search value ( by name, by price or by number ) *)
    eSearch        : E_SEARCH_CRITERIA := eSEARCH_BY_NAME;
(* / eSEARCH_BY_PRICE / eSEARCH_BY_NUMBER *)

    fbList        : FB_SpecialLinkedListCtrl;
(* Specialized linked list_control function block *)
    putValue       : ST_DataElement;
    getValue       : ST_DataElement;
    getPosPtr      : POINTER TO T_LinkedListEntry := 0;
    bInit         : BOOL := TRUE;
END_VAR

IF bInit THEN
    bInit := FALSE;
    fbList.A_Reset();(* reset / initialize list *)
END_IF

IF bAddTailValue THEN(* add some nodes to the list *)
    bAddTailValue := FALSE;

    putValue.number := 22222;
    putValue.name := 'TV set';
    putValue.price := 99.98;
    fbList.A_AddTailValue( putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbList.bOk THEN
        ;(* List overflow *)
    END_IF

    putValue.number := 11111;
    putValue.name := 'Couch';
    putValue.price := 99.98;
    fbList.A_AddTailValue( putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbList.bOk THEN
        ;(* List overflow *)
    END_IF

    putValue.number := 12345;
    putValue.name := 'Chair';
    putValue.price := 44.98;
    fbList.A_AddTailValue( putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbList.bOk THEN
        ;(* List overflow *)
    END_IF

    putValue.number := 67890;
    putValue.name := 'Table';
    putValue.price := 99.98;
    fbList.A_AddTailValue( putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbList.bOk THEN
        ;(* List overflow *)
    END_IF
END_IF

IF bAddHeadValue THEN
    bAddHeadValue := FALSE;

    putValue.number := 33333;

```

```

    putValue.name := 'Couch';
    putValue.price := 199.98;
    fbList.A_AddHeadValue( putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbList.bOk THEN
        ;(* List overflow *)
    END_IF

    putValue.number := 44444;
    putValue.name := 'Couch';
    putValue.price := 299.98;
    fbList.A_AddHeadValue( putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbList.bOk THEN
        ;(* List overflow *)
    END_IF
END_IF

IF bGetTail THEN(* enumerate all nodes in list (start at tail node) *)
    bGetTail := FALSE;
    sInfo := '';

    fbList.A_GetTail( getValue=>getValue, getPosPtr=>getPosPtr );
    IF fbList.bOk THEN
        sInfo := DATAELEMENT_TO_STRING( getValue );

        REPEAT
            fbList.A_GetPrev( putPosPtr := getPosPtr, getValue=>getValue, getPosPtr=>getPosPtr );
            IF fbList.bOk THEN
                sInfo := DATAELEMENT_TO_STRING( getValue );
            ELSE
                EXIT;
            END_IF
        UNTIL NOT fbList.bOk
        END_REPEAT
    END_IF
END_IF

IF bGetHead THEN(* enumerate all nodes in list (start at head node) *)
    bGetHead := FALSE;
    sInfo := '';

    fbList.A_GetHead( getValue=>getValue, getPosPtr=>getPosPtr );
    IF fbList.bOk THEN
        sInfo := DATAELEMENT_TO_STRING( getValue );

        REPEAT
            fbList.A_GetNext( putPosPtr := getPosPtr, getValue=>getValue, getPosPtr=>getPosPtr );
            IF fbList.bOk THEN
                sInfo := DATAELEMENT_TO_STRING( getValue );
            ELSE
                EXIT;
            END_IF
        UNTIL NOT fbList.bOk
        END_REPEAT
    END_IF
END_IF

IF bFind THEN(* search for node in the list by node value (name, price, number... )*)
    bFind := FALSE;
    getPosPtr := 0;(* start from first node element *)
    sInfo := '';

    REPEAT
        fbList.A_Find( eSearch := eSearch, putPosPtr := getPosPtr, putValue := search, getValue=>get
Value, getPosPtr=>getPosPtr );
        IF fbList.bOk THEN
            sInfo := DATAELEMENT_TO_STRING( getValue );
        ELSE
            EXIT;
        END_IF
    UNTIL NOT fbList.bOk
    END_REPEAT
END_IF

IF bRemoveTailValue THEN(* remove tail node from node list *)
    bRemoveTailValue := FALSE;

```

```
sInfo := '';
fbList.A_RemoveTailValue( getValue=>getValue, getPosPtr=>getPosPtr );
IF fbList.bOk THEN
    sInfo := DATAELEMENT_TO_STRING( getValue );
END_IF
END_IF

IF bRemoveHeadValue THEN(* remove head node from node list *)
    bRemoveHeadValue := FALSE;
    sInfo := '';
    fbList.A_RemoveHeadValue( getValue=>getValue, getPosPtr=>getPosPtr );
    IF fbList.bOk THEN
        sInfo := DATAELEMENT_TO_STRING( getValue );
    END_IF
END_IF

IF bCount THEN(* count nodes in list *)
    bCount := FALSE;
    sInfo := '';
    fbList.A_Count( );
    IF fbList.bOk THEN
        sInfo := UDINT_TO_STRING( fbList.nCount );
    END_IF
END_IF
```

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Uilities (System)

8.8 Beispiel: Schreiben/lesen einer CSV-Datei

Hier können Sie die kompletten Sourcen zum Beispielprojekt entpacken: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_Uilities/Resources/803601163.zip

Mit dem Beispielprojekt generierte CSV-Dateien:

Datenfelder ohne Binärdaten: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_Uilities/Resources/803605003.zip

Datenfelder beinhalten Binärdaten: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_Uilities/Resources/803430923.zip (bitte beachten Sie dass diese Datei nur von spezieller Software richtig interpretiert werden kann)

CSV steht für Comma-Separated Values. Folgende Dokumentation beschreibt wie CSV-Dateien mit Hilfe der SPS-CSV-Hilfsfunktionen geschrieben bzw. gelesen werden können. In den CSV-Dateien, die eigentlich Textdateien sind, können einfach strukturierte Datensätze gespeichert und zum Datenaustausch zwischen zwei Systemen verwendet werden. Dieses Format erlaubt eine Speicherung von unterschiedlich langen Tabellen oder Listen. Eine Zeile in der Tabelle entspricht einem Datensatz (auch Zeile) in der CSV-Datei. Eine Zelle in einer Tabelle entspricht einem Datenfeld in der CSV-Datei.

Allgemeine Informationen zum unterstützten CSV-Format

- Die Dateien im CSV-Format sollten die Endung **.csv** besitzen.
- Das CRLF-Zeichen (CR =Carriage Return, LF=Line Feed) wird zur Trennung der einzelnen Datensätze (Zeilen) verwendet (Windows-Betriebssysteme). D.h. nach jedem Datensatz muss ein CRLF folgen.
- Das Ende der CSV-Datei muss ebenfalls mit CRLF-Zeichen abgeschlossen werden.
- Binärdaten müssen in Hochkommas eingeschlossen werden. Wenn keine Hochkommas verwendet werden, dann dürfen im Datenfeld nur Zahlen und/oder Buchstaben verwendet werden.
- Felder mit Sonderzeichen/Steuerzeichen im Datenfeld werden in Anführungszeichen eingeschlossen. Wenn ein Einführungszeichen selbst im Datenfeld enthalten ist wird dieses verdoppelt.

- Ein spezielles Zeichen wird zur Trennung von Datenfeldern (Spalten) verwendet. Als Trennzeichen für die einzelnen Datenfelder wird von den Hilfsfunktionen standardmäßig ein Semikolon verwendet. In Deutschland und Europa wird als Datenfeldtrennzeichen ein Semikolon verwendet, in USA eher ein Komma. Über die globale SPS-Variable: **DEFAULT_CSV_FIELD_SEP** kann das Trennzeichen vom Semikolon auf Komma konfiguriert werden.
- Jeder Datensatz sollte die gleiche Anzahl an Datenfeldern (Spalten) besitzen.

Prinzipieller Aufbau einer CSV-Datei mit m-Spalten und n-Zeilen (die CRLF-Zeichen sind normalerweise nicht sichtbar und in der Abbildung vereinfacht mit den Buchstaben: **CRLF** dargestellt)

