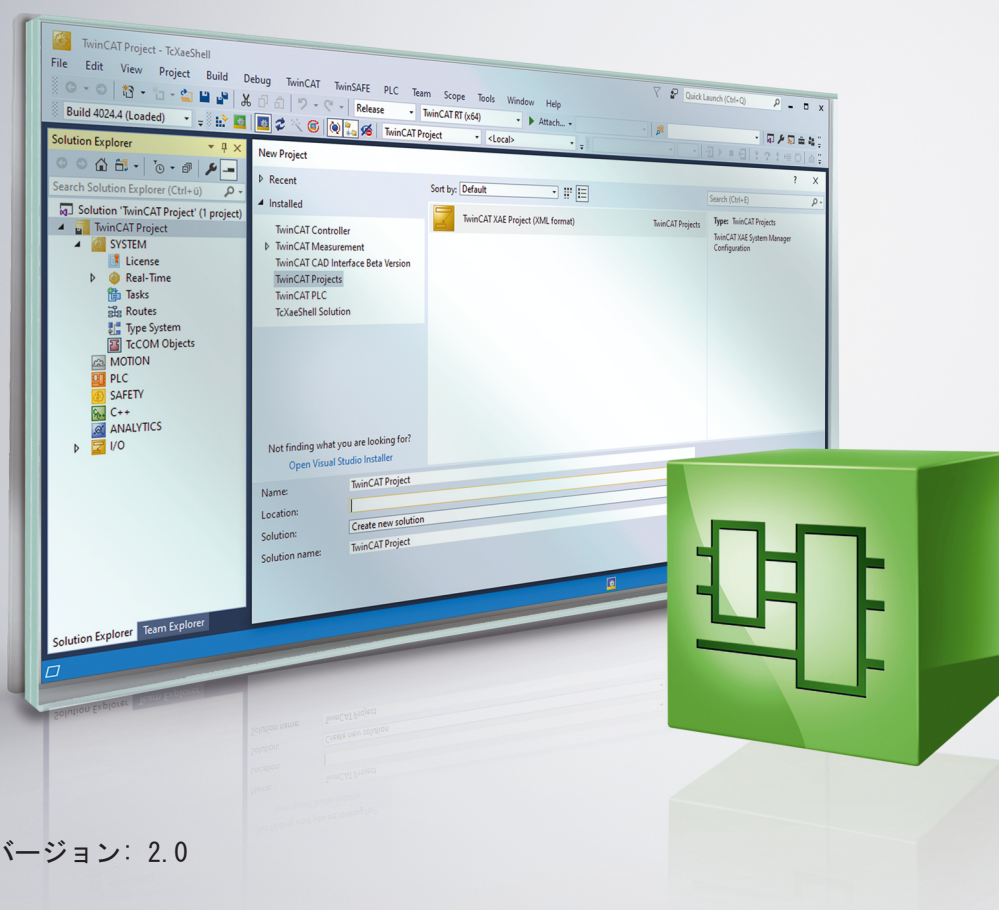


取扱説明書 | JA

TE1000

TwinCAT 3 | PLC Lib: Tc2_Utilities



目次

1	序文	13
1.1	取扱説明書に関する注記	13
1.2	安全に関する指示事項	13
2	概要	15
3	ファンクションブロック	25
3.1	BCD_TO_DEC	25
3.2	DCF77_TIME	26
3.3	DCF77_TIME_EX	29
3.4	DEC_TO_BCD	32
3.5	FB_AdsReadEvents	33
3.6	FB_AddRouteEntry	34
3.7	FB_AmsLogger	35
3.8	FB_BasicPID	38
3.9	FB_CheckLicense	39
3.10	FB_CSVMemBufferReader	40
3.11	FB_CSVMemBufferWriter	41
3.12	FB_EnumFindFileEntry	43
3.13	FB_EnumFindFileList	44
3.14	FB_EnumRouteEntry	46
3.15	FB_EnumStringNumbers	48
3.16	FB_FileRingBuffer	50
3.17	FB_FileTimeToTzSpecificLocalTime	52
3.18	FB_FormatString	55
3.19	FB_FormatString2	56
3.20	FB_GetAdaptersInfo	57
3.21	FB_GetDeviceIdentification	58
3.22	FB_GetDeviceIdentificationEx	59
3.23	FB_GetDongleSystemID	60
3.24	FB_GetHostAddrByName	61
3.25	FB_GetHostName	62
3.26	FB_GetLicenseDongle	63
3.27	FB_GetLicenses	64
3.28	FB_GetLicensesEx	65
3.29	FB_GetLocalAmsNetId	66
3.30	FB_GetRouterStatusInfo	67
3.31	FB_GetSystemId	68
3.32	FB_GetTimeZoneInformation	69
3.33	FB_GetVolumeId	70
3.34	FB_HashTableCtrl	71
3.35	FB_LicFileCopyFromDongle	72
3.36	FB_LicFileCopyToDongle	74
3.37	FB_LicFileCreate	75

3.38	FB_LicFileDelete	76
3.39	FB_LicFileGetStorageInfo	77
3.40	FB_LicFileRead	78
3.41	FB_LinkedListCtrl	79
3.42	FB_LocalSystemTime	81
3.43	FB_MemBufferMerge	84
3.44	FB_MemBufferSplit	86
3.45	FB_MemRingBuffer	87
3.46	FB_MemRingBufferEx	88
3.47	FB_MemStackBuffer	90
3.48	FB_RegQueryValue	92
3.49	FB_RegSetValue	95
3.50	FB_RemoveRouteEntry	97
3.51	FB_ScopeServerControl	98
3.52	FB_SetTimeZoneInformation	101
3.53	FB_StringRingBuffer	103
3.54	FB_SystemTimeToTzSpecificLocalTime	104
3.55	FB_TzSpecificLocalTimeToFileTime	106
3.56	FB_TzSpecificLocalTimeToSystemTime	108
3.57	FB_WritePersistentData	109
3.58	GetRemotePCInfo	110
3.59	NT_AbortShutdown	112
3.60	NT_GetTime	113
3.61	NT_Reboot	114
3.62	NT_SetLocalTime	115
3.63	NT_SetTimeToRTCTime	116
3.64	NT_Shutdown	117
3.65	NT_StartProcess	119
3.66	PLC_ReadSymInfo	120
3.67	PLC_ReadSymInfoByName	121
3.68	PLC_ReadSymInfoByNameEx	123
3.69	PLC_Reset	124
3.70	PLC_Start	125
3.71	PLC_Stop	126
3.72	プロファイラ	127
3.73	RTC	129
3.74	RTC_EX	130
3.75	RTC_EX2	131
3.76	TC_Config	132
3.77	TC_CpuUsage	133
3.78	TC_Restart	134
3.79	TC_Stop	135
3.80	TC_SysLatency	136
3.81	WritePersistentData	137

4	ファンクション	139
4.1	タイムファンクション	139
4.1.1	DT_TO_FILETIME	139
4.1.2	DT_TO_SYSTEMTIME	139
4.1.3	F_GetDayOfMonthEx	140
4.1.4	F_GetDayOfWeek	141
4.1.5	F_GetDOYOfYearMonthDay	141
4.1.6	F_GetMaxMonthDays	142
4.1.7	F_GetMonthOfDOY	143
4.1.8	F_GetWeekOfTheYear	144
4.1.9	F_TranslateFileTimeBias	144
4.1.10	F_YearIsLeapYear	146
4.1.11	FILETIME_TO_DT	146
4.1.12	FILETIME_TO_SYSTEMTIME	147
4.1.13	OTSTRUCT_TO_TIME	147
4.1.14	STRING_TO_SYSTEMTIME	148
4.1.15	SYSTEMTIME_TO_DT	148
4.1.16	SYSTEMTIME_TO_FILETIME	149
4.1.17	SYSTEMTIME_TO_STRING	150
4.1.18	TIME_TO_OTSTRUCT	151
4.2	拡張STRINGファンクション	151
4.2.1	CHAR_TO_WCHAR	151
4.2.2	CONCAT2	151
4.2.3	DELETE2	152
4.2.4	F_StringIsASCII	153
4.2.5	FIND2	154
4.2.6	FindAndDelete	154
4.2.7	FindAndDeleteChar	155
4.2.8	FindAndReplace	156
4.2.9	FindAndReplaceChar	157
4.2.10	INSERT2	157
4.2.11	LEN2	158
4.2.12	REPLACE2	159
4.2.13	sLiteral_TO_UTF8	159
4.2.14	STRING_TO_UTF8	160
4.2.15	STRING_TO_WSTRING2	161
4.2.16	STRNCPY	161
4.2.17	UTF8_TO_STRING	162
4.2.18	UTF8_TO_WSTRING	163
4.2.19	UTF8Len	164
4.2.20	WCHAR_TO_CHAR	165
4.2.21	WCONCAT2	165
4.2.22	WLEN2	166
4.2.23	wsLiteral_TO_UTF8	166
4.2.24	WSTRING_TO_STRING2	167

4.2.25	WSTRING_TO_UTF8	168
4.2.26	WSTRNCPY	168
4.3	バイトオーダー変換機能	169
4.3.1	ホストバイトオーダー/ネットワークバイトオーダー	169
4.3.2	HOST_TO_BE16	170
4.3.3	HOST_TO_BE32	170
4.3.4	HOST_TO_BE64	170
4.3.5	HOST_TO_BE64EX	171
4.3.6	HOST_TO_BE128	171
4.3.7	BE16_TO_HOST	172
4.3.8	BE32_TO_HOST	172
4.3.9	BE64_TO_HOST	173
4.3.10	BE64_TO_HOSTEX	173
4.3.11	BE128_TO_HOST	173
4.4	FLOATファンクション	174
4.4.1	BOOL_TO_FLOAT	174
4.4.2	DINT_TO_FLOAT	174
4.4.3	FLOAT_TO_BOOL	175
4.4.4	FLOAT_TO_DINT	175
4.4.5	FLOAT_TO_INT	175
4.4.6	FLOAT_TO_SINT	175
4.4.7	FLOAT_TO_STRING	176
4.4.8	FLOAT_TO_TIME	176
4.4.9	FLOAT_TO_UDINT	176
4.4.10	FLOAT_TO_UINT	177
4.4.11	INT_TO_FLOAT	177
4.4.12	SINT_TO_FLOAT	177
4.4.13	TIME_TO_FLOAT	177
4.4.14	UDINT_TO_FLOAT	178
4.4.15	UINT_TO_FLOAT	178
4.4.16	LrealIsFinite	178
4.4.17	LrealIsNaN	178
4.5	LCOMPLEXファンクション	179
4.5.1	LcomplexIsNaN	179
4.5.2	LcomplexAbs	180
4.6	P[TYPE]_TO_[TYPE]変換ファンクション	180
4.6.1	PBOOL_TO_BOOL	180
4.6.2	PBYTE_TO_BYTE	180
4.6.3	PDATE_TO_DATE	181
4.6.4	PDINT_TO_DINT	181
4.6.5	PDT_TO_DT	181
4.6.6	PDWORD_TO_DWORD	182
4.6.7	PHUGE_TO_HUGE	182
4.6.8	PINT_TO_INT	183
4.6.9	PLARGE_TO_LARGE	183

4. 6. 10	PLINT_TO_LINT	183
4. 6. 11	PLREAL_TO_LREAL	184
4. 6. 12	PLWORD_TO_LWORD	184
4. 6. 13	PMAXSTRING_TO_MAXSTRING	184
4. 6. 14	PREAL_TO_REAL	185
4. 6. 15	PSINT_TO_SINT	185
4. 6. 16	PSTRING_TO_STRING	185
4. 6. 17	PTIME_TO_TIME	186
4. 6. 18	PTOD_TO_TOD	186
4. 6. 19	PUDINT_TO_UDINT	187
4. 6. 20	PUHUGE_TO_UHUGE	187
4. 6. 21	PUINT_TO_UINT	187
4. 6. 22	PULARGE_TO_ULARGE	188
4. 6. 23	PULINT_TO_ULINT	188
4. 6. 24	PUSINT_TO_USINT	188
4. 6. 25	PWORD_TO_WORD	189
4. 6. 26	PUINT64_TO_UINT64	189
4. 7	16ビット固定小数点数ファンクション(符号付き)	190
4. 7. 1	FIX16_TO_LREAL	190
4. 7. 2	FIX16_TO_WORD	190
4. 7. 3	FIX16Add	191
4. 7. 4	FIX16Align	191
4. 7. 5	FIX16Div	192
4. 7. 6	FIX16Mul	193
4. 7. 7	FIX16Sub	193
4. 7. 8	LREAL_TO_FIX16	194
4. 7. 9	WORD_TO_FIX16	195
4. 8	64ビットファンクション(符号付き)	196
4. 8. 1	INT64_TO_LREAL	196
4. 8. 2	Int64Add64	196
4. 8. 3	Int64Add64Ex	196
4. 8. 4	Int64Cmp64	197
4. 8. 5	Int64Div64Ex	197
4. 8. 6	Int64IsZero	198
4. 8. 7	Int64Negate	198
4. 8. 8	Int64Not	199
4. 8. 9	Int64Sub64	199
4. 8. 10	LARGE_INTEGER	199
4. 8. 11	LARGE_TO_LINT	200
4. 8. 12	LARGE_TO_ULARGE	200
4. 8. 13	LINT_TO_LARGE	201
4. 8. 14	LREAL_TO_INT64	201
4. 8. 15	ULARGE_TO_LARGE	201
4. 9	64ビット整数ファンクション(符号なし)	202
4. 9. 1	LREAL_TO_UINT64	202

4.9.2	LWORD_TO_ULARGE	202
4.9.3	STRING_TO_UINT64	203
4.9.4	UInt32x32To64	203
4.9.5	UINT64_TO_LREAL	203
4.9.6	UINT64_TO_STRING	204
4.9.7	UInt64Add64	204
4.9.8	UInt64Add64Ex	205
4.9.9	UInt64And	205
4.9.10	UInt64Cmp64	206
4.9.11	UInt64Div16Ex	206
4.9.12	UInt64Div64	207
4.9.13	UInt64Div64Ex	207
4.9.14	UInt64isZero	208
4.9.15	UInt64Limit	208
4.9.16	UInt64Max	208
4.9.17	UInt64Min	209
4.9.18	UInt64Mod64	209
4.9.19	UInt64Mul64	210
4.9.20	UInt64Mul64Ex	210
4.9.21	UInt64Not	211
4.9.22	UInt64Or	211
4.9.23	UInt64Rol	211
4.9.24	UInt64Ror	212
4.9.25	UInt64Shl	212
4.9.26	UInt64Shr	213
4.9.27	UInt64Sub64	213
4.9.28	UInt64Xor	213
4.9.29	ULARGE_INTEGER	214
4.9.30	ULARGE_TO_ULINT	214
4.9.31	ULARGE_TO_LWORD	215
4.10	T_Argヘルプファンクション	215
4.10.1	F_ARGCMP	215
4.10.2	F_ARGCPY	216
4.10.3	F_ARGISZERO	217
4.10.4	F_BIGTYPE	217
4.10.5	F_BOOL	217
4.10.6	F_BYTE	218
4.10.7	F_DINT	218
4.10.8	F_DWORD	219
4.10.9	F_HUGE	219
4.10.10	F_INT	219
4.10.11	F_LARGE	220
4.10.12	F_LINT	220
4.10.13	F_LREAL	221
4.10.14	F_LWORD	221

4.10.15	F_REAL	221
4.10.16	F_SINT	222
4.10.17	F_STRING	222
4.10.18	F_STRINGEx	222
4.10.19	F_UDINT	223
4.10.20	F_UHUGE	223
4.10.21	F_UINT	224
4.10.22	F_ULARGE	224
4.10.23	F_ULINT	224
4.10.24	F_USINT	225
4.10.25	F_WORD	225
4.10.26	F_PVOID	225
4.10.27	IsFinite	226
4.11	[廃止]	227
4.11.1	ファンクションF_GetVersionToUtilities	227
4.11.2	FLOATIsFinite	228
4.11.3	FLOATIsNaN	228
4.12	ARG_TO_GSVFIELD	229
4.13	BYTE_TO_BINSTR	231
4.14	BYTE_TO_DECSTR	231
4.15	BYTE_TO_HEXSTR	232
4.16	BYTE_TO_LREALEX	232
4.17	BYTE_TO_OCTSTR	233
4.18	BYTEARR_TO_MAXSTRING	234
4.19	GSVFIELD_TO_ARG	234
4.20	GSVFIELD_TO_STRING	235
4.21	DATA_TO_HEXSTR	237
4.22	DEG_TO_RAD	238
4.23	DINT_TO_DECSTR	238
4.24	DWORD_TO_BINSTR	239
4.25	DWORD_TO_DECSTR	240
4.26	DWORD_TO_HEXSTR	241
4.27	DWORD_TO_LREALEX	242
4.28	DWORD_TO_OCTSTR	243
4.29	F_BYTE_TO_CRC16_CCITT	244
4.30	F_CheckSum16	244
4.31	F_CreateHashTableHnd	245
4.32	F_CreateLinkedListHnd	246
4.33	F_DATA_TO_CRC16_CCITT	247
4.34	F_FormatArgToStr	247
4.35	F_LTrim	250
4.36	F_RTrim	250
4.37	F_SwapRealEx	251
4.38	F_ToLCCase	252
4.39	F_ToUCCase	252

4.40	GUID_TO_REGSTRING	253
4.41	GUID_TO_STRING	254
4.42	GuidsEqualByVal	254
4.43	HEXASCNIBBLE_TO_BYTE	255
4.44	HEXCHRNIBBLE_TO_BYTE	255
4.45	HEXSTR_TO_DATA	256
4.46	LINT_TO_DECSTR	257
4.47	LREAL_TO_FMTSTR	258
4.48	LWORD_TO_BINSTR	259
4.49	LWORD_TO_DECSTR	260
4.50	LWORD_TO_HEXSTR	261
4.51	LWORD_TO_OCTSTR	262
4.52	MAXSTRING_TO_BYTEARR	262
4.53	PVOID_TO_BINSTR	263
4.54	PVOID_TO_DECSTR	264
4.55	PVOID_TO_HEXSTR	265
4.56	PVOID_TO_OCTSTR	266
4.57	PVOID_TO_STRING	267
4.58	RAD_TO_DEG	268
4.59	REGSTRING_TO_GUID	269
4.60	ROUTETRANSPORT_TO_STRING	269
4.61	STRING_TO_CSVFIELD	270
4.62	STRING_TO_GUID	271
4.63	STRING_TO_PVOID	272
4.64	UDINT_TO_LREALEX	272
4.65	UINT_TO_LREALEX	273
4.66	ULINT_TO_ULARGE	274
4.67	USINT_TO_LREALEX	275
4.68	WORD_TO_BINSTR	276
4.69	WORD_TO_DECSTR	276
4.70	WORD_TO_HEXSTR	277
4.71	WORD_TO_LREALEX	277
4.72	WORD_TO_OCTSTR	278
5	データ型	280
5.1	ADSDATATYPEID	280
5.2	E_AmsLoggerMode	281
5.3	E_ArgType	281
5.4	E_DbgContext	282
5.5	E_DbgDirection	282
5.6	E_EnumCmdType	283
5.7	E_LDevType	283
5.8	E_LDongleStatus	284
5.9	E_LicenseHResult	284
5.10	E_MIB_IF_Type	285

5.11	E_NumGroupTypes	285
5.12	E_PersistentMode	286
5.13	E_RegValueType	286
5.14	E_RouteTransportType	287
5.15	E_SBCSType	288
5.16	E_ScopeServerState	288
5.17	E_TimeZoneID	288
5.18	E_TypeFieldParam	289
5.19	FLOAT	289
5.20	GUID	290
5.21	OTSTRUCT	290
5.22	PROFILERSTRUCT	290
5.23	REMOTEPC	291
5.24	REMOTEPCINFOSTRUCT	291
5.25	ST_AmsRouteEntry	291
5.26	ST_CheckLicense	292
5.27	ST_DeviceIdentification	292
5.28	ST_DeviceIdentificationEx	293
5.29	ST_FileAttributes	294
5.30	ST_FileRBufferHead	295
5.31	ST_FindFileEntry	296
5.32	ST_IPAdapterHwAddr	297
5.33	ST_IPAdapterInfo	297
5.34	ST_LicenseDongle	298
5.35	ST_TcOnlineLicenseInfoDataEx	298
5.36	ST_TcOnlineLicensesInfoData	299
5.37	ST_TcRouterStatusInfo	300
5.38	ST_TimeZoneInformation	300
5.39	SYMINFOSTRUCT	301
5.40	T_Arg	302
5.41	T_FILETIME	302
5.42	T_FIX16	302
5.43	T_HashTableEntry	303
5.44	T_HHASHTABLE	304
5.45	T_HLINKEDLIST	304
5.46	T_HUGE_INTEGER	304
5.47	T_LARGE_INTEGER	305
5.48	T_LinkedListEntry	305
5.49	T_UHUGE_INTEGER	305
5.50	T_ULARGE_INTEGER	306
5.51	TIMESTRUCT	306
6	グローバル定数	308
6.1	ライブラリバージョン	308
7	グローバル変数	309

8 サンプル	310
8.1 例: 通信BC/BX<->PC/CX (F_SwapRealEx)	310
8.2 例: ファイル検索 (FB_EnumFindFileEntry、FB_EnumFindFileList)	315
8.3 例: ファイルリングFIFO (FB_FileRingBuffer)	317
8.4 例: メモリリングFiFo (FB_MemRingBuffer)	319
8.5 例: メモリリングFiFo (FB_MemRingBufferEx)	321
8.6 例: ハッシュテーブル (FB_HashTableCtrl)	322
8.7 例: リンクされたリスト (FB_LinkedListCtrl)	327
8.8 例: CSVファイルの書き込み/読み込み	333
8.9 例: ソフトウェアクロック (RTC、RTC_EX、RTC_EX2)	336
9 付録	338
9.1 永続データを書き込む場合する場合のシステム動作	338
9.2 フォーマット仕様	338
9.3 フォーマットエラーコード	341
9.4 スコープサーバエラーコード	341
9.5 ADSリターンコード	342
9.6 Win32エラーコード	345

1 序文

1.1 取扱説明書に関する注記

この説明は対応する国内規格を熟知した、トレーニングを受けた制御、オートメーションエンジニアリングの専門技術者のみの使用を対象としています。

コンポーネントのインストールとコミッショニングの際には、取扱説明書および以下の注意事項と説明に従うことが重要です。

技術者には各設置およびコミッショニングのそれぞれの時点で、発行された取扱説明書を使用する義務があります。

本製品を使用するうえでの責任者は、本製品の用途および使用方法が、関連するすべての法律、法規、ガイドラインおよび規格を含む、安全に関するすべての要件を満たしていることを確認してください。

免責事項

この取扱説明書の記載内容は、一般的な製品説明および性能を記載したものであり、場合により記載通りに動作しないことがあります。

製品の情報・仕様は予告なく変更されます。

この説明書に記載されているデータ、図および説明に基づいて、すでに納品されている製品の変更を要求することはできません。

商標

Beckhoff®、TwinCAT®、EtherCAT®、EtherCAT G®、EtherCAT G10®、EtherCAT P®、Safety over EtherCAT®、TwinSAFE®、XFC®、XTS®およびXPlanar®は、Beckhoff Automation GmbHの登録商標です。

この取扱説明書で使用されているその他の名称は商標である可能性があり、第三者が独自の目的のために使用すると所有者の権利を侵害する可能性があります。

特許出願

EtherCAT Technologyについては、欧州特許 EP1590927、EP1789857、EP1456722およびEP2137893、ドイツ特許DE102015105702に記載されていますが、これらに限定されるものではありません。

EtherCAT®

EtherCAT®は、Beckhoff Automation GmbH (ドイツ)によりライセンスを受けた登録商標および特許技術です。

著作権

© Beckhoff Automation GmbH & Co. KG, Germany.

明示的な許可なく、本書の複製、配布、使用、および他への内容の転載は禁止されています。

これに違反した者は損害賠償の責任を負います。すべての権利は、特許、実用新案、意匠の付与の際に留保されます。

1.2 安全に関する指示事項

安全に関する注意事項

この取扱説明書に記載された安全に関する指示や注意事項はよくお読みになり、必ず指示に従ってください。

納入仕様

すべての製品は、用途に適した特定のハードウェア構成およびソフトウェア構成を有する状態で供給されます。ハードウェアまたはソフトウェアに取扱説明書に記載されている以外の変更を加えることは許可されていません。許可されていない変更を加えると、Beckhoff Automation GmbH & Co. KGの保証の対象外となります。

使用者の資格

この説明書は関連する国内法規を熟知した、制御およびオートメーションエンジニアリングの専門家の使用を目的としています。

安全記号の説明

この取扱説明書では、安全に関する指示や注意事項とともに以下の安全記号を使用します。安全に関する指示事項はよくお読みになり、必ず指示に従ってください。

⚠ 危険

重大な人的傷害の危険

この記号が付いた安全に関する注意事項に従わないと、人命および健康に直ちに危害を及ぼします。

⚠ 警告

人的傷害の危険

この記号が付いた安全に関する注意事項に従わないと、人命および健康に危険を及ぼします。

⚠ 注意

人的傷害の恐れ

この記号が付いた安全に関する注意事項に従わないと、人命および健康に危険を及ぼす恐れがあります。

📌 注記

物的損害と環境汚染

この記号が付いた安全に関する注意事項に従わないと、物的損害と環境汚染をもたらす恐れがあります。

● ヒントまたはアドバイス

i この記号が示す情報により、さらに理解が深まります。

2 概要

PLC library Tc2_Uutilitiesは、TwinCATシステムのファンクション、OSの機能、さまざまな変換機能をコールするファンクションブロックおよび機能が含まれています。

- ・ [OSのファンクション \[▶ 15\]](#)
- ・ [PLC機能 \[▶ 16\]](#)
- ・ [チェックサム/CRCファンクション \[▶ 16\]](#)
- ・ [システムファンクション \[▶ 16\]](#)
- ・ [スコープビューファンクション \[▶ 16\]](#)
- ・ [スコープサーバファンクション \[▶ 17\]](#)
- ・ [ADSモニタファンクション \[▶ 17\]](#)
- ・ [変換機能 \[▶ 17\]](#)
- ・ [STRINGファンクション \[▶ 17\]](#)
- ・ [拡張STRINGファンクション \[▶ 18\]](#)
- ・ [64ビットファンクション\(符号なし\) \[▶ 19\]](#)
- ・ [64ビットファンクション\(符号付き\) \[▶ 20\]](#)
- ・ [16ビット固定小数点数ファンクション\(符号付き\) \[▶ 20\]](#)
- ・ [バイトオーダー変換機能 \[▶ 20\]](#)
- ・ [FLOATファンクション \[▶ 21\]](#)
- ・ [LCOMPLEXファンクション \[▶ 21\]](#)
- ・ [P\[TYPE\] TO \[TYPE\] 変換ファンクション \[▶ 21\]](#)
- ・ [T Argヘルプファンクション \[▶ 22\]](#)
- ・ [CSV形式ファンクション \[▶ 22\]](#)
- ・ [ライセンス機能 \[▶ 23\]](#)
- ・ [他のファンクション \[▶ 23\]](#)

OSのファンクション

名前	説明
NT_Shutdown [▶ 117]	OSのシャットダウン(シャットダウン)。
NT_AbortShutdown [▶ 112]	シャットダウンプロセスの中断。
NT_Reboot [▶ 114]	OSの再起動。
NT_GetTime [▶ 113]	現在のローカルWindowsシステム時間の読み込み。
NT_SetLocalTime [▶ 115]	現在のローカルWindowsシステム時間のセット。
NT_StartProcess [▶ 119]	PLCからWindowsアプリケーションを起動します。
NT_SetTimeToRTCTime [▶ 116]	PCのリアルタイムクロックとローカルWindowsシステム時間の同期。
FB_RegQueryValue [▶ 92]	レジストリから値を読み込み。
FB_RegSetValue [▶ 95]	レジストリに値を書き込み。
FB_EnumFindFileEntry [▶ 43]	指定した名前と類似した名前のファイルまたはサブディレクトリのディレクトリ内を検索します。検出されたすべてのエントリは、個々に読み込みできます。
FB_EnumFindFileList [▶ 44]	指定した名前と類似した名前のファイルまたはサブディレクトリのディレクトリ内を検索します。検出されたエントリは、グループで読み出しできます。
FB_GetAdaptersInfo [▶ 57]	ネットワークアダプタ情報を読み込みします。
FB_GetHostName [▶ 62]	ローカルPCのホスト名を読み込みします。
FB_GetHostAddrByName [▶ 61]	ホスト名を(IPv4)インターネットプロトコルのネットワークアドレスに変換します。

名前	説明
FB GetTimeZoneInformation [▶ 69]	OSのタイムゾーン設定を読み込みします。
FB SetTimeZoneInformation [▶ 101]	OSのタイムゾーン設定をセットします。
FB LocalSystemTime [▶ 81]	ローカルWindowsシステム時間と夏時間/冬時間を返します。

PLC機能

名前	説明
PLC Reset [▶ 124]	PLCをリセットします。
PLC Start [▶ 125]	PLCを起動します。
PLC Stop [▶ 126]	PLCを停止します。
PLC ReadSymInfo [▶ 120]	PLCのシンボル情報の読み込み。
PLC ReadSymInfoByName [▶ 121]	シンボル名でPLC変数のシンボル情報を読み込み。
PLC ReadSymInfoByNameEx [▶ 123]	シンボル名でPLC変数のシンボル情報を読み込み。使用可能なバッファサイズが十分でない場合、コメントは切り捨てられます。
プロファイラ [▶ 127]	PLCコードの実行時間の判定。
WritePersistentData [▶ 137]	PLCプログラムからデータストレージデバイスへの永続データの保存。
FB WritePersistentData [▶ 109]	PLCプログラムからデータストレージデバイスへの永続データの保存(拡張バージョン)。

チェックサム/CRCファンクション

名前	説明
F CheckSum16 [▶ 244]	16ビットチェックサムを計算します。
F DATA TO CRC16 CCITT [▶ 247]	任意のデータ型のCRC16-CCITT (巡回冗長性検査)を計算します。
F BYTE TO CRC16 CCITT [▶ 244]	個々のデータバイトのCRC16-CCITT (巡回冗長性検査)を計算します。

システムファンクション

名前	説明
TC Restart [▶ 134]	TwinCATシステムを再起動します。
TC Stop [▶ 135]	TwinCATシステムを停止します。
TC Config [▶ 132]	TwinCATシステムをConfigモードに切り替え。
TC CpuUsage [▶ 133]	TwinCATシステムのCPU使用率の判定。
TC SysLatency [▶ 136]	TwinCATシステムの現在の時間および最大遅延時間の判定
GetRemotePCInfo [▶ 110]	リモートPC経由のルータ情報の読み込み。
FB GetLocalAmsNetId [▶ 66]	ローカルTwinCAT PCのAmsNetIdの読み込み。
FB GetRouterStatusInfo [▶ 67]	TwinCATルータ ステータス情報の読み込み。
FB EnumRouteEntry [▶ 46]	ルータ接続情報の読み込み。
FB AddRouteEntry [▶ 34]	新しいルータ接続の追加。
FB RemoveRouteEntry [▶ 97]	既存のルータ接続の削除。
ROUTETRANSPORT TO STRING [▶ 269]	AMSメッセージルータのトランスポート層IDを文字列に変換します。
FB GetDeviceIdentification [▶ 58]	デバイスIDを読み込みします。
FB GetDeviceIdentificationEx [▶ 59]	デバイスIDを読み込みします。(ハードウェア型式とシリアル番号含む長い文字列に対応)。
FB GetLicences [▶ 64]	有効および無効なTwinCATライセンスを読み込みします。
FB GetSystemId [▶ 68]	SystemIDをGUID値で読み込みします。
FB GetVolumeId [▶ 70]	SystemIDとボリュームシステムIDを読み込みします。

スコープビューファンクション

TwinCAT Scope Viewファンクションは、TwinCAT 3ではサポートしていません。

スコープサーバファンクション

名前	説明
FB_ScopeServerControl [▶ 98]	データロギング用にスコープサーバを制御します(開始/保存..)。

ADSモニタファンクション

名前	説明
FB_AmsLogger [▶ 35]	PLCからのAMS Loggerの起動/停止。

変換機能

名前	説明
DT_TO_SYSTEMTIME [▶ 139]	DATE_AND_TIMEをWindowsシステム時間構造体に変換。
DT_TO_FILETIME [▶ 139]	DATE_AND_TIMEをWindowsファイル時間に変換。
SYSTEMTIME_TO_DT [▶ 148]	Windowsシステム時間構造体をDATE_AND_TIMEに変換。
SYSTEMTIME_TO_FILETIME [▶ 149]	Windowsシステム時間構造体をファイル時間に変換。
SYSTEMTIME_TO_STRING [▶ 150]	Windowsシステム時間構造体を文字列に変換。
STRING_TO_SYSTEMTIME [▶ 148]	文字列をWindowsシステム時間構造体に変換。
FILETIME_TO_DT [▶ 146]	Windowsのファイル時間をDATE_AND_TIMEに変換。
FILETIME_TO_SYSTEMTIME [▶ 147]	Windowsのファイル時間をシステム時間構造体に変換。
DEC_TO_BCD [▶ 32]	小数をBCD数に変換。
BCD_TO_DEC [▶ 25]	BCD数を小数に変換。
DEG_TO_RAD [▶ 238]	度単位の角度をラジアンに変換。
RAD_TO_DEG [▶ 268]	ラジアンを角度に変換。
TIME_TO_OTSTRUCT [▶ 151]	TIME変数をミリ秒、秒、分に分割された構造体に変換。
OTSTRUCT_TO_TIME [▶ 147]	ミリ秒、秒、分に分割された構造体をTIME変数に変換。
F_SwapRealEx [▶ 251]	REAL型変数のHiおよびLoワードに置き換え
BYTE_TO_LREALX [▶ 232]	LREAL型の正の浮動小数点数へのBYTE型の明示的な変換を可能にします。
DWORD_TO_LREALX [▶ 242]	LREAL型の正の浮動小数点数へのDWORD型の明示的な変換を可能にします。
UDINT_TO_LREALX [▶ 272]	LREAL型の正の浮動小数点数へのUDINT型の明示的な変換を可能にします。
UINT_TO_LREALX [▶ 273]	LREAL型の正の浮動小数点数へのUINT型の明示的な変換を可能にします。
ULINT_TO_ULARGE [▶ 274]	ULINT型の64ビット数をT_ULARGE_INTEGER型の64ビット数に変換します。
USINT_TO_LREALX [▶ 275]	LREAL型の正の浮動小数点数へのUSINT型の明示的な変換を可能にします。
BYTEARR_TO_MAXSTRING [▶ 234]	バイト列を文字列に変換します。
MAXSTRING_TO_BYTEARR [▶ 262]	文字列をバイト列に変換します。
F_TranslateFileTimeBias [▶ 144]	UTC時間を現地時間に変換します。またはその逆(バイアスによって)。
FB_TzSpecificLocalTimeToFileTime [▶ 106]	連続現地時間(ファイル時間の形式)をUTC時間に変換します。
FB_TzSpecificLocalTimeToSystemTime [▶ 108]	連続現地時間(構造体システム時間の形式)をUTC時間に変換します。
FB_FileTimeToTzSpecificLocalTime [▶ 52]	UTC時間(ファイル時間の形式)を現地時間に変換します。
FB_SystemTimeToTzSpecificLocalTime [▶ 104]	UTC時間(構造体システム時間の形式)を現地時間に変換します。
HEXASCNIBBLE_TO_BYTE [▶ 255]	16進ハーファイト文字のASCIIコードを10進数に変換します。
HEXCHRNibble_TO_BYTE [▶ 255]	16進ハーファイト文字を10進数に変換します。
GuidsEqualByVal [▶ 254]	2つのGUID値を比較します。

STRINGファンクション

名前	説明
LREAL_TO_FMTSTR [▶ 258]	浮動小数点数を必要な少数点以下の桁数の文字列に変換します。
DWORD_TO_DECSTR [▶ 240]	10進数を10進数文字列に変換します。

名前	説明
DWORD_TO_HEXSTR [▶ 241]	10進数を16進数文字列に変換します。
DWORD_TO_OCTSTR [▶ 243]	10進数を8進数文字列に変換します。
DWORD_TO_BINSTR [▶ 239]	10進数をバイナリ文字列に変換します。
LWORD_TO_DECSTR [▶ 260]	10進数を10進数文字列に変換します。
LWORD_TO_HEXSTR [▶ 261]	10進数を16進数文字列に変換します。
LWORD_TO_OCTSTR [▶ 262]	10進数を8進数文字列に変換します。
LWORD_TO_BINSTR [▶ 259]	10進数をバイナリ文字列に変換します。
PVOID_TO_DECSTR [▶ 264]	アドレス(ポインタ)を10進数文字列に変換します。
PVOID_TO_HEXSTR [▶ 265]	アドレス(ポインタ)を16進数文字列に変換します。
PVOID_TO_OCTSTR [▶ 266]	アドレス(ポインタ)を8進数文字列に変換します。
PVOID_TO_BINSTR [▶ 263]	アドレス(ポインタ)をバイナリ文字列に変換します。
PVOID_TO_STRING [▶ 267]	アドレス(ポインタ)を文字列に変換します。
STRING_TO_PVOID [▶ 272]	文字列をアドレス(ポインタ)に変換します。
LINT_TO_DECSTR [▶ 257]	符号付き10進数(64ビット)を10進数文字列に変換します。
DINT_TO_DECSTR [▶ 238]	符号付き10進数(32ビット)を10進数文字列に変換します。
F_FormatArgToStr [▶ 247]	10進数または浮動小数点数を文字列に変換しフォーマットします。
BYTE_TO_BINSTR [▶ 231]	BYTE型の10進数をバイナリ文字列に変換します。
BYTE_TO_DECSTR [▶ 231]	10進数を10進数文字列に変換します。
BYTE_TO_HEXSTR [▶ 232]	10進数を16進数文字列に変換します。
BYTE_TO_OCTSTR [▶ 233]	10進数を8進数文字列に変換します。
WORD_TO_BINSTR [▶ 276]	WORD型の10進数をバイナリ文字列に変換します。
WORD_TO_DECSTR [▶ 276]	WORD型の10進数を10進数文字列に変換します。
WORD_TO_HEXSTR [▶ 277]	WORD型の10進数を16進数文字列に変換します。
WORD_TO_OCTSTR [▶ 278]	WORD型の10進数を8進数文字列に変換します。
FB_FormatString [▶ 55]	最大10個の引数(10進数または浮動小数)を変換し、フォーマットします。
FB_EnumStringNumbers [▶ 48]	文字列内の数値部分を検索しその文字列を列挙します。
F_ToUpperCase [▶ 252]	文字列の小文字を大文字に変換します。
F_ToLowerCase [▶ 252]	文字列の大文字を小文字に変換します。
F_LTrim [▶ 250]	文字列の最初のスペースを削除します。
F_RTrim [▶ 250]	文字列の最後のスペースを削除します。
DATA_TO_HEXSTR [▶ 237]	バイナリデータを16進数文字列に変換します。
HEXSTR_TO_DATA [▶ 256]	16進数文字列をバイナリデータに変換します。
GUID_TO_STRING [▶ 254]	構造体GUID変数をGUID文字列変数に変換します。
GUID_TO_REGSTRING [▶ 253]	構造体GUID変数をレジストリGUID文字列変数に変換します。
REGSTRING_TO_GUID [▶ 269]	レジストリGUID文字列変数を構造体GUID変数に変換します。

拡張STRINGファンクション

名前	説明
CHAR_TO_WCHAR [▶ 151]	データ型STRINGの文字をデータ型WSTRINGの文字に変換します (null 終端)。
CONCAT2 [▶ 151]	データ型STRINGの2つの文字列を連結します。
DELETE2 [▶ 152]	文字列の位置nPosを開始点としてnLen文字を削除します。
F_StringIsASCII [▶ 153]	文字列がASCII文字(0x000~0x7F)のみを含んでいるかチェックし、ASCII文字の数を返します。
FIND2 [▶ 154]	他の文字列で1回以上出現している文字列を検出します。
FindAndDelete [▶ 154]	他の文字列で1回以上出現している文字列を検出し、削除します。
FindAndDeleteChar [▶ 155]	文字列で1回以上出現している文字列を検出し、削除します。
FindAndReplace [▶ 156]	他の文字列で1回以上出現している文字列を検出し、他の文字列と置換します。
FindAndReplaceChar [▶ 157]	文字列で1回以上出現している文字列を検出し、他の文字と置換します。