```
"Field1Record1";"Field2Record1"; ... ;"Field(m)Record1"CRLF
"Field1Record2";"Field2Record2"; ... ;"Field(m)Record2"CRLF
...
"Field1Record(n)";"Field2Record(n)"; ... ;"Field(m)Record(n)"CRLF
```

Verfügbare Funktionsbausteine und Funktionen

- [STRING TO CSVFIELD \[▶ 290\]](#), [ARG TO CSVFIELD \[▶ 247\]](#): Konvertiert SPS-Daten in ein Datenfeld im CSV-Format;
- [CSVFIELD TO STRING \[▶ 254\]](#), [CSVFIELD TO ARG \[▶ 253\]](#): Konvertiert Datenfeld im CSV-Format in SPS-Daten;
- [FB_CSVMemBufferWriter \[▶ 50\]](#): Generiert aus mehreren Datenfeldern Datensätze in einem Bytepuffer;
- [FB_CSVMemBufferReader \[▶ 49\]](#): Zerlegt Datensätze in einem Bytepuffer in einzelne Datenfelder;

CSV-Datei im Textmode oder Binärmode schreiben/lesen

Eine CSV-Datei kann im Textmode oder im Binärmode mit Hilfe der SPS-Funktionsbausteine für den Dateizugriff gelesen bzw. geschrieben werden. Abhängig von dem gewählten Modus ergeben sich Unterschiede mit Vor- und Nachteilen.

In 99% der Fälle können die CSV-Dateien im Textmode gelesen/geschrieben werden. Der Binärmode wird nur in den seltensten Fällen benötigt.

	Textmode	Binärmode
Funktionsbaustein für den Dateilesezugriff	FB_FileGets (Besonderheit: Das CR-Zeichen am Ende vom letzten Datensatz wird von diesem Baustein automatisch beim Lesezugriff aus der Datei entfernt. Damit der FB_CSVMemBufferReader Baustein einen solchen Datensatz interpretieren kann muss dieses Zeichen vorher wiederhergestellt/ eingefügt werden)	FB_FileRead
Funktionsbaustein für den Dateischreibzugriff	FB_FilePuts (Besonderheit: Ein zusätzliches CR-Zeichen am Ende vom letzten Datensatz wird von diesem Baustein automatisch beim Schreibzugriff in die Datei hinzugefügt. Der FB_CSVMemBufferWriter generiert aber die CR-Zeichen auch. Damit das Zeichen nicht doppelt in der CSV-Datei auftaucht muss dieses vor dem Schreibzugriff aus dem Puffer entfernt werden)	FB_FileWrite
Programmieraufwand	Kleiner	Größer
Sonderzeichen, nicht-druckbare Steuerzeichen im Datenfeld	Nicht erlaubt	Erlaubt
Maximale Datensatzlänge die geschrieben/gelesen werden kann	Auf 253 Zeichen begrenzt (Datensatz + CRLF). D.h. die Datensatzlänge darf 253 Zeichen nicht überschreiten.	Die maximale Datensatzlänge ist theoretisch unbegrenzt. Die Funktionsbausteine

	Textmode	Binärmode
		(FB_CSVMemBufferReader und FB_CSVMemBufferWriter) begrenzen ein CSV Feld jedoch auf die maximale Größe, welche in dem globalen Parameter cMaxCSVFieldValueSize angegeben ist.
Ein kompletter Datensatz kann mit dem Baustein für den Schreibzugriff geschrieben werden	Ja	Ja
Ein kompletter Datensatz kann mit dem Baustein für den Lesezugriff gelesen werden	Ja Ein Datensatz in einer reinen Textdatei endet mit CRLF. CRLF markiert in einer solchen Datei das Zeilenende. Der FB_FileGets-Funktionsbaustein liest die Daten bis zum CRLF.	Nein
Binärdaten im Datenfeld	Nicht erlaubt	Erlaubt
Hilfsfunktionen für die Konvertierung der SPS-Daten in den CSV-Format und umgekehrt	CSVFIELD TO STRING [▶ 254] STRING TO CSVFIELD [▶ 290]	CSVFIELD TO ARG [▶ 253] ARG TO CSVFIELD [▶ 247]
Unterstützte SPS Variablentypen die direkt geschrieben/gelesen werden können	T_MaxString (STRING mit 255 Zeichen), andere Datentypen müssen zuerst in einen String konvertiert werden und dann als Datenfeld im Stringformat geschrieben/gelesen werden.	Beliebige Datentypen können geschrieben/gelesen werden
Beispielcode	P_TextModeRead() P_TextModeWrite()	P_BinaryModeRead() P_BinaryModeWrite()

Beispielprojekt

Das Beispielprojekt beinhaltet eigentlich 4 Beispiele: 2 für den Schreib-/Lesezugriff im Textmode (bevorzugt) und 2 für den Schreib-/Lesezugriff im Binärmode (selten):

P_TextModeRead();

P_TextModeWrite();

P_BinaryModeRead();

P_BinaryModeWrite();

Prinzipieller Programmablauf beim Lesen einer CSV-Datei im Textmode:

1. Schritt: Die CSV-Datei im Textmode öffnen (FB_FileOpen). Wenn erfolgreich dann zu Schritt 2 gehen.
2. Schritt: Eine Zeile mit dem Funktionsbaustein FB_FileGets lesen. Das CR-Zeichen anhängen (siehe Hinweise in der Tabelle). Wenn erfolgreich dann zu Schritt 3 gehen, andernfalls zu Schritt 4 gehen (das Ende der Datei wurde erreicht oder ein Fehler ist aufgetreten).
3. Schritt: Die gelesene Zeile mit dem Funktionsbaustein FB_CSVMemBufferReader parsen. Es werden dabei die einzelnen Datenfelder gelesen. Danach zu Schritt 2 springen und die nächste Zeile lesen. Schritte 2 und 3 so lange wiederholen bis das Ende der Datei erreicht wurde oder ein Fehler aufgetreten ist.
4. Schritt: Die CSV-Datei schließen (FB_FileClose).

Prinzipieller Programmablauf beim Schreiben einer CSV-Datei im Textmode.

1. Schritt: Die CSV-Datei im Textmode öffnen (FB_FileOpen). Wenn erfolgreich dann zu Schritt 2 gehen.
2. Schritt: Mit dem Funktionsbaustein FB_CSVMemBufferWriter einen neuen Datensatz generieren. Die einzelnen Datenfelder werden dafür in einen Puffer hineingeschrieben. Dieser Puffer kann auch ein größerer String sein. Das CR-Zeichen vom Ende des Datensatzes entfernen und zu Schritt 3 gehen.

3. Schritt: Eine Zeile mit dem Funktionsbaustein FB_FilePuts schreiben. Dann Schritte 2 und 3 so lange wiederholen bis alle Datensätze geschrieben worden sind. Danach zu Schritt 4 gehen.

4. Schritt: Datei schließen (FB_FileClose).

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

8.9 Beispiel: Software-Uhren (RTC, RTC_EX, RTC_EX2)

Hier können Sie die kompletten Sourcen entpacken: https://infosys.beckhoff.com/content/1031/TcPlcLib_Tc2_Utilities/Resources/803608843.zip

Im folgenden Beispiel werden die drei Software-Uhren alle 5 Sekunden mit der lokalen Windows-Systemzeit synchronisiert (die lokale Windows-Systemzeit wird in der Taskleiste eingeblendet).