名前	説明
INSERT2 [▶ 157]	文字列を他の文字列の位置nposの後に挿入します。
LEN2 [▶ 158]	文字列の文字数を返します。
REPLACE2 [▶ 159]	文字列のnLen文字を他の文字列の位置nposから始まる文字と置換します。
sLiteral TO UTF8 [▶ 159]	データ型STRINGの任意の文字列をUTF-8形式の文字列に変換します。
STRING TO UTF8 [▶ 160]	データ型STRINGの変数の任意の文字列をUTF-8形式の文字列に変換します。
STRING TO WSTRING2 [▶ 161]	データ型STRINGの変数をデータ型WSTRINGの変数に変換します。
STRNCPY [▶ 161]	データ型STRINGの変数の文字列をコピーし、文字列をすべてコピーしたかどうかをチェックします。
UTF8 TO STRING [▶ 162]	UTF-8形式の文字列をデータ型STRINGの文字列に変換します。
UTF8 TO WSTRING [▶ 163]	UTF-8形式の文字列をデータ型WSTRINGの文字列に変換します。
UTF8Len [▶ 164]	UTF-8文字列の文字数を返します。
WCHAR TO CHAR [▶ 165]	データ型WSTRINGの変数をデータ型STRINGの変数に変換します (null終端)。
WCONCAT2 [▶ 165]	任意の長さのデータ型WSTRINGの2つの文字列を連結します。
WLEN2 [▶ 166]	データ型WSTRINGのUnicode文字列で文字数を返します。
wsLiteral TO UTF8 [▶ 166]	データ型WSTRINGの文字列をUTF-8形式の文字列に変換します。
WSTRING TO STRING2 [▶ 167]	データ型WSTRINGの変数をデータ型STRINGの変数に変換します。
WSTRING TO UTF8 [▶ 168]	データ型WSTRINGの変数の文字列をUTF-8形式の文字列に変換します。
WSTRNCPY [▶ 168]	データ型WSTRINGの変数の文字列をコピーし、文字列をすべてコピーしたかどうかをチェックします。

64ビットファンクション(符号なし)

名前	説明
ULARGE_INTEGER [▶ 214]	64ビット数を初期化/セットします。
UInt64Add64 [▶ 204]	2つの64ビット数を加算します。
UInt64Add64Ex [▶ 205]	2つの64ビット数を加算します (オーバーフローチェックを使用)。
UInt64Sub64 [▶ 213]	2つの64ビット数を減算します。
UInt64Cmp64 [▶ 206]	2つの64ビット数を比較します。
UInt32x32To64 [▶ 203]	2つの32ビット数を乗算します。結果は64ビット数になります。
UInt64Mul64 [▶ 210]	2つの64ビット数を乗算します。結果は64ビット数になります。
UInt64Mul64Ex [▶ 210]	2つの64ビット数を乗算します。結果は64ビット数になります (オーバーフローチェックを使用)。
UInt64Div64 [▶ 207]	2つの64ビット数の除算。
UInt64Div64Ex [▶ 207]	2つの64ビット数の除算 (剰余付き)。
Uint64Div16Ex [▶ 206]	1つの64ビット数を16ビット数で除算。結果は64ビット数になります。
UInt64Mod64 [▶ 209]	2つの64ビット数のモジュロ除算。
UInt64And [▶ 205]	2つの64ビット数のビット単位のAND演算。
UInt64Or [▶ 211]	2つの64ビット数のビット単位のOR演算。
UInt64Not [▶ 211]	1つの64ビット数のビット単位のNOT演算。
UInt64Xor [▶ 213]	2つの64ビット数のビット単位のXOR演算。
UInt64Rol [▶ 211]	1つの64ビット数のビット単位の左回転。
UInt64Ror [▶ 212]	1つの64ビット数のビット単位の右回転。
UInt64Shl [▶ 212]	1つの64ビット数のビット単位の左シフト。
UInt64Shr [▶ 213]	1つの64ビット数のビット単位の右シフト。
UInt64Min [▶ 209]	最小ファンクション
UInt64Max [▶ 208]	最大ファンクション
UInt64Limit [▶ 208]	制限
UInt64isZero [▶ 208]	64ビット数の値がゼロかどうかをチェックします。
UINT64 TO STRING [▶ 204]	64ビット数を文字列に変換します。

名前	説明
UINT64_TO_LREAL [▶ 203]	64ビット数をLREALに変換します。
STRING_TO_UINT64 [▶ 203]	STRINGを64ビット数に変換します。
LREAL_TO_UINT64 [▶ 202]	LREALを64ビット数に変換します。
LWORD_TO_ULARGE [▶ 202]	LWORD型の64ビット数をT_ULARGE_INTEGER型の64ビット数に変換します。
ULARGE_TO_LWORD [▶ 215]	T_ULARGE_INTEGER型の64ビット数をLWORD型の64ビット数に変換します。
ULARGE_TO_ULINT [▶ 214]	T_ULARGE_INTEGER型の64ビット数をULINT型の64ビット数に変換します。

64ビットファンクション (符号付き)

名前	説明
LARGE_INTEGER [▶ 199]	64ビット数を初期化/セットします。
Int64Add64 [▶ 196]	2つの64ビット数を加算します。
Int64Add64Ex [▶ 196]	2つの64ビット数を加算します(オーバーフローチェックを使用)。
Int64Sub64 [▶ 199]	2つの64ビット数を減算します。
Int64Cmp64 [▶ 197]	2つの64ビット数を比較します。
Int64Div64Ex [▶ 197]	2つの64ビット数の除算(剰余付き)。
Int64Not [▶ 199]	1つの64ビット数のビット単位のNOT演算。
Int64isZero [▶ 198]	64ビット数の値がゼロかどうかをチェックします。
Int64Negate [▶ 198]	64ビット数を反転します。
INT64_TO_LREAL [▶ 196]	64ビット数をLREALに変換します。
LREAL_TO_INT64 [▶ 201]	LREALを64ビット数に変換します。
LARGE_TO_ULARGE [▶ 200]	符号付き64ビット数を符号なし64ビット数に変換します。
ULARGE_TO_LARGE [▶ 201]	符号なし64ビット数を符号付き64ビット数に変換します。
LARGE_TO_LINT [▶ 200]	LINT型の符号付き64ビット数をT_LARGE_INTEGER型の符号付き64ビット数に変換します。
LINT_TO_LARGE [▶ 201]	T_LARGE_INTEGER型の符号付き64ビット数をLINT型の符号付き64ビット数に変換します。

16ビット固定小数点数ファンクション(符号付き)

名前	説明
FIX16Add [▶ 191]	2つの固定小数点数を加算します。
FIX16Align [▶ 191]	固定小数点数の精度を変更します。
FIX16Sub [▶ 193]	2つの固定小数点数を減算します。
FIX16Div [▶ 192]	2つの固定小数点数を除算します。
FIX16Mul [▶ 193]	2つの固定小数点数を乗算します。
LREAL_TO_FIX16 [▶ 194]	LREALを固定小数点数に変換します。
WORD_TO_FIX16 [▶ 195]	WORDを固定小数点数に変換します。
FIX16_TO_LREAL [▶ 190]	固定小数点数をLREALに変換します。
FIX16_TO_WORD [▶ 190]	固定小数点数をWORDに変換します。

バイトオーダー変換機能

名前	説明
HOST_TO_BE16 [▶ 170]	ホストバイトオーダーからネットワークバイトオーダーに変換(16ビット数)
HOST_TO_BE32 [▶ 170]	ホストバイトオーダーからネットワークバイトオーダーに変換(32ビット数)。
HOST_TO_BE64 [▶ 170]	「レガシー」T_ULARGE_INTEGER型→「レガシー」型 : T_ULARGE_INTEGER
HOST_TO_BE64EX [▶ 171]	「ネイティブ」LWORD型→「ネイティブ」型 : LWORD
HOST_TO_BE128 [▶ 171]	「レガシー」T_UHUGE_INTEGER→「レガシー」型 : T_UHUGE_INTEGER

名前	説明
BE16_TO_HOST [▶ 172]	ホストバイトオーダーからネットワークバイトオーダーに変換(16ビット数)
BE32_TO_HOST [▶ 172]	ネットワークバイトオーダーからホストバイトオーダーに変換(32ビット数)
BE64_TO_HOST [▶ 173]	「レガシー」 T_ULARGE_INTEGER型→「レガシー」型 : T_ULARGE_INTEGER
BE64_TO_HOSTEX [▶ 173]	「ネイティブ」 LWORD型→「ネイティブ」型 : LWORD
BE128_TO_HOST [▶ 173]	「レガシー」 T_UHUGE_INTEGER→「レガシー」型 : T_UHUGE_INTEGER

FLOATファンクション

名前	説明
BOOL_TO_FLOAT [▶ 174]	BOOL型の変数をLREAL型の変数に変換します。
DINT_TO_FLOAT [▶ 174]	DINT型の変数をFLOAT型の変数に変換します。
FLOAT_TO_BOOL [▶ 175]	FLOAT型の変数をBOOL型の変数に変換します。
FLOAT_TO_DINT [▶ 175]	FLOAT型の変数をDINT型の変数に変換します。
FLOAT_TO_INT [▶ 175]	FLOAT型の変数をINT型の変数に変換します。
FLOAT_TO_SINT [▶ 175]	FLOAT型の変数をSINT型の変数に変換します。
FLOAT_TO_STRING [▶ 176]	FLOAT型の変数をSTRING型の変数に変換します。
FLOAT_TO_TIME [▶ 176]	FLOAT型の変数をTIME型の変数に変換します。
FLOAT_TO_UDINT [▶ 176]	FLOAT型の変数をUDINT型の変数に変換します。
FLOAT_TO_UINT [▶ 177]	FLOAT型の変数をUINT型の変数に変換します。
INT_TO_FLOAT [▶ 177]	INT型の変数をFLOAT型の変数に変換します。
SINT_TO_FLOAT [▶ 177]	SINT型の変数をFLOAT型の変数に変換します。
TIME_TO_FLOAT [▶ 177]	TIME型の変数をFLOAT型の変数に変換します。
UDINT_TO_FLOAT [▶ 178]	UDINT型の変数をFLOAT型の変数に変換します。
UINT_TO_FLOAT [▶ 178]	UINT型の変数をFLOAT型の変数に変換します。
LrealIsFinite [▶ 178]	LREAL型の引数が有限値を持っている場合にTRUEを返します。
LrealIsNaN [▶ 178]	LREAL型の引数が未定義の値(NaN)を持っている場合にTRUEを返します。

LCOMPLEXファンクション

LcomplexIsNaN [▶ 179]	LCOMPLEX型の引数が未定義の値(NaN)を持っている場合にTRUEを返します。
LcomplexAbs [▶ 180]	送信された複素数の絶対値を返します。

P[TYPE]_TO_[TYPE] 変換ファンクション

名前	説明
PBOOL_TO_BOOL [▶ 180]	BOOLポインタ変数の内容を返します。
PBYTE_TO_BYTE [▶ 180]	BYTEポインタ変数の内容を返します。
PDATE_TO_DATE [▶ 181]	DATEポインタ変数の内容を返します。
PDINT_TO_DINT [▶ 181]	DINTポインタ変数の内容を返します。
PDT_TO_DT [▶ 181]	DTポインタ変数の内容を返します。
PDWORD_TO_DWORD [▶ 182]	DWORDポインタ変数の内容を返します。
PHUGE_TO_HUGE [▶ 182]	T_HUGE_INTEGERポインタ変数の内容を返します。
PINT_TO_INT [▶ 183]	INTポインタ変数の内容を返します。
PLARGE_TO_LARGE [▶ 183]	T_LARGE_INTEGERポインタ変数の内容を返します。
PLINT_TO_LINT [▶ 183]	LINTポインタ変数の内容を返します。
PLREAL_TO_LREAL [▶ 184]	LREALポインタ変数の内容を返します。
PLWORD_TO_LWORD [▶ 184]	LWORDポインタ変数の内容を返します。
PMaxSTRING_TO_MaxSTRING [▶ 184]	T_MaxStringポインタ変数の内容を返します。
PREAL_TO_REAL [▶ 185]	REALポインタ変数の内容を返します。

名前	説明
PSINT_TO_SINT [▶ 185]	SINTポインタ変数の内容を返します。
PSTRING_TO_STRING [▶ 185]	STRINGポインタ変数の内容を返します。
PTIME_TO_TIME [▶ 186]	TIMEポインタ変数の内容を返します。
PTOD_TO_TOD [▶ 186]	TODポインタ変数の内容を返します。
PUDINT_TO_UDINT [▶ 187]	UDINTポインタ変数の内容を返します。
PUHUGE_TO_UHUGE [▶ 187]	T_UHUGE_INTEGERポインタ変数の内容を返します。
PUINT_TO_UINT [▶ 187]	UINTポインタ変数の内容を返します。
PULARGE_TO_ULARGE [▶ 188]	T_ULARGE_INTEGERポインタ変数の内容を返します。
PULINT_TO_ULINT [▶ 188]	ULINTポインタ変数の内容を返します。
PUSINT_TO_USINT [▶ 188]	USINTポインタ変数の内容を返します。
PWORD_TO_WORD [▶ 189]	WORDポインタ変数の内容を返します。
PUINT64_TO_UINT64 [▶ 189]	T_ULARGE_INTEGERポインタ変数の内容を返します。

T_Argヘルプファンクション

名前	説明
F_ARGCMP [▶ 215]	T_Arg型の2つの変数を比較します。
F_ARGCPY [▶ 216]	T_Arg型の変数の値を他の変数にコピーし、正常にコピーされたデータバイト数を返します。
F_ARGISZERO [▶ 217]	T_Argメンバ変数の1つが値ゼロ、または初期化されていない場合にTRUEを返します。
F_BIGTYPE [▶ 217]	T_Arg型の構造体の構造体または配列変数の情報を返します。
F_BOOL [▶ 217]	T_Arg型の構造体のBOOL型変数についての情報を返します。
F_BYTE [▶ 218]	T_Arg型の構造体のBYTE型変数についての情報を返します。
F_DINT [▶ 218]	T_Arg型の構造体のDINT型変数についての情報を返します。
F_DWORD [▶ 219]	T_Arg型の構造体のDWORD型変数についての情報を返します。
F_HUGE [▶ 219]	T_Arg型の構造体のT_HUGE_INTEGER型変数についての情報を返します。
F_INT [▶ 219]	T_Arg型の構造体のINT型変数についての情報を返します。
F_LARGE [▶ 220]	T_Arg型の構造体のT_LARGE_INTEGER型変数についての情報を返します。
F_LINT [▶ 220]	T_Arg型の構造体のLINT型変数についての情報を返します。
F_LREAL [▶ 221]	T_Arg型の構造体のLREAL型変数についての情報を返します。
F_LWORD [▶ 221]	T_Arg型の構造体のLWORD型変数についての情報を返します。
F_REAL [▶ 221]	T_Arg型の構造体のREAL型変数についての情報を返します。
F_SINT [▶ 222]	T_Arg型の構造体のSINT型変数についての情報を返します。
F_STRING [▶ 222]	T_Arg型の構造体のT_MaxString型変数についての情報を返します。
F_UDINT [▶ 223]	T_Arg型の構造体のUDINT型変数についての情報を返します。
F_UHUGE [▶ 223]	T_Arg型の構造体のT_UHUGE_INTEGER型変数についての情報を返します。
F_UINT [▶ 224]	T_Arg型の構造体のUINT型変数についての情報を返します。
F_ULARGE [▶ 224]	T_Arg型の構造体のT_ULARGE_INTEGER型変数についての情報を返します。
F_ULINT [▶ 224]	T_Arg型の構造体のULINT型変数についての情報を返します。
F_USINT [▶ 225]	T_Arg型の構造体のUSINT型変数についての情報を返します。
F_WORD [▶ 225]	T_Arg型の構造体のWORD型変数についての情報を返します。
F_PVOID [▶ 225]	T_Arg型の構造体のPVOID型変数についての情報を返します。

CSV形式ファンクション

名前	説明
CSVFIELD_TO_STRING [▶ 235]	CSV形式 のデータフィールドの文字列の値をPLCの文字列変数に変換します。
STRING_TO_CSVFIELD [▶ 270]	PLCの文字列変数の値をCSV形式 のデータフィールドをもつ文字列に変換します。

名前	説明
GSVFIELD_TO_ARG [▶ 234]	CSV形式 のデータフィールドをもつバイトバッファをランダムな PLC変数の値に変換します。
ARG_TO_GSVFIELD [▶ 229]	ランダムなPLC変数の値をCSV形式 のデータフィールドをもつバイトバッファに変換します。
FB_CSVMemBufferReader [▶ 40]	バイトバッファ内にあるCSV形式のデータセットを個々のデータフィールドに分割します。
FB_CSVMemBufferWriter [▶ 41]	個々のデータフィールドからバイトバッファ内に個々のまたは複数のデータセットを生成します。

ライセンス機能

名前	説明
FB_LicFileGetStorageInfo [▶ 77]	ライセンスドングルおよびファイルディレクトリのStorageInfoを読み込みます。
FB_LicFileCreate [▶ 75]	ライセンスドングルにファイルを作成します。
FB_LicFileDelete [▶ 76]	ライセンスドングルからファイルを削除します。
FB_LicFileRead [▶ 78]	ライセンスドングルからバッファへファイルを読み込みます。
FB_LicFileCopyToDongle [▶ 74]	ハードディスクからライセンスドングルへファイルをコピーします。
FB_LicFileCopyFromDongle [▶ 72]	ライセンスドングルからハードディスクへファイルをコピーします。
FB_CheckLicense [▶ 39]	ライセンスIDのTwinCAT 3 ライセンスステータスを判定します。
FB_GetDongleSystemId [▶ 60]	TwinCAT 3 ライセンスドングルのシステムIDとボリュームIDをGUID値で読み込みます。
FB_GetLicenseDongle [▶ 63]	接続されたライセンスドングルの数を判定し、アドレスとステータスを返します。
FB_GetLicenses [▶ 64]	有効および無効なTwinCATライセンスを読み込みます。
FB_GetLicensesEx [▶ 65]	すべてのTwinCAT 3 ライセンスとOEMライセンスのステータスを判定します。

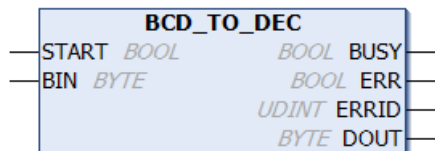
他のファンクション

名前	説明
FB_BasicPID [▶ 38]	シンプルPIDコントローラ
F_GetVersionToUtilities [▶ 227]	ライブラリのバージョン情報の読み込み。
IsFinite [▶ 226]	IEEE準拠の浮動小数点数のフォーマットを検証します。
F_YearIsLeapYear [▶ 146]	ある年がうるう年であるかどうかを判定します。
F_GetMaxMonthDays [▶ 142]	ある月の最大日数を判定します。
F_GetDOYOfYearMonthDay [▶ 141]	その年の日数を判定します。
F_GetMonthOfDOY [▶ 143]	その年の通算の日数に基づいて月を判定します。
F_GetDayOfWeek [▶ 141]	曜日を判定します。
F_GetWeekOfTheYear [▶ 144]	週番号を判定します。
F_GetDayOfMonthEx [▶ 140]	指定した月または年の週の最初の日、2番目の日などの判定をします。
F_GetWeekOfTheYear [▶ 144]	事前定義された日の週番号を返します。
RTC [▶ 129]	「ソフトウェア」RTC (リアルタイムクロック)
RTC_EX [▶ 130]	「ソフトウェア」RTC (リアルタイムクロック)
RTC_EX2 [▶ 131]	「ソフトウェア」RTC (リアルタイムクロック)
FB_FileRingBuffer [▶ 50]	ファイルへ、またはファイルからデータセットを読み込み/書き込みます。(FIFO)
FB_MemRingBuffer [▶ 87]	バッファ変数へ、またはバッファ変数からのデータセットを読み込み/書き込みます。(FIFO)
FB_MemRingBufferEx [▶ 88]	バッファ変数へ、またはバッファ変数からのデータセットを読み込み/書き込みます。(FIFO)
FB_StringRingBuffer [▶ 103]	バッファ変数へ、またはバッファ変数からの設定を読み込み/書き込みます。(FIFO)
FB_MemStackBuffer [▶ 90]	バッファ変数へのデータセットの書き込み/バッファ変数からデータセットの読み込みをします(LIFO)。

名前	説明
FB_MemBufferMerge [▶ 84]	個々の小さなデータセグメントを1つの大きなデータセグメントにマージします。
FB_MemBufferSplit [▶ 86]	メモリ領域(データバッファ)をいくつかの小さなセグメントに分割します。
FB_HashTableCtrl [▶ 71]、 F_CreateHashTableHnd [▶ 245]	シンプルなハッシュテーブル。
FB_LinkedListCtrl [▶ 79]、 F_CreateLinkedListHnd [▶ 246]	シンプルなリンクされたリスト(ダブルリンク)。
DCF77_TIME [▶ 26]	シンプルなDCF77デコーダ。
DCF77_TIME_EX [▶ 29]	2つの連続的なテレグラムとタイムゾーン情報の妥当性チェック付きDCF77デコーダ。