```
PROGRAM MAIN
VAR
    fbGetLocalTime : NT_GetTime;
    bBusy          : BOOL;
    bError         : BOOL;
    nErrID        : UDINT;
    presetTime    : TIMESTRUCT;

    syncTimer     : TON;
    syncTrigger   : F_TRIG;
    bSynchronize  : BOOL;

    fbRTC         : RTC;
    bValid_RTC    : BOOL;
    time_RTC      : DT;

    fbRTC_EX      : RTC_EX;
    bValid_RTCEX  : BOOL;
    time_RTCEX    : DT;
    nMilliseconds : DWORD;

    fbRTC_EX2     : RTC_EX2;
    bValid_RTCEX2 : BOOL;
    time_RTCEX2   : TIMESTRUCT;
    nMicroseconds : DWORD;
END_VAR
```

Expression	Type	Value	Prepared value	Comment
time_RTC	DATE_AND_TIME	DT#2012-8-10-16:5...		
fbRTC_EX	RTC_EX			
bValid_RTCEX	BOOL	TRUE		
time_RTCEX	DATE_AND_TIME	DT#2012-8-10-16:5...		
nMilliseconds	DWORD	185		
fbRTC_EX2	RTC_EX2			
bValid_RTCEX2	BOOL	TRUE		
time_RTCEX2	TIMESTRUCT			
wYear	WORD	2012		
wMonth	WORD	8		
wDayOfWeek	WORD	5		
wDay	WORD	10		
wHour	WORD	16		
wMinute	WORD	51		
wSecond	WORD	27		
wMilliseconds	WORD	185		
nMicroseconds	DWORD	383		

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

9 Anhang

9.1 Systemverhalten beim Schreiben von persistenten Daten

Schreibtrigger	Interne Optimierungen bei der Verwaltung der persistenten Daten	Datenkonsistenz der persistenten Daten	Mögliche Zyklusüberschreitung beim Schreiben
Funktionsbaustein WritePersistentData [▶ 146]	Keine	Alle Daten sind aus einem Zyklus	Ja, wenn das Schreiben aller Daten länger dauert als ca. der SPS-Zyklus
Funktionsbaustein FB_WritePersistentData [▶ 119] mit SPDM_2PASS [▶ 306]	Ja	Alle Daten sind aus einem Zyklus	Ja, wenn das Schreiben aller Daten länger dauert als ca. der SPS-Zyklus
Funktionsbaustein FB_WritePersistentData [▶ 119] mit SPDM_VAR_BOOST [▶ 306]	Ja	Daten einzelner persistenten Variablen sind aus einem Zyklus	Selten, wenn das Schreiben der größten persistenten Daten länger dauert als ca. der SPS-Zyklus)
TwinCAT-System-Stopp (die persistenten Daten werden beim Stopp automatisch geschrieben)	Ja	Alle Daten sind aus einem Zyklus	Keine

9.2 Formatspezifikation

Diese Formatspezifikation wird von den Funktionsbausteinen [FB_FormatString](#) [\[▶ 64\]](#) und [FB_FormatString2](#) [\[▶ 65\]](#) sowie der Funktion [F_FormatArgToStr](#) [\[▶ 266\]](#) verwendet. Während die Formatspezifikation den Funktionsbausteinen über eine String-Eingangsvariable übergeben wird, wird sie der Funktion über die einzelnen Funktionsparameter übergeben.

Die Formatspezifikation umfasst verschiedene erforderliche und optionale Parameterfelder:

- [Type](#) [\[▶ 356\]](#)
- [Flags](#) [\[▶ 357\]](#)
- [Witdh](#) [\[▶ 358\]](#)
- [Precision](#) [\[▶ 358\]](#)

Die einfachste Formatspezifikation beinhaltet nur das Prozentzeichen und das Type-Feld (z. B. %s). Alle Zeichen, die dem Prozentzeichen folgen, werden bis zu dem Type-Feld als Parameterfelder ausgewertet. Zeichen vor dem Prozentzeichen und hinter dem Type-Feld werden in den Ausgabestring kopiert. Beim Auftreten von nicht identifizierbaren oder unzulässigen Zeichen wird die Formatierung mit einem Fehler abgebrochen. Um das Prozentzeichen in dem Ausgabestring auszugeben, verwenden Sie zwei nacheinander folgende Prozentzeichen (%%).

Type

Erforderliches Parameterfeld. Das Type-Feld beinhaltet ein ASCII-Zeichen (Character), das bestimmt, ob das dazugehörige Argument als String, Integer oder Fließkommazahl interpretiert wird. Beachten Sie, dass einige der Type-Feld-Parameter „case-sensitive“ sind.

Typ	Argument	Ausgabe
b, B	BYTE, WORD, DWORD, REAL ¹ , SINT ² , INT ² , DINT ² , USINT, UINT, UDINT	Binärstring (z. B. '101010111000')

Typ	Argument	Ausgabe
o, O	BYTE, WORD, DWORD, REAL ¹ , SINT ² , INT ² , DINT ² , USINT, UINT, UDINT	Oktalstring
u, U	BYTE, WORD, DWORD, SINT ² , INT ² , DINT ² , USINT, UINT, UDINT	Dezimalstring ohne Vorzeichen
c, C	BYTE, USINT	Einzelner (ASCII) Byte-Character
f, F	REAL ³ , LREAL	<p>Fließkommazahl</p> <p>Der String hat die Form [-]dddd.dddd, (dddd sind Dezimalzahlen).</p> <p>Die Anzahl der Zahlen vor dem Dezimalpunkt hängt von dem Wert der Fließkommazahl ab. Die Anzahl der Zahlen hinter dem Dezimalpunkt hängt von der geforderten Präzision ab.</p> <p>Ein Vorzeichen erscheint nur bei negativen Werten. Bei einem unendlichen positiven Wert wird '#INF' und bei einem unendlichen negativen Wert wird '-#INF' zurückgeliefert.</p> <p>Wenn die übergebene Variable einen unzulässigen Wert besitzt (NaN, Not-a-Number), dann wird '#QNAN' oder '-#QNAN' zurückgeliefert.</p> <p>Wenn die Länge des formatierten Strings die maximal zulässige Länge des resultierenden Strings überschreitet, dann wird '#OVF' oder '-#OVF' zurückgeliefert.</p>
d, D	BYTE, WORD, DWORD, SINT, INT, DINT, USINT, UINT, UDINT	<p>Dezimalstring</p> <p>Ein Vorzeichen erscheint nur bei negativen Werten.</p>
s, S	STRING	<p>Single-byte-character string</p> <p>Zeichen werden ausgegeben, bis zur abschließenden Null oder bis der Precision-feld-Parameter erreicht wurde.</p>
X	BYTE, WORD, DWORD, REAL ¹ , SINT ² , INT ² , DINT ² , USINT, UINT, UDINT	<p>Hexadezimalstring</p> <p>Für die Formatierung werden die (upper-case) Zeichen benutzt ('ABCDEF').</p>
x	BYTE, WORD, DWORD, REAL ¹ , SINT ² , INT ² , DINT ² , USINT, UINT, UDINT	<p>Hexadezimalstring</p> <p>Für die Formatierung werden die (lower-case) Zeichen benutzt ('abcdef').</p>
E	Not implemented. Reserved for future use.	Fließkommazahlen in der wissenschaftlichen Notation
e	Not implemented. Reserved for future use.	Fließkommazahlen in der wissenschaftlichen Notation

¹ Es wird der Speicherinhalt der REAL-Variable als Binärstring-, Oktalstring, Hexadezimalstring oder Dezimalstring ausgegeben.

² Es wird der Speicherinhalt der vorzeichenbehafteten Typen als Binärstring-, Oktalstring, Hexadezimalstring oder Dezimalstring ausgegeben.

³ Die REAL-Variable wird in den LREAL-Typ konvertiert und dann formatiert.

Flags

Optionales Parameterfeld. Im Flag-Feld können ein oder mehrere Flags in einer beliebigen Reihenfolge bestimmt werden. Die Flag-Feld-Parameter bestimmen die Ausrichtung des formatierten Wertes, die Ausgabe von Vorzeichen, Leerzeichen und die Binär-/Oktal-/Hex-Präfixe.

Flag	Bedeutung	Type	Standard
-	Linksausrichtungs-Flag. Linksausrichtung des formatierten Wertes innerhalb der vorgegebenen Breite im Width-Parameter.	Kann in Verbindung mit allen Typen benutzt werden.	Rechtsausrichtung.
+	Vorzeichen-Flag. Erzwingt die Ausgabe des positiven Vorzeichens bei vorzeichenbehafteten positiven Zahlen.	Nur in Verbindung mit e, E, f, F, d, D , ansonsten wird das Flag ignoriert.	Das negative Vorzeichen erscheint nur bei negativen Werten.
0	Null-Flag. Wenn dieses Flag dem Width-Parameter vorangestellt wurde, wird der resultierende String von Links mit Nullen aufgefüllt, bis die gewünschte Breite erreicht wurde.	Nur in Verbindung mit e, E, f, F, s, S , ansonsten wird das Flag ignoriert. Das Null-Flag wird auch ignoriert, wenn zusätzlich das Linksausrichtungs-Flag (-) gesetzt ist.	Kein Auffüllen mit Nullen.
Leerzeichen (' ')	Leerzeichen-Flag Ein Leerzeichen wird einem positiven Wert vorangestellt.	Nur in Verbindung mit e, E, f, F, d, D , ansonsten wird das Flag ignoriert. Das Leerzeichen-Flag wird auch ignoriert, wenn gleichzeitig auch das Vorzeichen-Flag (+) gesetzt wurde	Kein Leerzeichen.
#	Präfix-Flag. Dem formatierten Wert wird ein IEC- oder Standard-C-Präfix vorangestellt. IEC-Präfixe: 2#, 8#, 16# (default) Standard-C-Präfixe: 0, 0x, 0X	Nur in Verbindung mit b, B, o, O, x, X , ansonsten wird das Flag ignoriert. Der Standard-C-Präfix-Typ kann durch das Setzen der globalen Variablen <code>GLOBAL_FORMAT_HASH_PREFIX_TYPE</code> im Programm aktiviert werden: <code>GLOBAL_FORMAT_HASH_PREFIX_TYPE := HASHPREFIX_STDC;</code>	Kein Präfix.

Width

Optionales Parameterfeld. Das Width-Feld beinhaltet einen positiven dezimalen Wert, der die minimale Anzahl der ausgegebenen Zeichen im Ausgabestring bestimmt.

In Abhängigkeit von dem Ausrichtungs-Flag werden dem Ausgabestring rechts oder links Leerzeichen hinzugefügt, bis die gewünschte Breite erreicht ist. Wenn das Null-Flag dem Width-Feld-Parameter vorangestellt ist, wird der resultierende String von Links mit Nullen aufgefüllt, bis die gewünschte Breite erreicht ist. Der Ausgabestring wird niemals durch den Width-Feld-Parameter auf die gewünschte Länge abgeschnitten.

Für den Width-Feld-Parameter kann auch ein Sternchen (*) angegeben werden. Der benötigte Wert wird dann von einem Argument geliefert (zulässige Typen: BYTE, WORD, DWORD, USINT, UINT, UDINT). Dem Argument für den Width-Feld-Parameter folgt dann das Argument für den zu formatierenden Wert.

Precision

Optionales Parameterfeld. Das Precision-Feld folgt nach dem Punkt (.) und beinhaltet einen positiven dezimalen Wert. Folgt dem Punkt kein Wert, dann wird für die Präzision der Standardwert genommen (siehe Tabelle).

Typ	Bedeutung	Standard
b, B, o, O, u, U, x, X, d, D	Der Precision-Feld-Parameter legt fest, wie viele Dezimalzeichen (Digits) in dem Ausgabestring ausgegeben werden. Bei fehlenden Zeichen wird von Links mit Nullen aufgefüllt. Der Ausgabestring wird aber niemals abgeschnitten.	Standard: 1
c, C	Hat keine Bedeutung und wird ignoriert.	Ein Zeichen wird ausgegeben.
f, F	Der Precision-Feld-Parameter legt die Anzahl der Nachkommastellen fest. Der Argument-Wert wird immer auf die entsprechende Anzahl der Nachkommastellen gerundet.	Standard: 6 Nachkommastellen
s, S	Der Precision-Feld-Parameter legt fest, wie viele Zeichen aus dem Argument-String ausgegeben werden. Zeichen, die den Präzisionswert überschreiten werden nicht ausgegeben.	Es werden alle Zeichen bis zur abschließenden Null ausgegeben.

Für den Precision-Feld-Parameter kann auch ein Sternchen (*) angegeben werden. Der benötigte Wert wird dann von einem Argument geliefert (zulässige Typen: BYTE, WORD, DWORD, USINT, UINT, UDINT). Dem Argument für den Precision-Feld-Parameter folgt dann das Argument für den zu formatierenden Wert.

9.3 Format-Fehlercodes

Folgende Fehlercodes werden von dem Funktionsbausteinen [FB FormatString \[► 64\]](#) und [FB FormatString2 \[► 65\]](#) sowie der Funktion [F.FormatArgToStr \[► 266\]](#) zurückgeliefert. Bei mehreren Argumenten wird zusätzlich zu dem Fehlercode die Argumentnummer (1..9) zurückgeliefert. Die Argumentnummer gibt Auskunft über die Stelle, an der ein Fehler während der Formatierung festgestellt wurde.

Fehlercode	Bedeutung
16#00000000	No error
16#00000010 + Argument number (1..9)	Percent sign (%) at invalid position
16#00000020 + Argument number (1..9)	Asterisk parameter at invalid position
16#00000040 + Argument number (1..9)	Invalid width field value
16#00000080 + Argument number (1..9)	Invalid precision field value
16#00000100 + Argument number (1..9)	One of the flags at invalid position
16#00000200 + Argument number (1..9)	The width or precision field value at invalid position
16#00000400 + Argument number (1..9)	Dot "." sign of precision field at invalid position
16#00000800 + Argument number (1..9)	Invalid (unsupported) type field value
16#00001000 + Argument number (1..9)	Different type field and argument parameter
16#00002000 + Argument number (1..9)	Invalid format string parameters
16#00004000 + Argument number (1..9)	Too much arguments in format string
16#00008000 + Argument number (1..9)	Destination string buffer overflow (formatted string is too long)
16#00010000 + Argument number (1..9)	Invalid pointer input

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

9.4 Scope-Server-Fehlercodes

Folgende Fehlercodes werden von dem Funktionsbaustein `FB_ScopeServerControl` [► 108] zurückgeliefert.