3 ファンクションブロック

3.1 BCD_TO_DEC



「BCD_TO_DEC」ファンクションブロックは、BCD数を10進数形式に変換します。変換するBCD数は、値の信頼性をチェックします。

VAR_INPUT

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

START: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

BIN: 変換するBCD数。

VAR_OUTPUT

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

BUSY: 変換プロセスの開始時にこの入力セットされると、変換が完了するまでセットされたままの状態になります。BUSY出力がリセットされると、10進数はDOUT出力で使用可能です。

ERR: エラーが発生した場合、この変数はTRUEにセットされます。

ERRID: エラーコード。

DOUT: プロセスが正常に完了した場合、10進数形式で変換された変数はこの出力で使用可能です。

エラーコード:

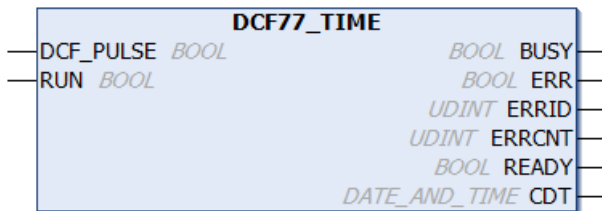
エラーコード	エラー説明
0	エラーなし
0x000F	BCD数の下位4ビットの信頼できない値。
0x00F0	BCD数の上位4ビットの信頼できない値。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.2 DCF77_TIME

i このファンクションブロックは、DCF77_TIME_EX [▶ 29] ファンクションブロックによってとって変わられています。



「DCF77_TIME」ファンクションブロックを使用して、DCF-77無線クロック信号をデコードできます。RUN入力の立ち上がりでデコード処理を開始し、RUN入力セットされている限り続きます。最悪のケースでは、ファンクションブロックは自身を同期させるために最大1分を必要とし、さらにデータをデコードするためにもう1分必要とします。その時間の間に、欠落している59秒目の秒マーカを待機します。内部的に、ファンクションブロックはDCF-77信号をサンプリングします。エラーなしでエッジをサンプリングするために、ファンクションブロックは各PLCサイクルで一度コールする必要があります。十分な結果は、サイクルタイムが25 ms以下で取得できます。DCF-77信号がないか、欠陥がある場合、ERR出力はTRUEにセットされ、対応するエラーコードはERRID出力にセットします。ERR出力とERRID出力は、次に正しい信号を受信したときにリセットされます。レシーバの中には、反転DCF-77信号を提供するものもあります。このような場合、DCF_PULSE入力に渡される前に、まず信号を反転する必要があります。エラーなしで稼働している場合、現在時間はCDT出力で毎分更新されます。この場合、最初の秒(ゼロ秒)で1PLCサイクルに対してREADY出力はTRUEにセットされます。この時、CDT出力のDCF-77時間は有効で、PLCプログラムで評価できます。次の分でデータにエラーが検出されない場合のみ、READY出力をセットします。送信されたパリティビットは、エラー検出に使用されます。受信状況が不良で、100%のエラーフリー識別を保証できないとき、2つの欠陥(反転)ビットがある場合、ファンクションブロックはエラーを検出できず、READY出力をTRUEにセットします。信頼できる時間情報を取得するためには、数分連続で時間情報を冗長性解析するなどの追加の安全防御を実装しなければいけません。

2つの連続テレグラムの単純な妥当性チェックが、DCF77_TIME ファンクションブロックに実装されています。このファンクションは、グローバルなブール変数を介してDCF77_TIMEブロックのすべてのインスタンスに対して有効にできます。妥当性チェックが有効な場合、最初の同期は最大3分までさらに延長されます。

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

受信中に発生するエラーは、ファンクションブロックによって登録されます。ERRCNT出力はエラーカウンタです。このカウンタは、最後に正しく受信した信号以降に発生したエラー数を示します。カウンタは、次に正しい信号を受信したときにリセットされます。

タイムコード

1分毎に、年、月、日、曜日、時間、分をエンコードしている数値が、秒符号のパルス変調によりBCDフォーマットで送信されます。送信された情報は、常に、後続の分を記述します。秒マーカは、秒ごとに送信されます。持続時間0.1の秒マーカはバイナリゼロを表し、一方持続時間0.2の秒マーカはバイナリ1を表します。情報は3つのチェックビットで拡張されます。秒マーカは59秒目には送信されず、レシーバは自己を同期させるためにこの「ギャップ」を使用できます。

ショートおよびロングパルス信号長は、グローバル変数で設定できます。信号不良の場合、パルス幅は小さくなります。レシーバの仕様には、通常、2つの論理信号用に最小、最大パルスについての情報があります。干渉の際には、値が大きいほどフィールド強度が大きく、値が小さいほどフィールド強度が小さくなります。また、論理ゼロのパルス幅が過大な場合、問題が送信者(フィールド強度が非常に大きい場所)の近くで発生する可能性があります。このため、レシーバの仕様に応じて、決められた制限がゼロと1の間を区別するためにセットされます。使用するレシーバの仕様をチェックし、適切なインパルス長を設定してください。

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

例: Atmel T4227の仕様(タイムコードの受信)は、以下の仕様のパルス長が含まれています。
 100 msパルス(ゼロ): 最小: 70 ms、標準: 95 ms、最大: 130 ms
 200 msパルス(1): 最小: 170 ms、標準: 195 ms、最大: 235 ms
 このICの場合、150 msのリミット値が理想的です(130 + ((170 ms - 130 ms) / 2))。

ヒント:

インパルス長の設定されたリミット値が小さすぎる場合、ショートインパルスがロングインパルスとして検出されてしまいます。逆に、設定されたリミット値が大きすぎる場合、ロングインパルスはショートインパルスとして検出されてしまいます。チェックサムが正しい場合、レシーバはこれらのエラーを検出できません。最初のケースの場合、レシーバは未来の範囲の時間を提供でき、2番目のケースでは過去の時間を提供している可能性があります。

VAR_INPUT

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

DCF_PULSE: DCF-77信号。

RUN: この入力での立ち上がりでファンクションブロックを初期化し、DCF-77信号のデコードを開始します。この入力のリセットされると、デコード処理は停止します。

VAR_OUTPUT

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

BUSY: この出力はファンクションブロックの実行中にセットされます。

ERR: この出力はデコード中にエラーが発生するとセットされます。

ERRID: ERR出力がセットされるとエラー番号を返します。

ERRCNT: 最後に正しく受信した信号以降に発生したエラー数。

READY: この出力をセットすると、CDT出力のデータが有効になります。

CDT: DATE_AND_TIME形式のDCF-77時間。

エラーコード	エラー説明
0	エラーはありません
0x100	タイムアウトエラー。DCF-77信号が検出されない可能性があるため。
0x200	パリティエラー。受信データに不正なビットが検出されました。
0x300	不正データを受信。パリティチェックは不正ビットを1つしか検出できないので、受信データの妥当性を再度チェックします(たとえば、month = 13の場合、このエラーコードが生成されます)。
0x400	最後のデコードサイクルが長すぎます。このエラーは、受信状態が不良の場合に発生します(十分な秒マーカーが受信されなかった)。
0x500	最後のデコードサイクルが短すぎます。このエラーは、受信状態が不良の場合に発生します(追加のエッジを受信した)。

例:

エラーのないデータを受信した場合、TwinCATソフトウェアクロック (RTC) はサンプルアプリケーションの無線報時と同期します。

```
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
```

オンラインビュー:

The screenshot displays the TwinCAT interface with the 'Source Control Explorer' showing the project 'TwinCAT_Device.Untitled1.P_DCF77_TIME'. Below this, a table lists the current values of the variables defined in the program:

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			

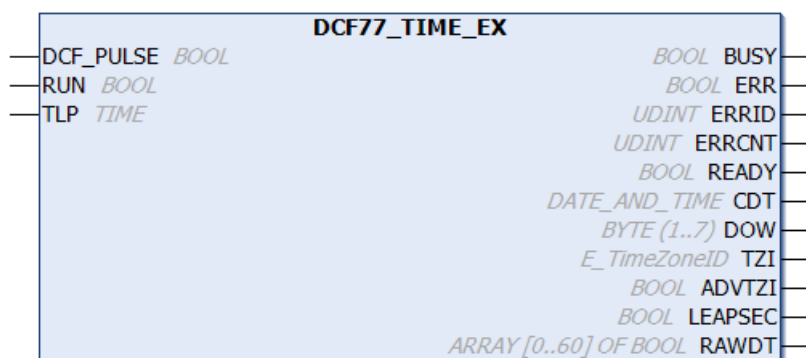
Below the table, a ladder logic diagram is shown. It features two function blocks: 'fbDcf' (DCF77_TIME) and 'fbRtc' (RTC). The 'fbDcf' block is triggered by 'bDcfPulse' (TRUE) through a 'NOT' gate. Its outputs include 'bBusy' (TRUE), 'bError' (FALSE), 'nErrID' (16#00000000), 'nErrCnt' (16#00000000), 'bDcfValid' (FALSE), and 'tDcfDT' (DT#2014-9-26-14:3:0). The 'fbRtc' block is triggered by 'bDcfValid' (FALSE) and 'tDcfDT' (DT#2014-9-26-14:3:0). Its outputs include 'bRtcValid' (TRUE) and 'tRtcDT' (DT#2014-9-26-14:3:11). The diagram is labeled '1' and '2' on the left side, and 'RET' is at the bottom.

DCF77 TIME EX [▶ 29]ファンクションブロックの説明も参照してください。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.3 DCF77_TIME_EX



「DCF77_TIME_EX」ファンクションブロックを使用して、DCF-77無線クロック信号をデコードできます。DCF77_TIME [▶ 26] ファンクションブロックとは対照的に、このブロックは標準的に妥当性のために2つの連続したテレグラムをチェックします。

RUN入力の立ち上がりでデコード処理を開始し、RUN入力セットされている限り続きます。ファンクションブロックの同期には最長1分かかり、データのデコードのためにさらに2分かかります。その間に、欠落している59秒目の秒マーカを待機します。内部的に、ファンクションブロックはDCF-77信号をサンプリングします。エラーなしでエッジをサンプリングするために、ファンクションブロックは各PLCサイクルで一度コールする必要があります。サイクルタイムが25 ms以下の場合、十分な結果が取得できます。DCF-77信号がないか、欠陥がある場合、ERR出力はTRUEにセットされ、対応するエラーコードはERRID出力にセットされます。ERR出力とERRID出力は、次に正しい信号を受信したときにリセットされます。レシーバの中には、反転DCF-77信号を返すものもあります。このような場合、DCF_PULSE入力に渡される前に、まず信号を反転する必要があります。エラーなしで稼働している場合、現在時間はCDT出力で毎分更新されます。この場合、1PLCサイクルの最初の秒(ゼロ秒)で、READY出力がTRUEにセットされます。この時、CDT出力のDCF-77時間は有効で、PLCプログラムで評価できます。エラーなしで次の分データを受信できる場合のみ、READY出力をセットします。送信されたパリティビットはフォルト検出のために使用され、最後の2つのテレグラムの妥当性がチェックされます。受信状況が不良で100%エラーがない識別が保証できない場合で、2つの後続のテレグラムに2つの欠陥(反転)ビットがある場合、ファンクションブロックはエラーを検出できず、READY出力をTRUEにセットします。妥当性チェックにより、適切なビットが歪む確率が極めて少なくなり、このようなエラーの検出が阻止されます。

受信中に発生するエラーは、ファンクションブロックによって登録されます。ERRCNT出力はエラーカウンタです。このカウンタは、最後に正しく受信した信号以降に発生したエラー数を示します。カウンタは、次に正しい信号を受信したときにリセットされます。

タイムコード

1分毎に、年、月、日、曜日、時間、分をエンコードしている数値が、秒符号のパルス変調によりBCDフォーマットで送信されます。送信された情報は、常に、後続の分を記述します。秒マーカは、秒ごとに送信されます。持続時間0.1の秒マーカはバイナリゼロを表し、一方持続時間0.2の秒マーカはバイナリ1を表します。情報は3つのチェックビットで拡張されます。秒マーカは59秒目には送信されず、レシーバは自己を同期させるためにこの「ギャップ」を使用できます。

VAR_INPUT

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

DCF_PULSE: DCF-77信号。

RUN: この入力の立ち上がりでファンクションブロックを初期化し、DCF-77信号のデコードを開始します。この入力のリセットされると、デコード処理は停止します。

TLP: この入力によって、受信側の仕様に依じて、ゼロと1の間を区別するために一定の制限をセットします。信号不良の場合、パルス幅は小さくなります。レシーバの仕様には、通常、2つの論理信号用に最小、最大パルスについての情報があり、干渉の際には、値が大きいほどフィールド強度が大きく、値が小さいほどフィールド強度が小さくなります。また、論理ゼロのパルス幅が過大な場合、問題が送信者(フィールド強度が非常に大きい場所)の近くで発生する可能性があります。**使用するレシーバの仕様をチェックし、適切なインパルス長を設定してください。**

例: Atmel T4227の仕様(タイムコードの受信)は、以下の仕様のパルス長が含まれています。

100 msパルス(ゼロ): 最小: 70 ms、標準: 95 ms、最大: **130 ms**

200 msパルス(1): 最小: **170 ms**、標準: 195 ms、最大: 235 ms

このICの場合、150 msのリミット値が最適です(130 + ((170 ms - 130 ms) / 2))。

ヒント:

インパルス長の設定されたリミット値が小さすぎる場合、ショートインパルスがロングインパルスとして検出されてしまいます。逆に、設定されたリミット値が大きすぎる場合、ロングインパルスはショートインパルスとして検出されてしまいます。チェックサムが正しい場合、レシーバはこれらのエラーを検出できません。最初のケースの場合、レシーバは未来の範囲の時間を提供でき、2番目のケースでは過去の時間を提供している可能性があります。

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: この出力はファンクションブロックの実行中にセットされます。

ERR: この出力はデコード中にエラーが発生するとセットされます。

ERRID: ERR出力がセットされるとエラー番号を返します。

ERRCNT: 最後に正しく受信した信号以降に発生したエラー数。

READY: この出力をセットすると、CDT出力のデータが有効になります。

CDT: DATE_AND_TIME形式のDCF-77時間。

DOW: ISO 8601に準拠した曜日: 1 = 月曜日~7 = 日曜日。

TZI: タイムゾーン情報(夏時間/冬時間)。

ADVTZI: 夏時間/冬時間の切り替え通知。例: CET→CESTまたはCEST→CET。CEST/CETの間の切り替えは、この時間の最後に発生します(テレグラム例を参照)。

LEAPSEC: うるう秒の通知。うるう秒はこの時間の最後に追加されます(テレグラム例を参照)。

RAWDT: 最後にデコードされた(未加工の)ビット情報。時間情報のパリティビットのみがチェックされるようにしてください。気象データのパリティビットは解析されません。

エラーコード	エラー説明
0	エラーはありません
0x100	タイムアウトエラー。DCF-77信号が検出されない可能性があるため。

エラーコード	エラー説明
0x200	パリティエラー。受信データに不正なビットが検出されました。
0x300	不正なデータが受信されました。パリティチェックは不正ビットを1つしか検出できないので、受信データの妥当性を再度チェックします(たとえば、month = 13の場合、このエラーコードが生成されます)。
0x400	最後のデコードサイクルが長すぎます。このエラーは、受信状態が不良の場合に発生します(十分な秒マーカーが受信されなかった)。
0x500	最後のデコードサイクルが短すぎます。このエラーは、受信状態が不良の場合に発生します(追加のエッジを受信した)。

テレグラム例:

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

```
'0 01110100100111 0 11001 00011011 0100001 011001 111 00001 000100000': Sunday, 26.10.08 02:58:00, TZI = eTimeZoneID_Daylight, ADVTZI = TRUE
'0 11110001101110 0 11001 10011010 0100001 011001 111 00001 000100000': Sunday, 26.10.08 02:59:00, TZI = eTimeZoneID_Daylight, ADVTZI = TRUE
'0 01000001001110 0 10101 00000000 0100001 011001 111 00001 000100000': Sunday, 26.10.08 02:00:00, TZI = eTimeZoneID_Standard, ADVTZI = TRUE
'0 01111110100000 0 00101 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 0 10101 00011011 1000001 000011 111 11000 000100000': Sunday, 30.03.08 01:58:00, TZI = eTimeZoneID_Standard, ADVTZI = TRUE
'0 01000010100111 0 10101 10011010 1000001 000011 111 11000 000100000': Sunday, 30.03.08 01:59:00, TZI = eTimeZoneID_Standard, ADVTZI = TRUE
'0 10000111100011 0 11001 00000000 1100000 000011 111 11000 000100000': Sunday, 30.03.08 03:00:00, TZI = eTimeZoneID_Daylight, ADVTZI = TRUE
'0 01010000010110 0 01001 10000001 1100000 000011 111 11000 000100000': Sunday, 30.03.08 03:01:00, TZI = eTimeZoneID_Daylight, ADVTZI = FALSE
```

Leap second

```
'0 10110000100001 0 00111 10011010 0000000 100000 001 10000 100100001': Thursday, 01.01.09 00:59:00, TZI = eTimeZoneID_Standard, LEAPSEC = TRUE
'0 11010010111000 0 00111 00000000 1000001 100000 001 10000 1001000010': Thursday, 01.01.09 01:00:00, TZI = eTimeZoneID_Standard, LEAPSEC = TRUE
'0 01000110011101 0 00101 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,

例:

エラーのないデータを受信した場合、TwinCATソフトウェアクロック (RTC) はサンプルアプリケーションの無線報時と同期します。

```
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 );
```

オンラインビュー:

The screenshot shows the TwinCAT Source Control Explorer interface. At the top, there are tabs for 'MAIN [Online]', 'P_DCF77_TIME [Online]', and 'Source Control Explorer'. Below this, a table lists variables and their current values:

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			

Below the table, a snippet of ladder logic is shown with variable values substituted:

```
1 fbDcf( DCF_PULSE := NOT bDcfPulse, RUN := TRUE, TLF := T#140ms, T := T#140ms,
2 BUSY := bBusy, ERR := bError, ERRID := 16#00000000, ERRCNT := 16#00000000,
3 READY := bDcfValid, CDT := DT#2014-9-26-14:4:0, DT := tDcfDt, DOW := 16#05, TZ := eTzi,
4 ADVTZI := bAdvTzi, LEAPSEC := bLeapSec, RAWDT := arRawDt );
5
6 fbRtc( EN := bDcfValid, PDT := tDcfDt, Q := bRtcValid, CDT := DT#2014-9-26-14:4:4,
7 tRtcDt := tRtcDt );
```

DCF77 TIME [▶ 26] ファンクションブロックの説明を参照してください。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.4 DEC_TO_BCD



「DEC_TO_BCD」ファンクションブロックは、10進数をBCDフォーマットに変換します。変換する数値は、値の信頼性をチェックします。

VAR_INPUT

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

START: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

DIN: 変換を必要とする10進数。

VAR_OUTPUT

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


BUSY: 変換プロセスの開始時にこの入力がセットされると、変換が完了するまでセットされたままの状態になります。BUSY出力がリセットされると、BCD値がBOUT出力で使用できます。

ERR: エラーが発生した場合、この変数はTRUEにセットされます。

ERRID: エラーコード。

BOUT: プロセスが正常に完了した場合、BCDフォーマットで変換された変数はこの出力で使用できます。

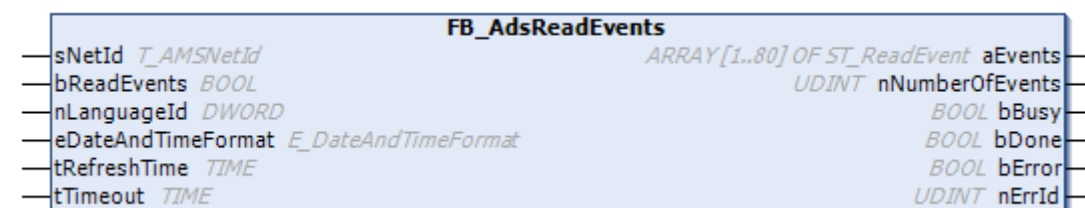
エラーコード:

エラーコード	エラー説明
0	エラーなし
0x00FF	変換する変数は、禁止された10進数の値です。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.5 FB_AdsReadEvents



ファンクションブロックはADSを介してアクティブなEventLoggerのメッセージを確認し、配列aEventsの形式で利用できるようにします。視覚化エレメントのメッセージを表示するためには、配列aEventsをプロパティメッセージデータ配列に入力する必要があります。

VAR_INPUT

```

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

EN_VAR

sNetId: EventLoggerのメッセージを確認するデバイスのAmsNetID。メッセージをローカルで読み込む場合、空の文字列を指定できます。

bReadEvents: この入力で、メッセージの読み込みを有効にします。有効がリセットされると、エラー出力(bErrorおよびnErrId)もリセットされます。

nLanguageId: (言語ID)、メッセージテキストをどの言語で表示するかを定義します。

eDateAndTimeFormat: タイムスタンプ形式を定義します。以下のオプションが使用できます。

- de_De - ドイツ語式: dd.MM.yyyy hh:mm:ss (24 h)
- en_GB - 英国式: dd/MM/yyyy hh:mm:ss (12 h)
- en_US - 米国式: MM/dd/yyyy hh:mm:ss (12 h)

tRefreshTime: メッセージ確認ダイアログが再送される時間間隔を定義します。

tTimeout: タイムアウトエラーがトリガされる時間間隔を定義します。

VAR_OUTPUT

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

aEvents: ファンクションブロックが、読み込みメッセージメッセージを使用できるようにするためにこの配列を使用します。配列は最大80メッセージを保存します。

nNumberOfEvents: 配列aEventsに現在、保存されているメッセージ件数を示します。

bBusy: ファンクションブロックが演算を実行しているかどうかを示します。

bDone: ファンクションブロックが現在、ビジーでなく、演算を最低1回実行した場合は、TRUE。

bError: エラーが発生したかどうかを示します。

nErrorId: エラー番号を指定します。

3.6 FB_AddRouteEntry



ファンクションブロックを使用して、新しいAMSルータ接続(リモートルート)をTwinCATシステムに追加できます。

● AMSルータ接続リスト

I AMSルータ接続の両方の通信パートナーが、AMSルータ接続リストを保有します。これらのリストは、AMSルータ接続を含みます。両方の通信パートナーがその接続リストに互いに入力された場合、AMSルータの接続が機能します。ファンクションブロックを使用している場合、1つの通信パートナーのリストのみが拡張されません。

VAR_INPUT

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

sNetID: これで、新しい接続がAMSルータ接続リスト(T_AmsNetID型)に追加されるTwinCATコンピュータのネットワークアドレスで文字列を指定できます。ローカルコンピュータで実行する場合、空の文字列を入力できます。

stRoute: 新しい接続(ST_AmsRouteEntry [▶_291]型)のためのパラメータ付き構造体エレメント。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットします。

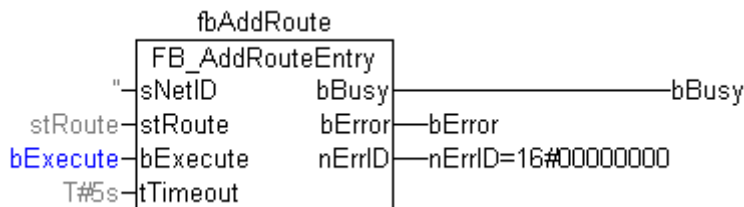
nErrId: *bError*出力がセットされると、ADSエラー番号 [▶ 342]を返します。

例:

ローカルTwinCATシステムでは、新しいAMSルータ接続が接続名付きで追加されます: 「TEST」、TwinCATネットワークアドレス: 「172.16.6.111.1.1」、IPアドレス: 「172.16.6.111」および伝達経路: 「TCP/IP」。

```
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
```

必要な接続パラメータは、宣言部分で既に初期化されています。立ち上がりが**bExecute**変数で検出された場合、新しい接続が追加されます。



要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.7 FB_AmsLogger



「TwinCAT AMS Logger」は、「TwinCAT ADS Monitor」(.¥TwinCAT¥AdsMonitor¥Logger¥TcAmsLog.exe)のコンポーネントです。ロガーは、データストレージデバイスのAMS/ADSコマンドを記録します。記録は後で表示でき、トラブルシューティングのために[TwinCAT AMS ADS Viewer]で分析できます。FB_AmsLoggerファンクションブロックを使用して、PLCプログラムからの記録を開始、または停止できます。FB_AmsLoggerファンクションブロックは、TcAmsLog.exeの既存のインスタンス/実行中のインスタンスとのみ通信できます。つまり、TcAmsLog.exeは、スタートメニューまたはNT_StartProcess [▶ 119]ブロックを介して手動で既に開始していなければいけません。

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: これで、TwinCATコンピュータのAmsNetIDを「TwinCAT AMS Logger」のステータスに指定できます(T_AmsNetID)。空の文字列をローカルコンピュータのロガー用に指定できます。

eMode: 「TwinCAT AMS Logger」がセットされる新しいステータス(E_AmsLoggerMode [▶ 281]、記録の開始/停止)。

sCfgFilePath: (オプション)「TwinCAT AMS Logger」コンフィグレーションファイルのパス(T_MaxString)。現在のところまだ実装されていませんが、将来の用途のために予約済みです(空の文字列を入力してください)。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後でこの出力をセットします。

nErrId: bError出力がセットされると、ADSエラー番号 [▶ 342]を返します。

例:

PLCプログラムが起動すると、TcAmsLog.exeのインスタンスがローカルシステムで起動します。bRecord変数がTRUEにセットされると、AMS/ADSコマンドの記録が開始します。変数がFALSEにリセットされると停止します。

宣言部分:

```
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

```

実装:

```

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

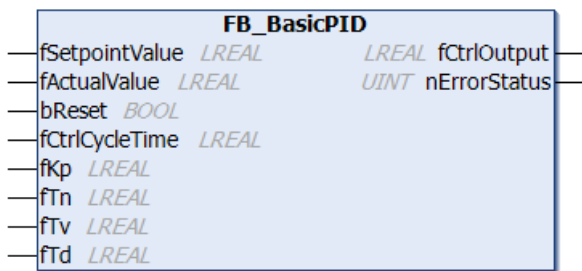
100: (* Error state *)
  ;
END_CASE

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.8 FB_BasicPID

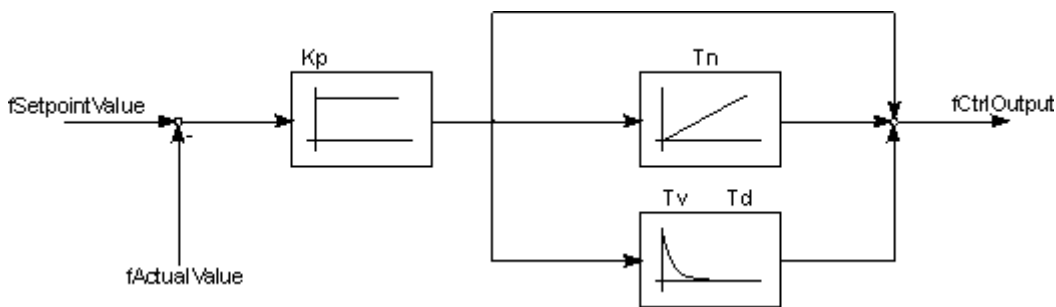


ファンクションブロックは、簡単な離散化したPIDエレメントです。

送信機能:

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

アクション図:



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 : 制御変数の設定値。

fActualValue : 制御変数の現在値。

bReset: この入力がTRUEの場合、内部状態変数とコントローラ出力をリセットします。

fCtrlCycleTime: ファンクションブロックをコールし、制御ループを処理するサイクルタイム[s]。

各PLCサイクルでブロックをコールする場合、ここで各PLCタスクのサイクルタイムを指定しなければいけません。

あるいは、PLCタスクのサイクルタイムを適切な倍数にしなければいけません。

fKp : コントローラの増幅/コントローラ係数

fTn : 積分動作時間[s]

fTv : 微分時間[s]

fTd : 減衰時間[s]

VAR_OUTPUT

```
VAR_OUTPUT
    fCtrlOutput : LREAL;
    nErrorStatus : UINT
END_VAR
```

fCtrlOutput : PIDエレメントの出力。

nErrorStatus: エラーの場合のエラー番号を示します (nErrorStatus <> 0)。

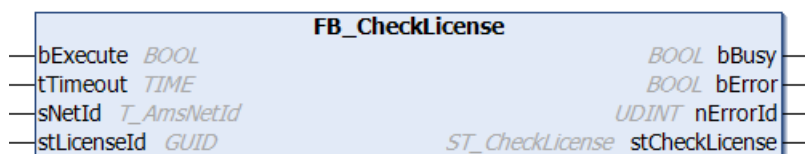
エラーコード:

値	定数	エラー説明
0	nERR_NOERROR	エラーはありません
1	nERR_INVALIDPARAM	無効なパラメータ
2	nERR_INVALIDCYCLETIME	無効なサイクルタイム。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PGまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.9 FB_CheckLicense



ファンクションブロックFB_CheckLicenseは、ライセンスIDに紐づいたTwinCAT 3 ライセンスのステータスを判定します。

VAR_INPUT

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

bExecute: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: コマンドが実行されるときに、超過してはならないタイムアウト時間。

sNetId: そのライセンスステータスが読み込まれるTwinCATコンピュータのAmsNetId (AMSネットワーク識別子) (T_AmsNetId)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

stLicenseId: ライセンスID (GUID)

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy : BOOL;
    bError : BOOL;
```

```
nErrorId      : UDINT;
stCheckLicense : ST_CheckLicense
END_VAR
```

bBusy: ファンクションブロック実行中はTRUEです。

bError: コマンド実行中にエラーが発生するとTRUEになります。

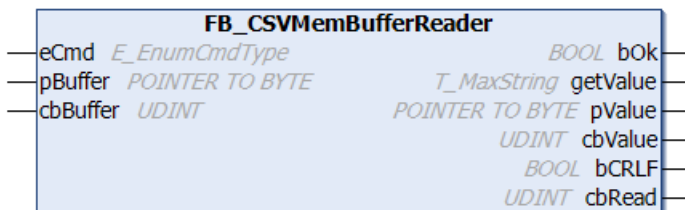
nErrorId: bError出力がセットされると、ADSエラー番号を返します。

stCheckLicense: ライセンスデータをもつ構造体(ST_CheckLicense)

要件

開発環境	ターゲットシステム	必要なPLCライブラリ
TwinCAT v3.1.4022	PCまたはCX (x64、x86)	Tc2_Utilities >= 3.3.24.0

3.10 FB_CSVMemBufferReader



このファンクションブロックを使用して、外部バッファに保存されたデータセットを個々のデータフィールドに分解したり/解釈したりできます。たとえば、最初にファイルアクセスのためにファンクションブロックを用いて、ファイルからバッファデータを読み込みできます。ファンクションブロックは最初のデータフィールド、または次のデータフィールドを読み込み、その値をgetValue出力で文字列として、またはpValue/cbValue出力でアドレス/バイト値としてのどちらかで返します。

バッファのデータは、ファンクションブロックがデータを正しく解釈できるようにするために特定の形式である必要があります。CRLF (CR=キャリッジリターン、LF=ラインフィード)は、データセット区切り文字として使用しなければいけません。データセットの最後はCRLFで終了しなければいけません。個々のデータフィールドは、データフィールド区切り文字で分離する必要があります。デフォルトのデータフィールド区切り文字はセミコロン(;)です。区切り文字は、グローバルPLC変数DEFAULT_CSV_FIELD_SEPを使用してセミコロン(;)からカンマ(,)に変更できます。

VAR_INPUT

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

eCmd: バッファコンポーネント (E_EnumCmdType [▶ 283]) の制御パラメータ。eEnumCmd_Firstは最初のデータフィールドを読み込み、eEnumCmd_Nextは次のデータフィールドを読み込みします。その他のパラメータ値は使用されません。

pBuffer: ソースバッファ変数用のアドレス(ポインタ)。アドレスは、ADR演算子で判定できます。このバッファは、読み込むデータセット/データフィールドデータが含まれています。

cbBuffer: ソースバッファ(データセット/データフィールドデータ)で解釈されるデータのバイトサイズ。バッファサイズは、解釈されるデータ量よりはるかに大きい可能性があります。解釈される実際のデータ長を入力してください。

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 = 成功、FALSE = 欠陥データ/欠陥のある入力パラメータ、またはデータの最後に到達してデータフィールドをこれ以上読み込むことができなかった。

getValue: 文字列としての最新読み込みデータデータフィールド(T_MaxString)。制御文字やバイナリデータの無いデータフィールドの場合、この出力はデータフィールド全体をゼロ終端文字列として返します。制御文字、またはバイナリデータをもつデータフィールドは、この出力で不完全な文字列が返されることになる可能性があります。この場合、出力pValue/cbValueを使用して、最新の読み込みデータフィールドにアクセスします。

pValue: データフィールドの最初のデータバイトへのアドレス(ポインタ)。空の無効のデータフィールドはゼロで終わらないため(PLC文字列では通常)、データがないことに注意してください。この場合、アドレスはゼロです。

cbValue: データフィールド長(バイト単位)。空の無効のデータフィールドはゼロで終わらないため(PLC文字列では通常)、データがないことに注意してください。この場合は、長さもゼロです。

bCRLF: 最後の読み込みコマンドコマンド の間にデータセットの最後まで達した場合、この出力をセットします。最後の読み込みデータフィールドは、前のデータセットに属します。次のデータフィールドは、新しいデータセットに属します。

cbRead: 正常に読み込み/解釈されたデータバイト数。この数字は、cbValue出力でデータフィールド長よりも大きいことがあります。cbRead出力での長さは、解釈されたデータフィールド/データセット区切り文字を含みます。

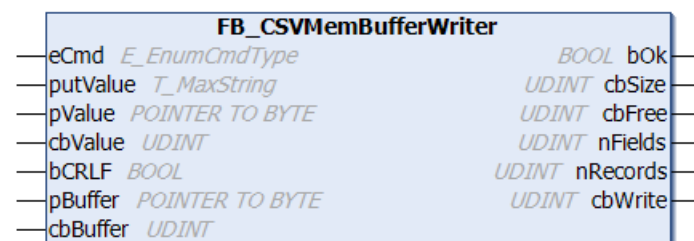
例:

参照: [例: CSV ファイルの書き込み/読み込み \[▶ 333\]](#)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.11 FB_CSVMemBufferWriter



このファンクションブロックを使用して、個々のデータフィールドからCSV形式で外部バッファにデータセットを生成できます。バッファの内容は、その後、ファイルアクセスのためにファンクションブロックを用いて、ファイルに書き込みできます。新しいデータフィールドは、制御文字(文字列)なしのデータフィールド、制御文字付きのデータフィールド、またはバイナリデータがデータセットに書き込まれるかどうかに応じて、putValue変数(文字列)、またはオプションの pValueおよびcbValue変数のいずれかによってブロックに送信できます。ファンクションブロックは、最大使用可能バッファサイズに到達するまでバッファに複数のデータセットを生成できます。データセットの終わり(現在のデータセットの最後のデータフィールド)は、データフィールドの書き込み中にbCRLF変数がTRUEにセットされると、自動的に

データフィールドに追加されます。ブロックは、自動的にデータフィールド区切り文字を追加します。デフォルトのデータフィールド区切り文字はセミコロン(;)です。区切り文字は、グローバルPLC変数 `DEFAULT_GSV_FIELD_SEP` を使用してセミコロン(;)からカンマ(,)に変更できます。

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: バッファコンポーネントの制御パラメータ (`E_EnumCmdType` [▶ 283])。 `eEnumCmd_First` は最初のデータフィールドをバッファに追加し、 `eEnumCmd_Next` は次のデータフィールドを追加します。その他のパラメータ値は使用されません。

putValue: 文字列としての新しいデータフィールド (`T_MaxString`)。この入力かわりに、オプションのパラメータ `pValue` と `cbValue` を使用する場合、この入力は空の文字列でなければいけません。

pValue: オプション: 新しいデータフィールドを含む外部バイトバッファのアドレス。 `cbValue` パラメータと共に、この入力を使用して、たとえば、制御文字またはバイナリデータ付きデータフィールドをデータセットに書き込みできます。データフィールドの制御文字またはバイナリデータは、不要な場所で `putValue` 文字列を切り捨てることができ、そのためバイトバッファとして送信されます。使用しない場合、この入力はゼロでなければいけません。

cbValue: オプション: 外部バイトバッファのデータフィールドのデータ長。使用しない場合、この入力はゼロでなければいけません。

bCRLF: この入力をセットすると、新しいデータフィールドはCRLFデータセット区切り文字で終わります。後続のデータフィールドは、新しいデータセットに属します。

pBuffer: ターゲットバッファ変数のアドレス(ポインタ)アドレスは、ADR演算子で判定できます。このバッファでは、ファンクションブロックはCSV形式でデータセットを生成します。

cbBuffer: ターゲットバッファ変数の最大使用可能サイズ(バイト単位)。サイズは、`SIZEOF`演算子で判定できます。

VAR_OUTPUT

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

bOk: TRUE = 成功、FALSE = バッファオーバーフロー、または欠陥のある入力パラメータ。

cbSize: 現在のバッファ入力ステータス(バッファで作成されたデータバイト数)。

cbFree: バッファの空のデータバイト数。

nFields: 書き込みデータデータフィールド数。

nRecords: 書き込みデータセット数。

cbWrite: 最後に書き込まれたデータされたデータバイト数(最後のデータフィールド長 + 任意のデータセットまたはデータフィールドの区切り文字)。

例:

参照: 例: CSV ファイルの書き込み/読み込み [▶ 333]。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.12 FB_EnumFindFileEntry

FB_EnumFindFileEntry	
sNetId T_AmsNetId	BOOL bBusy
sPathName T_MaxString	BOOL bError
eCmd E_EnumCmdType	UDINT nErrID
bExecute BOOL	BOOL bEOE
tTimeout TIME	ST_FindFileEntry stFindFile

このファンクションブロックは、指定された名前に類似した名前のファイルのディレクトリ、またはサブディレクトリを検索します。検出されたすべてのエントリは、個々に読み込みできます。FB_EnumFindFileListファンクションブロックの説明を参照してください。入力パラメータ *eCmd* を使用して、エントリリストをナビゲートします。*eCmd* 入力は、たとえば、最初の入力、または次の入力を読み込むかどうかを判定します。

重要な注記:

前の検索が完全に終了した場合のみ、新しい検索を開始できます。全数検索のためには、ファンクションブロックを複数回実行する必要があります (bExecute 入力の立ち上がりで)。検索は bEOE = TRUE に達した場合、または ECMD = eEnumCmd_Abort で途中で中止された場合にのみ完了します。

TwinCAT システムの場合、PLC アプリケーションが検索したファイルまたはディレクトリを既に検出している場合、検索が完了しないことがあります。

すべてのエントリを読み込んでいないしていない (すなわち、bEOE = TRUE に達していない) 場合、続いて入力パラメータ *eCmd* = eEnumCmd_Abort でファンクションブロックをコールしなければいけません。検索を完了するためにこの動作が必要で、すべての内部リソース (ファイルハンドラー) をリリースする必要があります。bEOE = TRUE に達したか、またはエラーが発生した場合、内部的に eEnumCmd_Abort が自動的に実行されません。

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: ディレクトリの検索が実行される TwinCAT コンピュータのネットワークアドレスを含む文字列 (T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

sPathName: 文字列として有効なディレクトリ名またはファイル名付きディレクトリ (T_MaxString)。文字列には、ワイルドカード (*および?) を使用できます。パスがワイルドカード、ドット、ディレクトリ名で終わる場合、ユーザはこのパスとサブディレクトリへのアクセス権を持っている必要があります。

eCmd: 列挙型ブロック用のコマンドパラメータ (E_EnumCmdType [▶ 283])。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADS コマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットしません。

nErrId: *bError*出力がセットされると、ADSエラー番号 [▶ 342]を返します。

bEOE: 列挙が完了しました。存在しないエントリを読み込む最初の試行中に、この出力がTRUEにセットされます。これは、読み込みエントリが**bEOE** = FALSEおよび**bError** = FALSEである限り有効であることを意味します。

stFindFile: 成功した場合、この構造体変数は検出されたファイルについての情報を返します (ST_FindFileEntry [▶ 296])。

例:

参照: 例: ファイル検索 (FB_EnumFindFileEntry, FB_EnumFindFileList)。 [▶ 315]

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.13 FB_EnumFindFileList



このファンクションブロックは、指定された名前に類似した名前のファイルのディレクトリ、またはサブディレクトリを検索します。検出されたすべてのエントリは、個々に読み取ることができます。FB_EnumFindFileEntryファンクションブロックの説明を参照してください。入力パラメータ *eCmd* は、エントリリストをナビゲートするために使用されます。*eCmd*入力、たとえば、最初の入力、または次の入力のどちらを読み取るかを決定します。