```

TYPE E_UTILITIES_ERRORCODES :
(
  eUtilError_NoError           := 0,
  eUtilError_ScopeServerNotAvailable := 16#8001,
  eUtilError_ScopeServerStateChange := 16#8002
);
END_TYPE

```

Fehlercode	Enum	Bedeutung
0x0000	eUtilError_NoError	kein Fehler
0x8001	eUtilError_ScopeServerNotAvailable	TwinCAT Scope Server ist nicht verfügbar. Vielleicht ist er nicht installiert.
0x8002	eUtilError_ScopeServerStateChange	Der angeforderte Statuswechsel ist nicht zulässig. Schlagen Sie die zulässigen Statuswechsel in <code>B_ScopeServerControl state diagram</code> [► 108] nach.

Voraussetzungen

Entwicklungsumgebung	Zielplattform	Einzubindende SPS-Bibliotheken (Kategoriegruppe)
TwinCAT v3.1.0	PC oder CX (x86, x64, ARM)	Tc2_Utilities (System)

9.5 ADS Return Codes

Gruppierung der Fehlercodes:

Globale Fehlercodes: `0x0000` [► 360]... (`0x9811_0000` ...)

Router Fehlercodes: `0x0500` [► 361]... (`0x9811_0500` ...)

Allgemeine ADS Fehler: `0x0700` [► 361]... (`0x9811_0700` ...)

RTime Fehlercodes: `0x1000` [► 363]... (`0x9811_1000` ...)

Globale Fehlercodes

Hex	Dec	HRESULT	Name	Beschreibung
0x0	0	0x98110000	ERR_NOERROR	Kein Fehler.
0x1	1	0x98110001	ERR_INTERNAL	Interner Fehler.
0x2	2	0x98110002	ERR_NORTIME	Keine Echtzeit.
0x3	3	0x98110003	ERR_ALLOCLOCKEDMEM	Zuweisung gesperrt - Speicherfehler.
0x4	4	0x98110004	ERR_INSERTMAILBOX	Postfach voll – Es konnte die ADS Nachricht nicht versendet werden. Reduzieren der Anzahl der ADS Nachrichten pro Zyklus bringt Abhilfe.
0x5	5	0x98110005	ERR_WRONGRECEIVEHMSG	Falsches HMSG.
0x6	6	0x98110006	ERR_TARGETPORTNOTFOUND	Ziel-Port nicht gefunden – ADS Server ist nicht gestartet oder erreichbar.
0x7	7	0x98110007	ERR_TARGETMACHINENOTFOUND	Zielrechner nicht gefunden – AMS Route wurde nicht gefunden.
0x8	8	0x98110008	ERR_UNKNOWNCMDID	Unbekannte Befehl-ID.
0x9	9	0x98110009	ERR_BADTASKID	Ungültige Task-ID.
0xA	10	0x9811000A	ERR_NOIO	Kein IO.
0xB	11	0x9811000B	ERR_UNKNOWNAMSCMD	Unbekannter AMS-Befehl.
0xC	12	0x9811000C	ERR_WIN32ERROR	Win32 Fehler.
0xD	13	0x9811000D	ERR_PORTNOTCONNECTED	Port nicht verbunden.
0xE	14	0x9811000E	ERR_INVALIDAMSLLENGTH	Ungültige AMS-Länge.
0xF	15	0x9811000F	ERR_INVALIDAMSNETID	Ungültige AMS Net ID.
0x10	16	0x98110010	ERR_LOWINSTLEVEL	Installations-Level ist zu niedrig –TwinCAT 2 Lizenzfehler.
0x11	17	0x98110011	ERR_NODEBUGINTAVAILABLE	Kein Debugging verfügbar.

Hex	Dec	HRESULT	Name	Beschreibung
0x12	18	0x98110012	ERR_PORTDISABLED	Port deaktiviert – TwinCAT System Service nicht gestartet.
0x13	19	0x98110013	ERR_PORTALREADYCONNECTED	Port bereits verbunden.
0x14	20	0x98110014	ERR_AMSSYNC_W32ERROR	AMS Sync Win32 Fehler.
0x15	21	0x98110015	ERR_AMSSYNC_TIMEOUT	AMS Sync Timeout.
0x16	22	0x98110016	ERR_AMSSYNC_AMSERROR	AMS Sync Fehler.
0x17	23	0x98110017	ERR_AMSSYNC_NOINDEXINMAP	Keine Index-Map für AMS Sync vorhanden.
0x18	24	0x98110018	ERR_INVALIDAMSPORT	Ungültiger AMS-Port.
0x19	25	0x98110019	ERR_NOMEMORY	Kein Speicher.
0x1A	26	0x9811001A	ERR_TCPSEND	TCP Sendefehler.
0x1B	27	0x9811001B	ERR_HOSTUNREACHABLE	Host nicht erreichbar.
0x1C	28	0x9811001C	ERR_INVALIDAMSFAGMENT	Ungültiges AMS Fragment.
0x1D	29	0x9811001D	ERR_TLSSSEND	TLS Sendefehler – Secure ADS Verbindung fehlgeschlagen.
0x1E	30	0x9811001E	ERR_ACCESSDENIED	Zugriff Verweigert – Secure ADS Zugriff verweigert.

Router Fehlercodes

Hex	Dec	HRESULT	Name	Beschreibung
0x500	1280	0x98110500	ROUTERERR_NOLOCKEDMEMORY	Lockerter Speicher kann nicht zugewiesen werden.
0x501	1281	0x98110501	ROUTERERR_RESIZEMEMORY	Die Größe des Routerspeichers konnte nicht geändert werden.
0x502	1282	0x98110502	ROUTERERR_MAILBOXFULL	Das Postfach hat die maximale Anzahl der möglichen Meldungen erreicht.
0x503	1283	0x98110503	ROUTERERR_DEBUGBOXFULL	Das Debug Postfach hat die maximale Anzahl der möglichen Meldungen erreicht.
0x504	1284	0x98110504	ROUTERERR_UNKNOWNPORTTYPE	Der Porttyp ist unbekannt.
0x505	1285	0x98110505	ROUTERERR_NOTINITIALIZED	Router ist nicht initialisiert.
0x506	1286	0x98110506	ROUTERERR_PORTALREADYINUSE	Die Portnummer ist bereits vergeben.
0x507	1287	0x98110507	ROUTERERR_NOTREGISTERED	Der Port ist nicht registriert.
0x508	1288	0x98110508	ROUTERERR_NOMOREQUEUES	Die maximale Portanzahl ist erreicht.
0x509	1289	0x98110509	ROUTERERR_INVALIDPORT	Der Port ist ungültig.
0x50A	1290	0x9811050A	ROUTERERR_NOTACTIVATED	Der Router ist nicht aktiv.
0x50B	1291	0x9811050B	ROUTERERR_FRAGMENTBOXFULL	Das Postfach hat die maximale Anzahl für fragmentierte Nachrichten erreicht.
0x50C	1292	0x9811050C	ROUTERERR_FRAGMENTTIMEOUT	Fragment Timeout aufgetreten.
0x50D	1293	0x9811050D	ROUTERERR_TOBEREMOVED	Port wird entfernt.

Allgemeine ADS Fehlercodes

Hex	Dec	HRESULT	Name	Beschreibung
0x700	1792	0x98110700	ADSERR_DEVICE_ERROR	Allgemeiner Gerätefehler.
0x701	1793	0x98110701	ADSERR_DEVICE_SRVNOTSUPP	Service wird vom Server nicht unterstützt.
0x702	1794	0x98110702	ADSERR_DEVICE_INVALIDGRP	Ungültige Index-Gruppe.
0x703	1795	0x98110703	ADSERR_DEVICE_INVALIDOFFSET	Ungültiger Index-Offset.
0x704	1796	0x98110704	ADSERR_DEVICE_INVALIDACCESS	Lesen oder Schreiben nicht gestattet.
0x705	1797	0x98110705	ADSERR_DEVICE_INVALIDSIZE	Parametergröße nicht korrekt.
0x706	1798	0x98110706	ADSERR_DEVICE_INVALIDDATA	Ungültige Daten-Werte.
0x707	1799	0x98110707	ADSERR_DEVICE_NOTREADY	Gerät nicht betriebsbereit.
0x708	1800	0x98110708	ADSERR_DEVICE_BUSY	Gerät beschäftigt.
0x709	1801	0x98110709	ADSERR_DEVICE_INVALIDCONTEXT	Ungültiger Kontext vom Betriebssystem - Kann durch Verwendung von ADS Bausteinen in unterschiedlichen Tasks auftreten. Abhilfe kann die Multitasking-Synchronisation in der SPS geben.