重要な通知:

前の検索が完全に終了した場合のみ、新しい検索を開始できます。全数検索のためには、ファンクションブロックを複数回実行する必要があります (*bExecute*入力の立ち上がりで)。検索は**bEOE** = TRUEに達した場合、または**ECMD** = *eEnumCmd_Abort*で途中で中止された場合にのみ完了します。

TwinCATシステムの場合、PLCアプリケーションが検索したファイルまたはディレクトリを既に検出している場合、検索が完了しないことがあります。

すべてのエントリを読み込んでいない(すなわち、*bEOE=TRUE*に達していない)場合、続いて入力パラメータ *eCmd = eEnumCmd_Abort* でファンクションブロックをコールしなければいけません。検索を完了するためにこの動作が必要で、すべての内部リソース(ファイルハンドラー)をリリースする必要があります。*bEOE=TRUE*に達したか、またはエラーが発生した場合、内部的に *eEnumCmd_Abort* が自動的に実行されます。

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: ディレクトリの検索が実行されるTwinCATコンピュータのネットワークアドレスの文字列(T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

sPathName: 有効なディレクトリ名またはファイル名付きディレクトリの文字列(T_MaxString)。文字列には、ワイルドカード(*および?)を使用できます。パスがワイルドカード、ドット、ディレクトリ名で終わる場合、ユーザはこのパスとそのサブディレクトリへのアクセス権を持っている必要があります。

eCmd: 列挙型ブロック用の制御コマンド(E_EnumCmdType [▶ 283])。

pFindList: ST_FindFileEntry [▶ 296]の配列変数のアドレス(ポインタ変数)。

cbFindList: ST_FindFileEntry [▶ 296]の配列変数のバイトサイズ。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットします。

nErrId: *bError*出力がセットされると、ADSエラー番号 [▶ 342]を返します。

bEOE: 列挙が完了しました。存在しないエントリを最初に読み込みしようとするときに、この出力がTRUEにセットされます。これは、読み込みエントリが*bEOE = FALSE*および*bError = FALSE*である限り有効であることを意味します。

nFindFiles: バッファの有効なエントリ数。

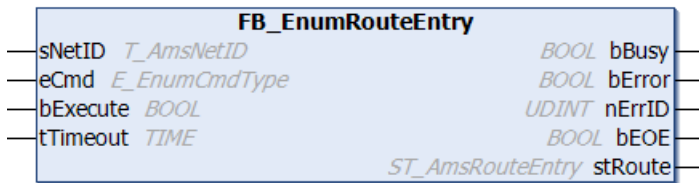
例:

参照: 例: ファイル検索(FB_EnumFindFileEntry、FB_EnumFindFileList) [▶ 315]。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.14 FB_EnumRouteEntry



このファンクションブロックを使用して、AMSルータ接続経由で他のTwinCATシステムへ情報を送信します（リモートルート）。いくつかの接続を使用する場合、ファンクションブロックを繰り返してコールする必要があります。コールごとに、1エントリのみを処理できます。入力パラメータ *eCmd* を使用して、エントリリストをナビゲートします。*eCmd* 入力、最初の入力、または次の入力を読み込むかどうかを判定します。

VAR_INPUT

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

sNetID: そのAMSルータ接続が読み込まれるされるTwinCATコンピュータのネットワークアドレスをもつ文字列を入力できます (T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

eCmd: 列挙型ブロック用の制御コマンド (E_EnumCmdType [▶ 283])。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットしません。

nErrId: *bError*出力がセットされると、ADSエラー番号 [▶ 342]を返します。

bEOE: 列挙の終わりに達しました。存在しないエントリを読み込む最初の試行中に、この出力がTRUEにセットされます。これは、読み込みエントリが *bEOE* = FALSE および *bError* = FALSE である限り有効であることを意味します。

stRoute: 最後の読み込み接続パラメータを格納する構造体エレメント (ST_AmsRouteEntry [▶ 291])。

例:

ローカルTwinCATシステム上の設定されたAMSルータ接続は読み込まれ、TwinCATシステムマネージャのロガー出力にメッセージとして書き込まれます。

```
PROGRAM P_EnumRouteEntries
VAR
  fbEnum : FB_EnumRouteEntry := ( sNetID := '', tTimeout := T#5s );
  bEnum  : BOOL := TRUE;
```

```

nState : BYTE := 0;
sInfo  : T_MaxString;
END_VAR

```

bEnum変数の立ち上がりで接続情報の読み込みをトリガしますをトリガします。

```

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

```

TwinCATシステムマネージャのロガー出力のログメッセージ:

Error List	
0 Errors	0 Warnings
2 Messages	Clear
Description	
1	19.09.2014 16:31:44 400 ms 'PlcTask' (350): ROUTE INFO: Name: CX-1088CE Address: CX-1088CE Transport: TCP/IP
2	19.09.2014 16:31:44 430 ms 'PlcTask' (350): ROUTE INFO: Name: TEST Address: 1.2.3.4 Transport: TCP/IP

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.15 FB_EnumStringNumbers

FB_EnumStringNumbers			
sSearch	T_MaxString	T_MaxString	sNumber
eCmd	E_EnumCmdType	INT	nPos
eType	E_NumGroupTypes	BOOL	bEOS

このファンクションブロックを使用して、REPEATまたはWHILEループの文字列の数値を検索します。文字列は、複数の数値を含んでいる可能性があります。検出される数値はすべて、ブロック出力で部分文字列として出力されます。ファンクションは、現在の位置から数値として解釈できる最初の文字を検索します。数値として解釈できない文字を検出した場合は、検索は中止されます。eCmdパラメータは、検索が最初の数値か次の数値かを判定します。eTypeパラメータは、検索文字列の数値書式を判定します。

VAR_INPUT

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

sSearch: 数値の検索対象の検索文字列(T_MaxString型)。

eCmd: 列挙型ブロックの制御コマンド(E_EnumCmdType [▶ 283]型)。

eType: 検索された数の数値書式(E_NumGroupTypes [▶ 285]型)。このパラメータは、どの文字を無視するか、どの文字を数値として解釈するか判定します。

値	意味
eNumGroup_Float	数「0」から「9」、「+」、「-」(符号)および「e」または「E」(指数文字)は有効な数値として解釈されます。
eNumGroup_Unsigned	数「0」から「9」は有効な数値として解釈されます。「+」、「-」、「e」、「E」文字は無視されます。
eNumGroup_Signed	数「0」から「9」、「+」、「-」(符号)は有効な数値として解釈されます。「e」、「E」文字は無視されます。

文字列が指数表現で数値を含む場合、eType = eNumGroup_Floatをセットしなければいけません(デフォルト)。

VAR_OUTPUT

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

sNumber: 文字列の最後に検出された数値(T_MaxString)。

nPos: この変数は、最後に検出されて正しくフォーマットされた数値の後の位置を返します。すなわち、ブロックは次にコールされるときに、この位置から新しい数値の検索を開始します。sSearch文字列の最後のゼロに達したとき、nPosはゼロです。文字列の最初の文字の位置番号は1です(非ゼロベースの位置)。

bEOS: 新しい数値が検出され、文字列の最後に達していない場合、この変数はFALSEです。この場合、sNumberは有効な数値を文字列として返します。さらに数値が検出されない場合、この変数はTRUEです。この場合、これ以上の検索は中止してください(sNumberは有効な値を返しません)。

例:

次の例では、sNumber変数は、有効な数値を検索します。検出されたすべての部分文字列は、配列変数 arrNumsに保存されます。


```

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

```

検出された文字列:

- ・ '-12e-34'
- ・ '-56'
- ・ '+78'

eType パラメータ *eNumGroup_Signed*は、次の結果を返します。

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

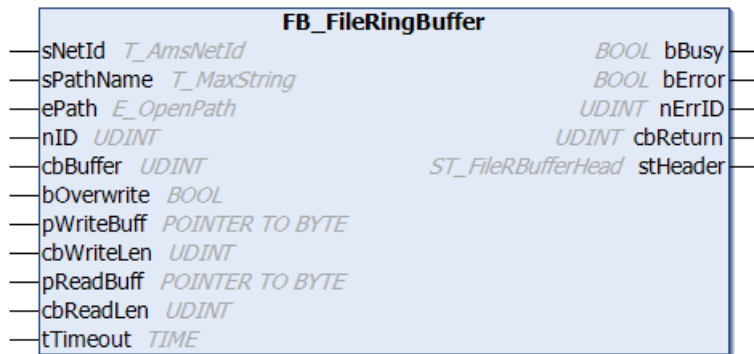
eType パラメータ *eNumGroup_Unsigned*は、次の結果を返します。

- ・ '12'
- ・ '34'
- ・ '56'
- ・ '78'

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.16 FB_FileRingBuffer



FB_FileRingBufferファンクションブロックにより、さまざまな長さのデータレコードをリングバッファファイルに書き込みできるできるようになり、以前にそこに書き込まれたデータレコードをリングバッファファイルから削除できます。書き込まれたデータセットは、FIFO原理(すなわち、最も古いエントリが最も早く読み込まれる)に基づいて以前にリングバッファファイルに書き込まれているされているので、同じ順序で読み込まれます。データレコードを開く、閉じる、書き込む、する、および読み込むは、アクションのコールによって制御されます。ファンクションブロックには、次のタスクがあります。

- ・ **A_Open** (新しいデータセットの追加、または生成のために既存のリングバッファファイルを開きます。ファイルが既に開いている場合、エラーを返します。)
- ・ **A_Close** (開いているリングバッファファイルを閉じます。ファイルが既に閉じている場合、エラーを返します。)
- ・ **A_Create** (新しいリングバッファファイルを開きます。ファイルが既に存在している場合、このファイルは上書きされます。ファイルが既に開いている場合、エラーを返しません。)
- ・ **A_AddTail** (新しいデータレコードをリングバッファファイルに書き込みます。)
- ・ **A_GetHead** (リングバッファファイルから最も古いデータレコードを読み込みますが、削除しません。データポインタは次のデータレコードに移動しません。)
- ・ **A_RemoveHead** (リングバッファファイルから最も古いデータレコードを読み込み、削除します。データポインタは次のデータレコードに移動します。)
- ・ **A_Reset** (開いているリングバッファファイルからすべてのデータレコードを削除します。データポインタとデータレコード数のみがリセットされます。既存のファイルの物理的なサイズは変わりません。ただし、最も古いデータレコードは新しいデータレコードで上書きされます。)

既に存在しているリングバッファファイルが開かれている場合、ファイルヘッダーが最初に読み込まれます。エラーなしで以前に閉じられたリングバッファファイルのヘッダーステータス(Header.status.Bit 0)のビット0は、ゼロでなければいけません。ゼロでない場合、ファイルは以前に適切に閉じられていなかったと想定され、破損したファイルは新しい空のファイルと置き換えられ、一方、Header.status.Bit 1は1(ファイルが破損)にセットされます。ファイルが閉じている場合、Header.status.Bit 0は0にセットされ、ファイルヘッダー全体がファイルで更新されます。

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: バッファファイルが書き込まれ、読み込まれるTwinCATコンピュータのネットワークアドレスを含む文字列をここで指定できます (T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

sPathName: 開くバッファファイルのパスとファイル名を含みます (T_MaxString)。Notice: パスはローカルコンピュータのファイルシステムのみを指すことができます。これは、ネットワークパスはここでは使用できないということです。

ePath: この入力を使用して、ファイルを開くためのターゲットデバイス上のTwinCATシステムパスを選択します (E_OpenPath)。

nID: ユーザ定義の32ビット値。新しいファイルを開くと、この値はファイルに保存されるので、たとえば、バッファファイルのバージョンのチェックに使用できます。

cbBuffer: 開かれるバッファファイルの最大サイズ(バイト単位)。このパラメータはファイルが作成されたときにファイルヘッダーに保存され、同じファイルを再度開くときにチェックされます。同じ最大バッファサイズを使用して、作成したファイルのみを再度開くことができます。つまり、小さいバッファサイズでファイルを作成し、データレコードをファイルに入力してから、そのファイルを大きいバッファサイズで再度、開くことはできません。最大バッファサイズのチェックに失敗した場合、新しいバッファサイズをもつ新しいファイルが自動的に作成され、開きます。また、ビット1 (ファイルが破損)がファイルヘッダーステータスにセットされます。

bOverwrite: 最大ファイルバッファサイズに達したときの、書き込み動作動作。この入力にTRUEがアサートされていて、最大ファイルバッファサイズに既に達した場合、最も古いエントリは上書きされます(新しいエントリを保存するために、バッファサイズが十分な空き容量になるまでエントリは削除されます)。この入力がFALSEになっていて、最大ファイルバッファサイズに達した場合、バッファオーバーフローエラーが報告されます。

pWriteBuff: 書き込まれる値される値データを含むPLC変数、またはバッファ変数のアドレス。アドレスは、ADR演算子で判定できます。cbWriteLenデータバイトを取得できるバッファ変数のサイズは、プログラマが決定する必要があります。

cbWriteLen: 書き込まれる値データバイト数。(文字列変数の場合には、これは最後のヌルを含みます)。

pReadBuff: 読み込まれる値データがコピーされるPLC変数、またはバッファ変数のアドレス。アドレスは、ADR演算子で判定できます。cbReadLenデータバイトを受け入れることができるバッファ変数のサイズは、プログラマが決定する必要があります。バッファ変数のサイズ(バイト単位)は、読み込まれるデータレコードのサイズ以上でなければいけません。

cbReadLen: 読み込まれる値データバイト数。バッファサイズが小さすぎる場合、データはコピーされません。ファンクションブロックはバッファアンダーフローエラーを報告し、読み込まれる次のデータレコードに必要なバッファサイズをcbReturn出力で返します。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```

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

```

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、bBusy出力がリセットされるとこの出力をセットしません。

nErrId: *bError*出力のセット時にADSエラー番号 [▶ 342]、またはコマンド固有のエラーコード(テーブル)を返します。

cbReturn: 正常に読み込んだ値した値データバイト数。読み込みバッファバッファのアンダーフローエラーが発生した場合、この出力は読み込みバッファサイズをバイト単位で返します。

stHeader: リングバッファファイルのヘッダー/ステータス (ST_FileRBufferHead [▶ 295])。

コマンド固有のエラーコード	エラー説明
0x8000	書き込み: ファイルバッファが空です。読み込み: ファイルバッファのオーバーフロー。
0x8001	PLCアプリケーション: 読み込みバッファのアンダーフロー (pReadBuff、cbReadLen)。バッファサイズが小さすぎます。
0x8002	リングバッファファイルが閉じています。最初に開く必要があります。
0x8003	入力パラメータは不正な値です。

PLCアプリケーションがファイルのバイナリ構造を認識することは重要ではありません。ただし、以下の説明は使用するリングバッファファイルの一般的な構造を示しています。

ヘッダー (48バイト)。 ST_FileRBufferHeadの説明を参照	データセット1の長さ (4バイト)	データセット1	データセット2の長さ (4バイト)	データセット2	データセット3の長さ (4バイト)	データセット3	データセットnの長さ (4バイト)	データセットn
---	-------------------	---------	-------------------	---------	-------------------	---------	-------------------	---------

空のリングバッファファイルだけにヘッダーが含まれています。バッファ自体はヘッダーの後に続きます。変数Header.ptrFirstおよびHeader.ptrLastはヘッダーのすぐ後ろの位置を指します。書き込みにより、ptrLastデータポインタは前方に移動します。ptrFirstデータポインタは、読み込み中にptrLastデータポインタの後に続きます。最大バッファサイズに達すると、ポインタはバッファの初めに戻ります。

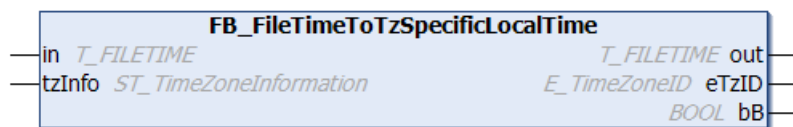
例:

参照: [例: ファイルリングFiFo \(FB_FileRingBuffer\)](#). [▶ 317]

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.17 FB_FileTimeToTzSpecificLocalTime

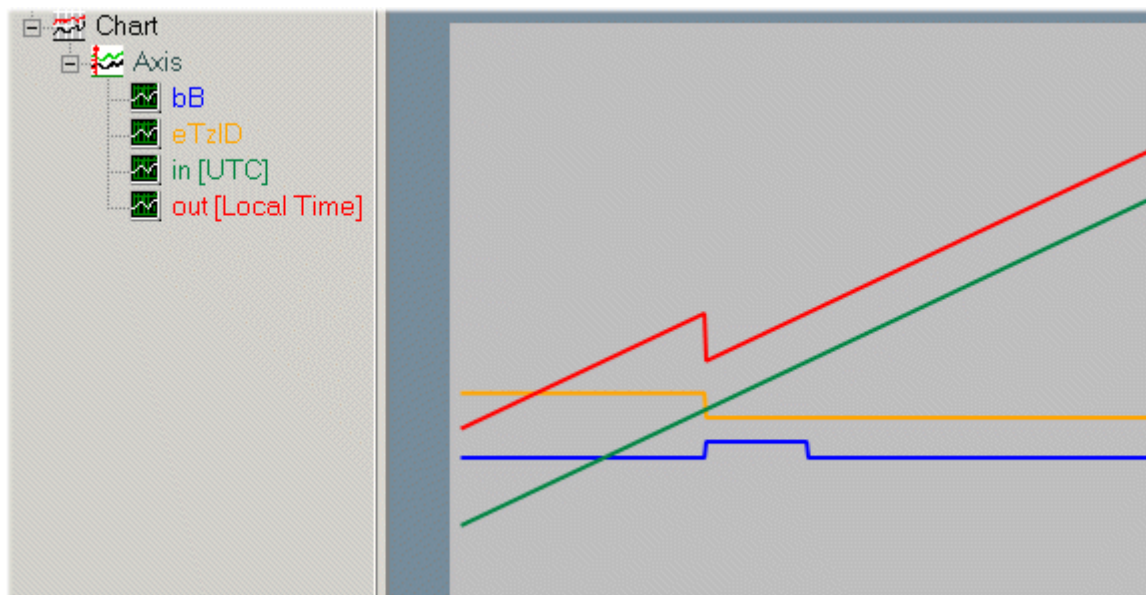


指定したタイムゾーン情報を考慮して、ファンクションブロックはUTC時間(ファイル時間の形式)を現地時間(ファイル時間の形式)に変換します。ファンクションブロックFB_SystemTimeToTzSpecificLocalTime [▶ 104]は同様の機能を持っていますが、異なる時間の形式(構造体システム時間の形式)を使用します。

ブロックは、**連続的な**UTCタイムスタンプ情報の変換にのみ適しています。ファンクションブロックはタイムゾーン情報を使用して、現地時間で必要な時間ステップ(夏時間/冬時間の切り替え)を計算します。UTC入力時間での時間ステップは、不正な変換になるため許可されていません。理由: ブロックは最後に変換された時間を内部的に保存し、そのため現地時間が変更されたときにUTC入力時間と保存された値からB時間(下記参照)を検出できます。

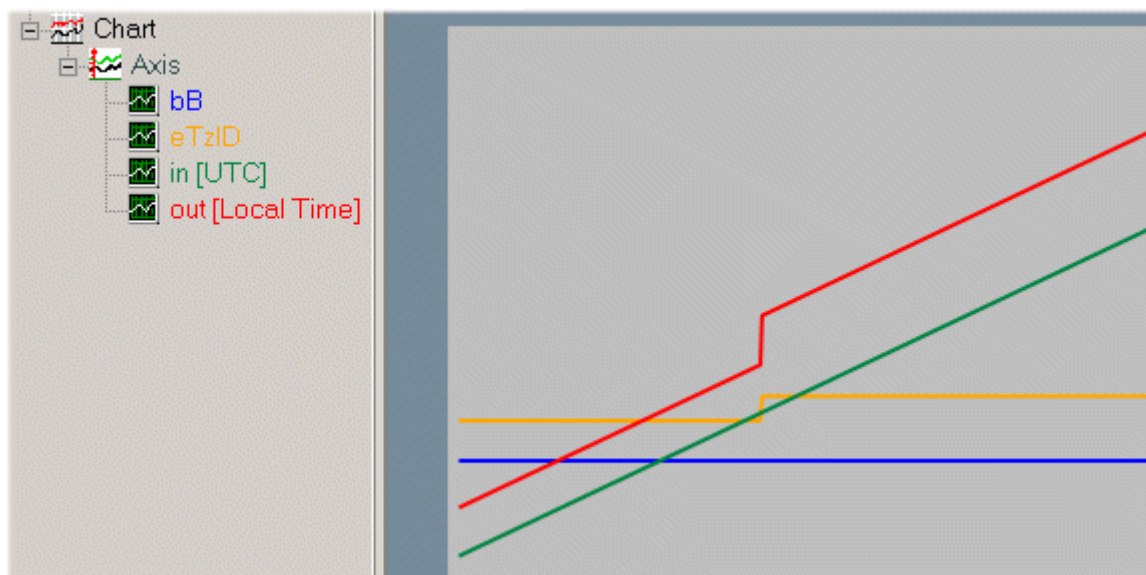
ブロックはアクションと関連があります: `A_Reset()`。このアクションがブロック出力をコールし、ローカルに保存された(最後に変換された)時間がゼロにリセットされます。

1. 夏時間から冬時間への切り替えのグラフ表示 (tzInfo = WEST_EUROPE_TZI):



UTC入力時間(緑色)は連続的です。現地時間(赤)はジャンプバックします。現地時間: **02h:59m:59s:999ms..** の後にすぐに**02h:00m:00s:000ms..**が続きます。2時と3時の間の時間が2度発生します。切り替え前の複製時間が**02:05:00 GEST A**として参照されます。たとえば、切り替え後の時間は**02:05:00 CET B**として参照されます。出力変数**bB**は、最初または2番目のパスかどうかを示します。2番目のパスの間に、**bB**出力変数(青)はTRUEにセットされます。複製時間が経過すると、**bB**出力変数は自動的にリセットされます。タイムゾーンID(オレンジ)は、*eTimeZoneID_Daylight*(夏時間)から*eTimeZoneID_Standard*(冬時間)に変わります。

2. 冬時間から夏時間 (tzInfo = WEST_EUROPE_TZI) への切り替えのグラフ表示。



UTC入力時間(緑色)は連続的です。現地時間(赤)ジャンプフォワードします。現地時間: **2h:59m:59s:999ms..** の後にすぐに: **3h:00m:00s:000ms..**が続きます。タイムゾーンID(オレンジ)は、*eTimeZoneID_Standard*(冬時間)から*eTimeZoneID_Daylight*(夏時間)に変化します。

VAR_INPUT

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

in: 変換するUTC時間(ファイル時間の形式) ([T_FILETIME \[▶ 302\]](#))。

tzInfo: OSの現在のタイムゾーン情報をもつ構造体変数 ([ST_TimeZoneInformation \[▶ 300\]](#))。

VAR_OUTPUT

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

out: 変換された現地時間(ファイル時間の形式、[T_FILETIME \[▶ 302\]](#))。

eTzID: 追加の夏時間/冬時間情報。 ([E_TimeZoneID \[▶ 288\]](#))

bB: TRUE → B時間(例: 02:05:00 CET B)、FALSE → 他の時間(例: 02:05:00 CEST A)。この出力は現地時間がジャンプバックするとセットされ、複製時間が経過するとリセットされます。

例:

UTC時間: DT#2011-09-02-09:01:31は現地時間に変換されます。結果: 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 );
```

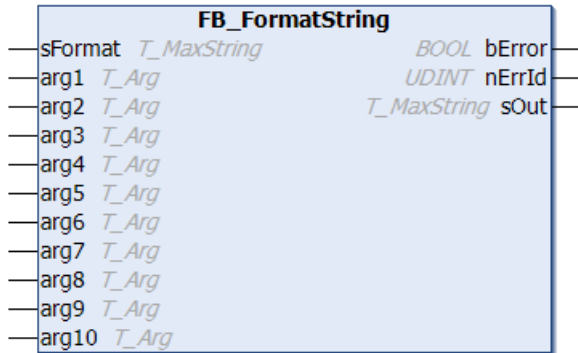
時間とタイムゾーンのその他のファンクションとファンクションブロック

- ・ [FB_TzSpecificLocalTimeToSystemTime \[▶ 108\]](#)
- ・ [FB_TzSpecificLocalTimeToFileTime \[▶ 106\]](#)
- ・ [FB_SystemTimeToTzSpecificLocalTime \[▶ 104\]](#)
- ・ [FB_GetTimeZoneInformation \[▶ 69\]](#)
- ・ [FB_SetTimeZoneInformation \[▶ 101\]](#)
- ・ [NT_SetLocalTime \[▶ 115\]](#)
- ・ [NT_GetTime \[▶ 113\]](#)
- ・ [NT_SetTimeToRTCTime \[▶ 116\]](#)
- ・ [F_TranslateFileTimeBias \[▶ 144\]](#)
- ・ [FB_LocalSystemTime \[▶ 81\]](#)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.18 FB_FormatString



このファンクションブロックを使用して、最大10個の引数 (fprintfと同様) を文字列に変換でき、フォーマット仕様 [▶ 338] にしたがってフォーマットできます。フォーマットは同じPLCサイクルで実行され、FBがコールされた後で出力文字列は直ちに利用できます。

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: 文字列のフォーマット仕様 (T_MaxString) (e.g. '+20.5f' or 'Measure X: %+..10d, Y: %+..10d')。

arg1~arg10: フォーマットされる引数 (T_Arg [▶ 302])。以下の補助ファンクションを使用して、異なるタイプのPLC変数を必要なデータ型 T_Arg [▶ 302] に変換できます。 F_BYTE [▶ 218]、F_WORD [▶ 225]、F_DWORD [▶ 219]、F_LWORD [▶ 221]、F_SINT [▶ 222]、F_INT [▶ 219]、F_DINT [▶ 218]、F_LINT [▶ 220]、F_USINT [▶ 225]、F_UINT [▶ 224]、F_UDINT [▶ 223]、F_ULINT [▶ 224]、F_STRING [▶ 222]、F_REAL [▶ 221]、F_LREAL [▶ 221]。

VAR_OUTPUT

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

bError: フォーマット中にエラーが発生した場合、この出力をセットします。

nErrId: bError出力がセットされている場合、フォーマットエラーコード [▶ 341] を返します。

sOut: 成功した場合、この出力はフォーマットされた出力文字列を返します (T_MaxString)。

例:

```
PROGRAM MAIN
VAR
  fbFormat     : FB_FormatString;
  iY           : DINT;
  iX           : DINT;
  bError       : BOOL;
END_VAR
```

```

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
);

```

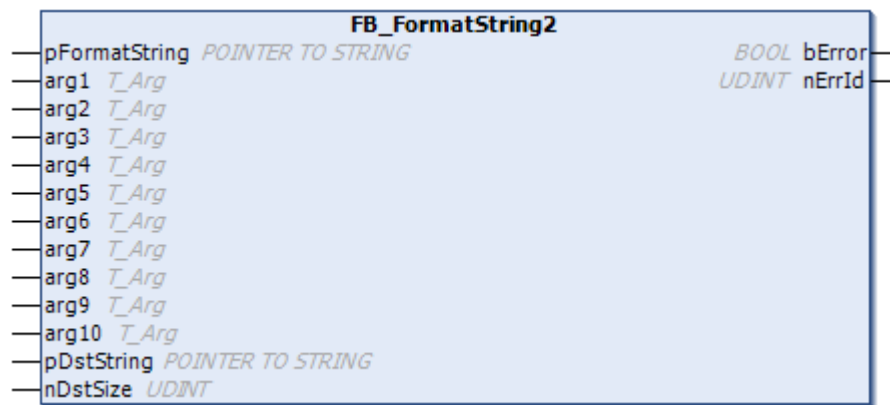
結果:

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

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.19 FB_FormatString2



このファンクションブロックを使用して、最大10個の引数 (fprintfと同様) を文字列に変換でき、フォーマット仕様 [▶ 338]にしたがってフォーマットできます(例: 「%+20.5f」または「Measure X: %+.10d, Y: %+.10d」)。フォーマットは同じPLCサイクルで実行されます。これは、ファンクションブロックをコールした後で、直ちに出力文字列を使用できるということです。

ファンクションブロックFB_FormatString [▶ 55]とは対照的に、フォーマット文字列と出力文字列のサイズは255文字に制限されません。ただし、各引数は最大250文字表記に制限されます。文字数が超過した場合、または出力文字列がフォーマットされた文字列に対し小さすぎる場合、ファンクションブロックはエラーを出力します。

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: 文字列のフォーマット仕様へのポインタ。ファンクションブロックがコールされるごとに、アドレスを割り当てる必要があります。割り当てには、演算子ADR()を使用できます。

arg1 to arg10: 引数がフォーマットされます(T_Arg [▶ 302])。次のヘルパーファンクションを使用して、異なるタイプのPLC変数を必要なデータ型T_Arg [▶ 302]に変換できます。F_BYTE [▶ 218]、F_WORD [▶ 225]、F_DWORD [▶ 219]、F_LWORD [▶ 221]、F_SINT [▶ 222]、F_INT [▶ 219]、F_DINT [▶ 218]、F_LINT [▶ 220]、F_USINT [▶ 225]、F_UINT [▶ 224]、F_UDINT [▶ 223]、F_ULINT [▶ 224]、F_STRING [▶ 222]、F_REAL [▶ 221]、F_LREAL [▶ 221]のヘルパーファンクション。

pDstString: 生成される文字列変数へのポインタ。成功した場合、フォーマットされた文字列がここに書き込まれます。ファンクションブロックがコールされるごとに、アドレスを割り当てる必要があります。割り当てには、演算子ADR()を使用できます。

nDstSize: 生成される文字列変数のサイズ(バイト単位)。割り当てには、演算子sizeof()を使用できます。

VAR_OUTPUT

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

bError: フォーマット中にエラーが発生するとTRUEとなります。

nErrId: bError出力がセットされている場合、フォーマットエラーコード [▶ 341]を返します。

サンプル

```
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 );
```

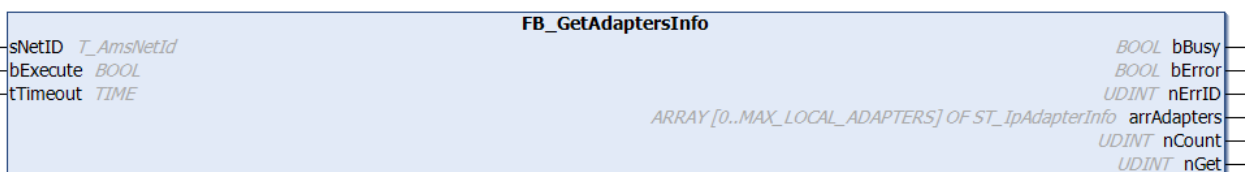
結果:

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

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、3.3.35.0以降

3.20 FB_GetAdaptersInfo



このファンクションブロックを使用して、TwinCAT PC用にアダプタ情報を読み込みできます。読み込むことのできるアダプタデータの最大数は、現在、MAX_LOCAL_ADAPTERS + 1 (デフォルト = 6)に制限されます。

VAR_INPUT

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

sNetID: このパラメータを使用して、そのアダプタ情報が読み込まれるTwinCATコンピュータのAmsNetIDを指定できます。ローカルPCの場合、空の文字列を指定できます (T_AmsNetID)。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: コマンドの実行中に、超過してはならないタイムアウトの長さを表します。

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: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットします。

nErrId: *bError*出力がセットされると、ADSエラー番号 [▶ 342]を返します。

arrAdapters: 最後に読み込んだアダプタ情報を格納する配列変数。各配列エレメントは、1台のアダプタに情報を提供します (ST_IpAdapterInfo [▶ 297])。

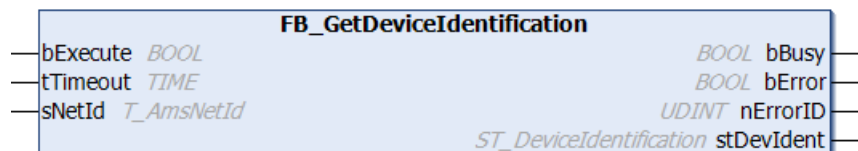
nCount: 検出されたローカルアダプタの最大数。

nGet: *arrAdapters*出力変数内の有効なエントリ数。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.21 FB_GetDeviceIdentification



ブロックはデバイスIDを読み込みします。ハードウェア型式とハードウェアシリアル番号文字列が長い場合、ブロック FB_GetDeviceIdentificationEx [▶ 59]を使用します。

VAR_INPUT

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

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

sNetId: このパラメータを使用して、そのデバイスIDが読み込まれるTwinCATコンピュータのAmsNetIDを指定できます (T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後でこの出力をセットします。

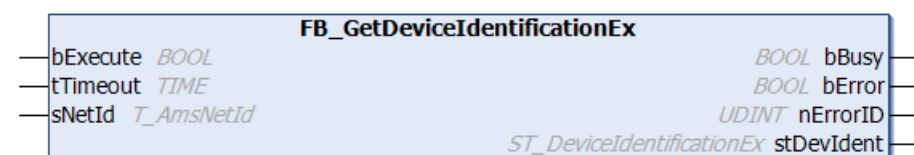
nErrorId: bError出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

stDevIdent: デバイスIDを提供します ([ST_DeviceIdentification \[▶ 292\]](#))。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.22 FB_GetDeviceIdentificationEx



ブロックはデバイスIDを読み込みます。ハードウェア型式とハードウェアシリアル番号用に [FB_GetDeviceIdentification \[▶ 58\]](#)より長い文字列を許容します。

VAR_INPUT

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

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

sNetId: このパラメータを使用して、デバイスIDが読み込まれるTwinCATコンピュータのAmsNetIDを指定できます (T_AmsNetId)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後でこの出力をセットします。

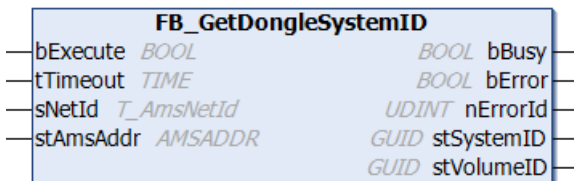
nErrorId: bError出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

stDevIdent: デバイスIDを提供します ([ST_DeviceIdentificationEx \[▶ 293\]](#))。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.23 FB_GetDongleSystemID



ファンクションブロックFB_GetDongleSystemIDは、TwinCAT 3ライセンス dongle のシステムIDとボリュームIDをGUID値で読み込みします。

VAR_INPUT

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

bExecute: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: コマンドが実行される時に、超過してはならないタイムアウト時間。

sNetId: そのライセンスステータスが読み込まれるTwinCATコンピュータのAmsNetId (AMSネットワーク識別子) (T_AmsNetId)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

stAmsAddr: ライセンス dongle のネットワークアドレス (AmsNetIdとポート) (AMSADDR)。

VAR_OUTPUT

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

```

stSystemID : GUID;
stVolumeID : GUID;
END_VAR

```

bBusy: ファンクションブロックがアクティブな間はTRUEです。

bError: コマンド実行中にエラーが発生するとTRUEとなります。

nErrorId: bError出力がセットされると、ADSエラー番号を返します。

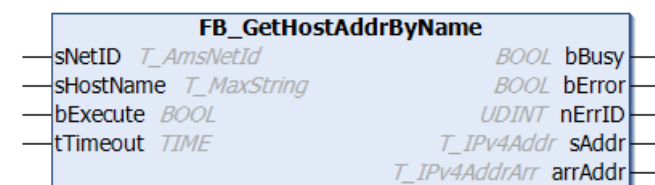
stSystemID: ライセンスドングルのシステムID (GUID)。

stVolumeID: ライセンスドングルのボリュームID (GUID)。

要件

開発環境	ターゲットシステム	必要なPLCライブラリ
TwinCAT v3.1.4022	PCまたはCX (x64, x86)	Tc2_Utilities >= 3.3.24.0

3.24 FB_GetHostAddrByName



このファンクションブロックを使用して、指定したホスト名のインターネットプロトコルのネットワークアドレス (IPv4) を読み込みできます。アドレスは、文字列およびバイト列として返されます。

VAR_INPUT

```

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

```

sNetID: コマンドが実行されるTwinCATコンピュータのネットワークアドレスを指定できます (T_AmsNetId)。ローカルコンピュータ (デフォルト) の場合、空の文字列を指定できます。

sHostName: 文字列のホスト名 (T_MaxString)。例: 「DataServer1」。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: コマンドの実行中に、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```

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

```

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットしません。

nErrID: *bError*出力がセットされると、ADSエラー番号 [▶ 342]を返します。

sAddr: 文字列のインターネットプロトコルのネットワークアドレス (IPv4) (T_Ipv4Addr)。例:
「172.16.7.199」

arrAddr: バイト配列のインターネットプロトコルのネットワークアドレス (T_Ipv4AddrArr)。

サンプル:

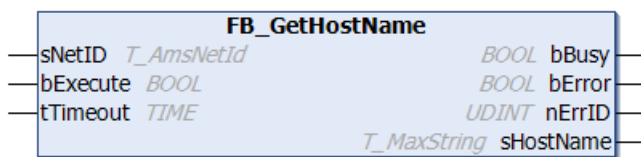
```
PROGRAM MAIN
VAR
    fbGet    : FB_GetHostAddrByName;
    bGet     : BOOL := TRUE;
    bError   : BOOL;
    nErrID   : UDINT;
    sIPv4    : T_IPv4Addr; (* Result: '87.106.8.100' *)
    arrIPv4  : T_IPv4AddrArr;
    state    : BYTE;
END_VAR

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
END_PROGRAM
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.25 FB_GetHostName



このファンクションブロックを使用して、TwinCAT PCのホスト名を読み込みできます。

VAR_INPUT

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

sNetID: このパラメータを使用して、そのホスト名が読み込まれるTwinCATコンピュータのネットワークアドレスを指定できます (T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: コマンドの実行中に、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットしません。

nErrId: *bError*出力がセットされると、ADSエラー番号 [▶ 342]を返します。

sHostName: 文字列のホスト名 (T_MaxString)。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3. 26 FB_GetLicenseDongle



ファンクションブロックは、接続されたライセンス dongle 数を判定し、アドレスとステータスを返します。

VAR_INPUT

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

bExecute: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: コマンドが実行されるときに、超過してはならないタイムアウト時間。

sNetId: そのライセンスステータスが読み込まれるTwinCATコンピュータのAmsNetId (AMSネットワーク識別子) (T_AmsNetId)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

VAR_OUTPUT

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

bBusy: ファンクションブロックがアクティブな間はTRUEです。

bError: コマンド実行中にエラーが発生するとTRUEとなります。

nErrorId: bError出力がセットされると、ADSエラー番号を返します。

nLicenseDeviceDongles: ライセンス dongle 数。

aLicenseDeviceDongles: 接続されたライセンス dongle の識別データ。

要件

開発環境	ターゲットシステム	必要なPLCライブラリ
TwinCAT v3.1.4022	PCまたはCX (x64、x86)	Tc2_Utilities >= 3.3.24.0

3.27 FB_GetLicenses



ファンクションブロックは、有効および無効なTwinCATライセンスを読み込みします。

VAR_INPUT

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

bExecute: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

sNetId: このパラメータで、現在のTwinCATライセンスが読み込まれるTwinCATコンピュータのAmsNetIDを指定します (T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

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: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後でこの出力をセットします。

nErrorId: bError出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

nValidLicenses: 有効なTwinCATライセンス数を返します。

aValidLicenses: データ型[ST_TcOnlineLicenseInfodata \[▶ 299\]](#)の有効なTwinCATライセンスのリストを返します。

nInvalidLicenses: 無効なTwinCATライセンス数を返します。

aInvalidLicenses: データ型 `ST_TcOnlineLicenseInfoData` [▶ 299] の無効な TwinCAT ライセンスのリストを返します。



デフォルトでは、ライセンスリストのエントリの最大数は50です。この制限は、`nMaxLicenses` を使用してライブラリのパラメータリストで変更できます。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.4018	PC または CX (x86、x64、ARM)	Tc2_Utilities (システム) v3.3.9.0以降

3.28 FB_GetLicensesEx

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

ファンクションブロック `FB_GetLicensesEx` は、すべての TwinCAT 3 ライセンスと OEM ライセンスのステータスを判定します。

VAR_INPUT

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

bExecute: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: コマンドが実行されるときに、超過してはならないタイムアウト時間。

sNetId: そのライセンスステータスが読み込まれる TwinCAT コンピュータの `AmsNetId` (AMS ネットワーク識別子) (`T_AmsNetId`)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