0x70A	1802	0x9811070A	ADSERR_DEVICE_NOMEMORY	Nicht genügend Speicher.
0x70B	1803	0x9811070B	ADSERR_DEVICE_INVALIDPARM	Ungültige Parameter-Werte.
0x70C	1804	0x9811070C	ADSERR_DEVICE_NOTFOUND	Nicht gefunden (Dateien,...).
0x70D	1805	0x9811070D	ADSERR_DEVICE_SYNTAX	Syntax-Fehler in Datei oder Befehl.
0x70E	1806	0x9811070E	ADSERR_DEVICE_INCOMPATIBLE	Objekte stimmen nicht überein.

Hex	Dec	HRESULT	Name	Beschreibung
0x70F	1807	0x9811070F	ADSERR_DEVICE_EXISTS	Objekt ist bereits vorhanden.
0x710	1808	0x98110710	ADSERR_DEVICE_SYMBOLNOTFOUND	Symbol nicht gefunden.
0x711	1809	0x98110711	ADSERR_DEVICE_SYMBOLVERSIONINVALID	Symbol-Version ungültig – Kann durch einen Online-Change auftreten. Erzeuge einen neuen Handle.
0x712	1810	0x98110712	ADSERR_DEVICE_INVALIDSTATE	Gerät (Server) ist im ungültigen Zustand.
0x713	1811	0x98110713	ADSERR_DEVICE_TRANSMODENOTSUPP	AdsTransMode nicht unterstützt.
0x714	1812	0x98110714	ADSERR_DEVICE_NOTIFYHANDINVALID	Notification Handle ist ungültig.
0x715	1813	0x98110715	ADSERR_DEVICE_CLIENTUNKNOWN	Notification-Client nicht registriert.
0x716	1814	0x98110716	ADSERR_DEVICE_NOMOREHDLS	Keine weiteren Handles verfügbar.
0x717	1815	0x98110717	ADSERR_DEVICE_INVALIDWATCHSIZE	Größe der Notification zu groß.
0x718	1816	0x98110718	ADSERR_DEVICE_NOTINIT	Gerät nicht initialisiert.
0x719	1817	0x98110719	ADSERR_DEVICE_TIMEOUT	Gerät hat einen Timeout.
0x71A	1818	0x9811071A	ADSERR_DEVICE_NOINTERFACE	Interface Abfrage fehlgeschlagen.
0x71B	1819	0x9811071B	ADSERR_DEVICE_INVALIDINTERFACE	Falsches Interface angefordert.
0x71C	1820	0x9811071C	ADSERR_DEVICE_INVALIDCLSID	Class-ID ist ungültig.
0x71D	1821	0x9811071D	ADSERR_DEVICE_INVALIDOBJID	Object-ID ist ungültig.
0x71E	1822	0x9811071E	ADSERR_DEVICE_PENDING	Anforderung steht aus.
0x71F	1823	0x9811071F	ADSERR_DEVICE_ABORTED	Anforderung wird abgebrochen.
0x720	1824	0x98110720	ADSERR_DEVICE_WARNING	Signal-Warnung.
0x721	1825	0x98110721	ADSERR_DEVICE_INVALIDARRAYIDX	Ungültiger Array-Index.
0x722	1826	0x98110722	ADSERR_DEVICE_SYMBOLNOTACTIVE	Symbol nicht aktiv.
0x723	1827	0x98110723	ADSERR_DEVICE_ACCESSDENIED	Zugriff verweigert.
0x724	1828	0x98110724	ADSERR_DEVICE_LICENSENOTFOUND	Fehlende Lizenz.
0x725	1829	0x98110725	ADSERR_DEVICE_LICENSEEXPIRED	Lizenz abgelaufen.
0x726	1830	0x98110726	ADSERR_DEVICE_LICENSEEXCEEDED	Lizenz überschritten.
0x727	1831	0x98110727	ADSERR_DEVICE_LICENSEINVALID	Lizenz ungültig.
0x728	1832	0x98110728	ADSERR_DEVICE_LICENSESYSTEMID	Lizenzproblem: System-ID ist ungültig.
0x729	1833	0x98110729	ADSERR_DEVICE_LICENSENOTIMELIMIT	Lizenz nicht zeitlich begrenzt.
0x72A	1834	0x9811072A	ADSERR_DEVICE_LICENSEFUTUREISSUE	Lizenzproblem: Zeitpunkt in der Zukunft.
0x72B	1835	0x9811072B	ADSERR_DEVICE_LICENSETIMETOLONG	Lizenz-Zeitraum zu lang.
0x72C	1836	0x9811072C	ADSERR_DEVICE_EXCEPTION	Exception beim Systemstart.
0x72D	1837	0x9811072D	ADSERR_DEVICE_LICENSEDUPLICATED	Lizenz-Datei zweimal gelesen.
0x72E	1838	0x9811072E	ADSERR_DEVICE_SIGNATUREINVALID	Ungültige Signatur.
0x72F	1839	0x9811072F	ADSERR_DEVICE_CERTIFICATEINVALID	Zertifikat ungültig.
0x730	1840	0x98110730	ADSERR_DEVICE_LICENSEOEMNOTFOUND	Public Key vom OEM nicht bekannt.
0x731	1841	0x98110731	ADSERR_DEVICE_LICENSERESTRICTED	Lizenz nicht gültig für diese System.ID.
0x732	1842	0x98110732	ADSERR_DEVICE_LICENSEDEMODENIED	Demo-Lizenz untersagt.
0x733	1843	0x98110733	ADSERR_DEVICE_INVALIDFNCID	Funktions-ID ungültig.
0x734	1844	0x98110734	ADSERR_DEVICE_OUTOFRANGE	Außerhalb des gültigen Bereiches.
0x735	1845	0x98110735	ADSERR_DEVICE_INVALIDALIGNMENT	Ungültiges Alignment.
0x736	1846	0x98110736	ADSERR_DEVICE_LICENSEPLATFORM	Ungültiger Plattform Level.
0x737	1847	0x98110737	ADSERR_DEVICE_FORWARD_PL	Kontext – Weiterleitung zum Passiv-Level.
0x738	1848	0x98110738	ADSERR_DEVICE_FORWARD_DL	Kontext – Weiterleitung zum Dispatch-Level.
0x739	1849	0x98110739	ADSERR_DEVICE_FORWARD_RT	Kontext – Weiterleitung zur Echtzeit.
0x740	1856	0x98110740	ADSERR_CLIENT_ERROR	Clientfehler.
0x741	1857	0x98110741	ADSERR_CLIENT_INVALIDPARG	Dienst enthält einen ungültigen Parameter.
0x742	1858	0x98110742	ADSERR_CLIENT_LISTEMPTY	Polling-Liste ist leer.
0x743	1859	0x98110743	ADSERR_CLIENT_VARUSED	Var-Verbindung bereits im Einsatz.
0x744	1860	0x98110744	ADSERR_CLIENT_DUPLINVOKEID	Die aufgerufene ID ist bereits in Benutzung.
0x745	1861	0x98110745	ADSERR_CLIENT_SYNC_TIMEOUT	Timeout ist aufgetreten – Die Gegenstelle antwortet nicht im vorgegebenen ADS Timeout. Die Routeneinstellung der Gegenstelle kann falsch konfiguriert sein.
0x746	1862	0x98110746	ADSERR_CLIENT_W32ERROR	Fehler im Win32 Subsystem.
0x747	1863	0x98110747	ADSERR_CLIENT_TIMEOUTINVALID	Ungültiger Client Timeout-Wert.
0x748	1864	0x98110748	ADSERR_CLIENT_PORTNOTOPEN	Port nicht geöffnet.
0x749	1865	0x98110749	ADSERR_CLIENT_NOAMSADDR	Keine AMS Adresse.
0x750	1872	0x98110750	ADSERR_CLIENT_SYNCINTERNAL	Interner Fehler in Ads-Sync.

Hex	Dec	HRESULT	Name	Beschreibung
0x751	1873	0x98110751	ADSERR_CLIENT_ADDHASH	Überlauf der Hash-Tabelle.
0x752	1874	0x98110752	ADSERR_CLIENT_REMOVEHASH	Schlüssel in der Tabelle nicht gefunden.
0x753	1875	0x98110753	ADSERR_CLIENT_NOMORESVM	Keine Symbole im Cache.
0x754	1876	0x98110754	ADSERR_CLIENT_SYNCRESINVALID	Ungültige Antwort erhalten.
0x755	1877	0x98110755	ADSERR_CLIENT_SYNCPORTLOCKED	Sync Port ist verriegelt.
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	Die Anfrage wurde abgebrochen.

RTime Fehlercodes

Hex	Dec	HRESULT	Name	Beschreibung
0x1000	4096	0x98111000	RTERR_INTERNAL	Interner Fehler im Echtzeit-System.
0x1001	4097	0x98111001	RTERR_BADTIMERPERIODS	Timer-Wert nicht gültig.
0x1002	4098	0x98111002	RTERR_INVALIDTASKPTR	Task-Pointer hat den ungültigen Wert 0 (null).
0x1003	4099	0x98111003	RTERR_INVALIDSTACKPTR	Stack-Pointer hat den ungültigen Wert 0 (null).
0x1004	4100	0x98111004	RTERR_PRIOEXISTS	Die Request Task Priority ist bereits vergeben.
0x1005	4101	0x98111005	RTERR_NOMORETCB	Kein freier TCB (Task Control Block) verfügbar. Maximale Anzahl von TCBs beträgt 64.
0x1006	4102	0x98111006	RTERR_NOMORESEMAS	Keine freien Semaphoren zur Verfügung. Maximale Anzahl der Semaphoren beträgt 64.
0x1007	4103	0x98111007	RTERR_NOMOREQUEUES	Kein freier Platz in der Warteschlange zur Verfügung. Maximale Anzahl der Plätze in der Warteschlange beträgt 64.
0x100D	4109	0x9811100D	RTERR_EXTIRQALREADYDEF	Ein externer Synchronisations-Interrupt wird bereits angewandt.
0x100E	4110	0x9811100E	RTERR_EXTIRQNOTDEF	Kein externer Sync-Interrupt angewandt.
0x100F	4111	0x9811100F	RTERR_EXTIRQINSTALLFAILED	Anwendung des externen Synchronisierungs-Interrupts ist fehlgeschlagen.
0x1010	4112	0x98111010	RTERR_IRQLNOTLESSOREQUAL	Aufruf einer Service-Funktion im falschen Kontext
0x1017	4119	0x98111017	RTERR_VMXNOTSUPPORTED	Intel VT-x Erweiterung wird nicht unterstützt.
0x1018	4120	0x98111018	RTERR_VMXDISABLED	Intel VT-x Erweiterung ist nicht aktiviert im BIOS.
0x1019	4121	0x98111019	RTERR_VMXCONTROLSMISSING	Fehlende Funktion in Intel VT-x Erweiterung.
0x101A	4122	0x9811101A	RTERR_VMXENABLEFAILS	Aktivieren von Intel VT-x schlägt fehl.

Spezifische positive HRESULT Return Codes:

HRESULT	Name	Beschreibung
0x0000_0000	S_OK	Kein Fehler.
0x0000_0001	S_FALSE	Kein Fehler. Bsp.: erfolgreiche Abarbeitung, bei der jedoch ein negatives oder unvollständiges Ergebnis erzielt wurde.
0x0000_0203	S_PENDING	Kein Fehler. Bsp.: erfolgreiche Abarbeitung, bei der jedoch noch kein Ergebnis vorliegt.
0x0000_0256	S_WATCHDOG_TIMEOUT	Kein Fehler. Bsp.: erfolgreiche Abarbeitung, bei der jedoch eine Zeitüberschreitung eintrat.

TCP Winsock-Fehlercodes

Hex	Dec	Name	Beschreibung
0x274C	10060	WSAETIMEDOUT	Verbindungs Timeout aufgetreten - Fehler beim Herstellen der Verbindung, da die Gegenstelle nach einer bestimmten Zeitspanne nicht ordnungsgemäß reagiert hat, oder die hergestellte Verbindung konnte nicht aufrecht erhalten werden, da der verbundene Host nicht reagiert hat.
0x274D	10061	WSAECONNREFUSED	Verbindung abgelehnt - Es konnte keine Verbindung hergestellt werden, da der Zielcomputer dies explizit abgelehnt hat. Dieser Fehler resultiert normalerweise aus dem Versuch, eine Verbindung mit einem Dienst herzustellen, der auf dem fremden Host inaktiv ist—das heißt, einem Dienst, für den keine Serveranwendung ausgeführt wird.
0x2751	10065	WSAEHOSTUNREACH	Keine Route zum Host - Ein Socketvorgang bezog sich auf einen nicht verfügbaren Host.

Weitere Winsock-Fehlercodes: [Win32-Fehlercodes \[▶ 364\]](#)

9.6 Win32 Error Codes

Die folgende Tabelle enthält eine Liste der Win32-Fehlercodes.

[0](#) [[▶ 364](#)], [100](#) [[▶ 365](#)], [200](#) [[▶ 368](#)], [1001](#) [[▶ 369](#)], [1100](#) [[▶ 371](#)], [1200](#) [[▶ 373](#)], [1400](#) [[▶ 379](#)], [1600](#) [[▶ 380](#)], [1800](#) [[▶ 384](#)], [2000](#) [[▶ 386](#)], [3000](#) [[▶ 387](#)], [5000](#) [[▶ 390](#)], [6000](#) [[▶ 393](#)], [8000](#) [[▶ 395](#)], [8500](#) [[▶ 404](#)], [9001](#) [[▶ 409](#)], [10004](#) [[▶ 410](#)], [12000](#) [[▶ 413](#)],

		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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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 Netzwerkkarte 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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 Dateibezeichner 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.
136	0x00000088	ERROR_NOT_JOINED	Das System versucht, die Verknüpfung eines Laufwerks zu löschen, die nicht Mitglied ist.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 Sperrtyp.
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.
183	0x000000B7	ERROR_ALREADY_EXISTS	Eine Datei kann nicht erstellt werden, wenn die Datei bereits vorhanden ist.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 Semaphore.
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 Startausstehende 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.
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 Dateimarke.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 abzubrechen, 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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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 Gerätenamen 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 Gerätenamen 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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 Freigabennamens 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 angeben 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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 Anmelde-server 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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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).
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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_THRE AD	Von einem anderen Thread erstellte Objekt kann nicht zerstört werden.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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_HWNDS_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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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 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

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
			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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
1901	0x0000076D	ERROR_INVALID_TIME	Die angegebene Zeit ist ungültig.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
1902	0x0000076E	ERROR_INVALID_FORM_NAME	Der angegebene Formularename 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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
5031	0x000013A7	ERROR_CLUSTERLOG_EXCEEDS_MAXSIZE	Das Clusterprotokoll überschreitet die maximale Größe.
5032	0x000013A8	ERROR_CLUSTERLOG_CHKPOINT_NOT_FOUND	Kein Prüfpunkt Datensatz 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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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_CHANGED	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 Ressourcenname 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 Verknüpfungsoperation passieren, wenn die Clusterdatenbank während der Verknüpfung geändert wurde.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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_LOCALLY	Der Verzeichnisdienst ausgewertet Gruppenmitgliedschaften lokal.
8202	0x0000200A	ERROR_DS_NO_ATTRIBUTE_OR_VALUE	Die angegebene Verzeichnisdienstattribut oder der Wert ist nicht vorhanden.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 Replikat 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 Objektname 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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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-Anzeigenname.
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-enthalten
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 festgestellt.
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 Zielservers bereits vorhanden ist.
8449	0x00002101	ERROR_DS_DRA_REF_NOT_FOUND	Die Replikationsverweisinformationen für den Zielservers 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.
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 Zielservers 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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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 hinzufügen, 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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 Zielsystem 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 domänenübergreifend 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_MEMBERSHIPS	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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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_INFILE	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.
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.

Dezi- mal	Hexadezimal	Fehler		Beschreibung
		Name		
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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_NC	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_DS_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.
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 Verzeichnispartition existiert nicht.
9902	0x000026AE	DNS_ERROR_DP_ALREADY_EXISTS	Die angegebene Verzeichnispartition ist bereits vorhanden.
9903	0x000026AF	DNS_ERROR_DP_NOT_ENLISTED	Der DS ist nicht in der angegebenen Verzeichnispartition eingetragen.
9904	0x000026B0	DNS_ERROR_DP_ALREADY_ENLISTED	Der DS wird in die angegebene Verzeichnispartition 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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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 Datagrammsocket 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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 Diensttyp 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 Strukturierungsrateobjekt gefunden.
11031	0x00002B17	WSA_QOS_RESERVED_PETYPE	Im QOS-spezifischen Anbieterpuffer wurde ein reserviertes Richtlinienelement 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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
12006	0x00002EE6	ERROR_SXS_ACTIVATION_CONTEXT_DISABLE_D	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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 Doppelzähler 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.
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 Dateiende.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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 Nachschlüssel 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.
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_KEY	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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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.
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.

Fehler			Beschreibung
Dezimal	Hexadezimal	Name	
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	

Mehr Informationen:
www.beckhoff.com/te1000

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