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
```

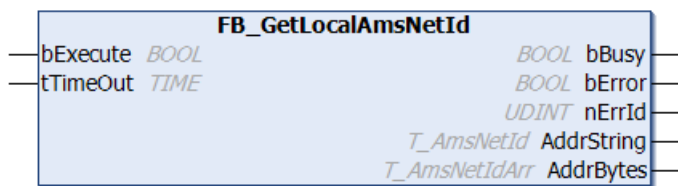
```
nFailedLicenses : UDINT
aFailedLicenses : ARRAY[1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx
nInvalidLicenses : UDINT
aInvalidLicenses : ARRAY[1..nMaxLicenses] OF ST_TcOnlineLicenseInfoDataEx
END_VAR
```

- bBusy**: ファンクションブロックがアクティブな間はTRUEです。
- bError**: コマンド実行中にエラーが発生するとTRUEとなります。
- nErrorId**: bError出力がセットされると、ADSエラー番号を返します。
- nValidLicenses**: 有効なライセンス数
- aValidLicenses**: 有効なライセンスについての情報
- nPendingLicenses**: オープンライセンス数
- aPendingLicenses**: 未処理のライセンスについての情報
- nDemoLicenses**: 有効なデモライセンス数
- aDemoLicenses**: 有効なデモライセンスについての情報
- nOemLicenses**: 有効なOEMライセンス数
- aOemLicenses**: 有効なOEMライセンスについての情報
- nFailedLicenses**: 失敗したライセンス数
- aFailedLicenses**: 有効なライセンスについての情報
- nInvalidLicenses**: 無効なライセンス数
- aInvalidLicenses**: 無効なライセンスについての情報

要件

開発環境	ターゲットシステム	必要なPLCライブラリ
TwinCAT v3.1.4022	PCまたはCX (x64, x86)	Tc2_Uilities >= 3.3.24.0

3.29 FB_GetLocalAmsNetId



このファンクションブロックを使用して、ローカルTwinCAT PCのネットワークアドレス (AmsNetID) を読み込みできます。

VAR_INPUT

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

- bExecute**: ブロックは、この入力での立ち上がりエッジによって有効になります。
- tTimeout**: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```

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

```

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットしません。

nErrId: *bError*出力がセットされると、ADSエラー番号 [▶ 342]を返します。

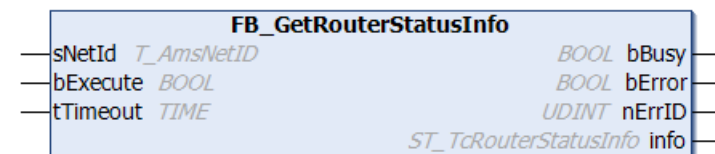
AddrString: 文字列のローカルPCのAmsNetID (T_AmsNetID)。

AddrBytes: バイト列のローカルPCのAmsNetID (T_AmsNetIDArr)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.30 FB_GetRouterStatusInfo



ファンクションブロックFB_GetRouterStatusInfoを使用して、PLCからTwinCATルータのステータス情報を読み込みできます(使用可能なメモリ、登録したポート数など)。

VAR_INPUT

```

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

```

sNetId: TwinCATルータ情報が読み込まれるTwinCATコンピュータのネットワークアドレスをもつ文字列を入力できます(T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```

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

```

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットしません。

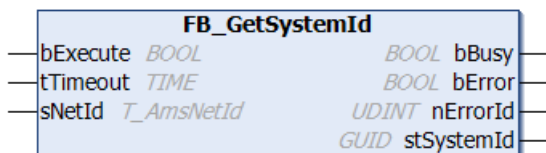
nErrId: *bError*出力がセットされると、ADSエラー番号 [▶ 342]を返します。

info: TwinCATルータステータス情報をもつ構造体変数 (ST_TcRouterStatusInfo [▶ 300])。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.31 FB_GetSystemId



ファンクションブロックは、システムIDをGUID値で読み込みします (システムトレイのTwinCATアイコンで [About TwinCAT...] を参照してください)。

VAR_INPUT

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

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

sNetId: このパラメータを使用して、そのシステムIDが読み込まれるTwinCATコンピュータのAmsNetIDを指定できます (T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンドの送信中にエラーが発生した場合、*bBusy*出力がリセットされた後でこの出力をセットします。

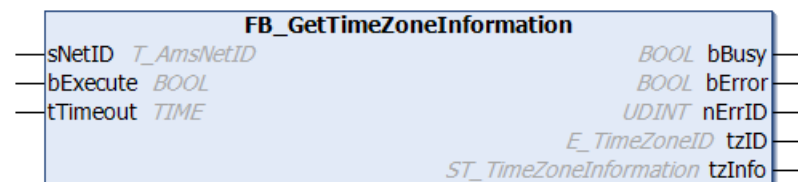
nErrorId: *bError*出力がセットされると、ADSエラー番号 [▶ 342]を返します。

stSystemId: システムIDをGUID [▶ 290]値で返します。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.32 FB_GetTimeZoneInformation



このファンクションブロックを使用して、OSのタイムゾーン設定を読み込みできます。

VAR_INPUT

```

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

```

sNetID: そのタイムゾーン設定が読み込まれるTwinCATコンピュータのネットワークアドレスをもつ文字列を入力できます (T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```

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

```

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットします。

nErrId: *bError*出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

tzID: 追加の夏時間/冬時間情報(常に存在するとは限らない) ([E_TimeZoneID \[▶ 288\]](#))。

tzInfo: 成功した場合、この構造体変数はOSの現在のタイムゾーン情報を返します ([ST_TimeZoneInformation \[▶ 300\]](#))。

例:

[FB_SetTimeZoneInformation \[▶ 101\]](#)ファンクションブロックの説明を参照してください。

時間とタイムゾーンのその他のファンクションとファンクションブロック

- ・ [FB_TzSpecificLocalTimeToSystemTime \[▶ 108\]](#)
- ・ [FB_TzSpecificLocalTimeToFileTime \[▶ 106\]](#)
- ・ [FB_SystemTimeToTzSpecificLocalTime \[▶ 104\]](#)

- ・ [FB_FileTimeToTzSpecificLocalTime](#) [[▶ 52](#)]
- ・ [FB_SetTimeZoneInformation](#) [[▶ 101](#)]
- ・ [NT_SetLocalTime](#) [[▶ 115](#)]
- ・ [NT_GetTime](#) [[▶ 113](#)]
- ・ [NT_SetTimeToRTCTime](#) [[▶ 116](#)]
- ・ [F_TranslateFileTimeBias](#) [[▶ 144](#)]
- ・ [FB_LocalSystemTime](#) [[▶ 81](#)]

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.33 FB_GetVolumeId



ファンクションブロックFB_GetVolumeIdは、システムIDとボリュームシステムIDをGUID値で読み込みします。

VAR_INPUT

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

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

sNetId: そのシステムIDが読み込まれるTwinCATコンピュータのAmsNetId (T_AmsNetId)はここで指定できません。ローカルコンピュータで実行する場合、空の文字列を入力できます。

VAR_OUTPUT

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

bBusy: ファンクションブロックの実行中に確認レスポンスを受信するまでセットされたままの状態となります。

bError: コマンドの送信中にエラーが発生した場合、bBusy出力がリセットされた後でこの出力をセットします。

nErrorId: bError出力がセットされると、[ADSエラー番号](#) [[▶ 342](#)]を返します。

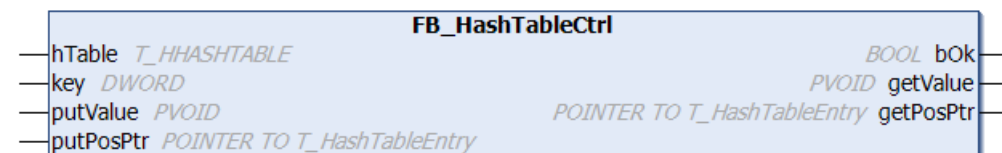
stVolumeId: ボリュームシステムIDをGUID [[▶ 290](#)]値で返します。

stSystemId: システムIDをGUID [▶ 290]値で返します。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4018	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities

3.34 FB_HashTableCtrl



ハッシュテーブルは、たくさんのデータエレメントの中から個々のデータエレメントを検出するための効率的なツールとして使用できます。データオブジェクトは固有のキーをもつ必要があります。キーにより、データオブジェクトを明確に特定可能で、テーブルの中から迅速に検出できます。

ファンクションブロックFB_HashTableCtrlを使用して、チェイニング(分離連鎖法)プロシージャによるハッシュ法によりPLCプロジェクトでシンプルなハッシュテーブルを実現できます。チェイニング(分離連鎖法)プロシージャによるハッシュ法を使用します。

データエレメントの最大数は実行時には変更できないので、事前に指定する必要があります。データエレメントの追加/削除/検出は、アクションのコールにより制御されます。ファンクションブロックには、次のタスクがあります。

- ・ **A_Add** (新しいデータエレメントをテーブルに追加(キー/値)。同じキーをもつエレメントが既に存在する場合、エレメントは上書きされます。)
- ・ **A_GetFirst** (最初のテーブルデータエレメントを読み込みします。成功した場合、*getValue*は関連した値を返します。)
- ・ **A_GetNext** (次のテーブルのデータエレメントを読み込みします。アドレス*putPosPtr*は、前のデータエレメントを指さなければいけません)
- ・ **A_Lookup** (キーと一致するデータエレメントを探します。成功した場合、*getValue*は関連した値を返します。)
- ・ **A_Remove** (キーと一致するデータエレメントを削除します。)
- ・ **A_RemoveAll** (すべてのデータエレメントを削除します。)
- ・ **A_RemoveFirst** (最初のデータエレメントを削除します。)
- ・ **A_reset** (すべてのデータエレメントを削除し、テーブルをリセットします。)
- ・ **A_GetIndexAtPosPtr** (アドレス*putPosPtr*でのデータエレメントの配列インデックスを返します。成功した場合、*getValue*はヌルベースの配列インデックスを返します。*putValue*の値は使用されません。値*getValue*はデータエレメントインデックスを戻し、データエレメント値ではないことに注意してください)

VAR_IN_OUT

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

hTable: ハッシュテーブルハンドル([T_HHASHTABLE \[▶ 304\]](#))。ハンドルは使用する前に、ファンクション [F_CreateHashTableHnd \[▶ 245\]](#) で一度、初期化する必要があります。ハンドル変数の対応するインスタンスを作成し、各テーブル用に初期化する必要があります。

VAR_INPUT

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

key: キー(符号なし32ビット数またはポインタ)。このキーにより、データオブジェクトを特定可能で、テーブルの中から迅速に検出できます。

putValue: 値/データエレメント(入力パラメータ、32/64ビット、符号なし数またはポインタ)。

putPosPtr: データエレメントのアドレス(入力パラメータ、[T_HashTableEntry \[▶ 303\]](#))。

VAR_OUTPUT

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

bOk: 新しいデータエレメントがテーブルに追加/テーブルから削除、またはテーブルで検出された場合は、TRUEを返します。検索したデータエレメントが検出されなかった、テーブルが空になった、またはオーバーフローが発生した場合(テーブルは空のデータエレメントを持ちません)、FALSEを返します。

getValue: キー/データエレメントに一致する値(出力パラメータ、32/64ビット、符号なし数またはポインタ)。

getPosPtr: データエレメント用のアドレス(出力パラメータ、[T_HashTableEntry \[▶ 303\]](#))。

例:

参照: [例: ハッシュテーブル\(FB HashTableCtrl\)](#)。 [[▶ 322](#)]

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.35 FB_LicFileCopyFromDongle



このファンクションブロックは、ライセンス dongle からハードディスクにファイルをコピーします。ファイルがバッファ (*cbCopyLen*) よりも大きい場合、ファイル全体がコピーされるまでファイルコピー手順が自動的に複数の読み込み/書き込み操作に分割されます。その後でのみ、*bBusy* が FALSE に切り替わります。

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: ライセンス ドングルのAmsNetId (AMSネットワークID) (T_AmsNetId)

- ・ USB ドングル: TwinCAT コンピュータのAmsNetId。ローカルコンピュータで実行する場合、空の文字列を入力できます。
- ・ EL6070: EtherCAT マスタのAmsNetId (EL6070のInfoDataのAdsAddr.netIdを参照)

nPortSrc: ライセンス ドングルのAMSポート

- ・ USB: ESB デバイスのADSポート (USB ドングルの [ESB Device] タブのADSポートを参照、デフォルトは 16#7100)
- ・ EL6070: EtherCAT ターミナルのADSポート (EL6070のInfoDataのAdsAddr.portを参照)

sNetIdDest: TwinCAT コンピュータのAmsNetId (AMSネットワークID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

sFileNameSrc: ライセンス ドングル上のファイル名

sFilePathNameSrc: ハードディスク上のファイルのパス名 (T_MaxString)

pCopyBuff: 書き込み用のバッファアドレス

cbCopyLen: 書き込み用のバイトカウント

bExecute: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

dwPassCode: ファイルアクセス用のパスワード

tTimeout: コマンドが実行されるときに、超過してはならないタイムアウト時間。

VAR_OUTPUT

```

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

```

END_VAR

bBusy: ファンクションブロックがアクティブな間はTRUEです。

bError: コマンド実行中にエラーが発生するとTRUEとなります。

nErrorId: bError出力がセットされると、ADSエラー番号を返します。

要件

開発環境	ターゲットシステム	必要なPLCライブラリ
TwinCAT v3.1.4022	PCまたはCX (x64、x86)	Tc2_Utilities、3.3.26.0以降

3.36 FB_LicFileCopyToDongle

FB_LicFileCopyToDongle		
sNetIdSrc	T_AmsNetId	BOOL bBusy
sNetIdDest	T_AmsNetId	BOOL bError
nPortDest	UINT	UDINT nErrorId
sFilePathNameSrc	T_MaxString	
sFileNameDest	STRING	
pCopyBuff	PVOID	
cbCopyLen	UDINT	
bExecute	BOOL	
dwPassCode	DWORD	
tTimeout	TIME	

このファンクションブロックは、ハードディスクからライセンス dongle にファイルをコピーします。ファイルがバッファ (cbCopyLen) よりも大きい場合、ファイル全体がコピーされるまでファイルコピー手順が自動的に複数の読み込み/書き込み操作に分割されます。その後でのみ、bBusyがFALSEに切り替わります。

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: TwinCATコンピュータのAmsNetId (AMSネットワークID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。(T_AmsNetId)

sNetIdDest: ライセンス dongle のAmsNetId (AMSネットワークID) (T_AmsNetId)

- ・ USB dongle: TwinCATコンピュータのAmsNetId。ローカルコンピュータで実行する場合、空の文字列を入力できます。
- ・ EL6070: EtherCATマスタのAmsNetId (EL6070のInfoDataのAdsAddr.netIdを参照)

nPortDest: ライセンス dongle のAMSポート

- ・ USB: ESBデバイスのADSポート (USB dongle の [ESB Device] タブのADSポートを参照、デフォルトは 16#7100)
- ・ EL6070: EtherCATターミナルのADSポート (EL6070のInfoDataのAdsAddr.portを参照)

sFilePathNameSrc: ハードディスク上のファイルのパス名 (T_MaxString)

sFileNameDest: ライセンス dongle 上のファイル名

pCopyBuff: 書き込み用のバッファアドレス

cbCopyLen: 書き込み用のバイトカウント

bExecute: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

dwPassCode: ファイルアクセス用のパスコード

tTimeout: コマンドが実行されるときに、超過してはならないタイムアウト時間。

VAR_OUTPUT

```

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

```

bBusy: ファンクションブロックがアクティブな間はTRUEです。

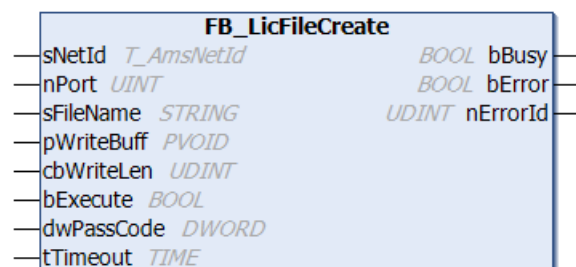
bError: コマンド実行中にエラーが発生するとTRUEとなります。

nErrorId: bError出力がセットされると、ADSエラー番号を返します。

要件

開発環境	ターゲットシステム	必要なPLCライブラリ
TwinCAT v3.1.4022	PCまたはCX (x64, x86)	Tc2_Utilities、3.3.26.0以降

3.37 FB_LicFileCreate



このファンクションブロックは、ライセンス dongle 上でファイルを作成します。bExecuteでの立ち上がりエッジで、バッファ (pWriteBuff および cbWriteLen) から dongle 上の新しいファイルへの直接のデータ書き込みがトリガがトリガされます。

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: ライセンス dongle の AmsNetId (AMS ネットワーク ID) (T_AmsNetId)

- ・ USB dongle: TwinCAT コンピュータの AmsNetId。ローカルコンピュータで実行する場合、空の文字列を入力できます。
- ・ EL6070: EtherCAT マスタの AmsNetId (EL6070 の InfoData の AdsAddr.netId を参照)

nPort: ライセンス dongle の AMS ポート

- ・ USB: ESB デバイスの ADS ポート (USB dongle の [ESB Device] タブの ADS ポートを参照、デフォルトは 16#7100)
- ・ EL6070: EtherCAT ターミナルの ADS ポート (EL6070 の InfoData の AdsAddr.port を参照)

sFileName: 作成されるファイル名

pWriteBuff: 書き込み用のバッファアドレス

cbWriteLen: 書き込み用のバイトカウント

bExecute: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

dwPassCode: ファイルアクセス用のパスコード

tTimeout: コマンドが実行されるときに、超過してはならないタイムアウト時間。

VAR_OUTPUT

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

bBusy: ファンクションブロックがアクティブな間はTRUEです。

bError: コマンド実行中にエラーが発生するとTRUEとなります。

nErrorId: bError出力がセットされると、ADSエラー番号を返します。

要件

開発環境	ターゲットシステム	必要なPLCライブラリ
TwinCAT v3.1.4022	PCまたはCX (x64, x86)	Tc2_Uilities、3.3.26.0以降

3.38 FB_LicFileDelete



このファンクションブロックは、ライセンス dongle からファイルを削除します。ファイル名とファイル長がゼロになります。また、削除されるファイルのデータバイトが dongle 上で解放されます (しかし、上書きはされません)。

VAR_INPUT

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

sNetId: ライセンス dongle の AmsNetId (AMS ネットワーク ID) (T_AmsNetId)

- ・ USB Dongle: TwinCAT コンピュータの AmsNetId。ローカルコンピュータで実行する場合、空の文字列を入力できます。
- ・ EL6070: EtherCAT マスタの AmsNetId (EL6070 の InfoData の AdsAddr.netId を参照)

nPort: ライセンス dongle の AMS ポート

- ・ USB: ESB デバイスの ADS ポート (USB Dongle の [ESB Device] タブの ADS ポートを参照、デフォルトは 16#7100)
- ・ EL6070: EtherCAT ターミナルの ADS ポート (EL6070 の InfoData の AdsAddr.port を参照)

sFileName: 削除するファイル名

pWriteBuff: 書き込み用のバッファアドレス

cbWriteLen: 書き込み用のバイトカウント

dwPassCode: ファイルアクセス用のパスワード

tTimeout: コマンドが実行されるときに、超過してはならないタイムアウト時間。

VAR_OUTPUT

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

bBusy: ファンクションブロックがアクティブな間はTRUEです。

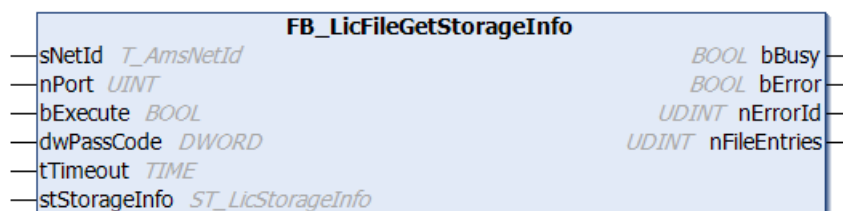
bError: コマンド実行中にエラーが発生するとTRUEとなります。

nErrorId: bError出力がセットされると、ADSエラー番号を返します。

要件

開発環境	ターゲットシステム	必要なPLCライブラリ
TwinCAT v3.1.4022	PCまたはCX (x64、x86)	Tc2_Uilities、3.3.26.0以降

3.39 FB_LicFileGetStorageInfo



このファンクションブロックは、ライセンス dongle の StorageInfo とファイルディレクトリを読み込みします。

StorageInfo には、ストレージメディアの管理情報 (容量、空きバイト数、ファイルの数など) と個々のファイルエントリ (ファイルの名前、サイズ、アトリビュートなど) の配列が含まれています。

VAR_INPUT

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

sNetId: ライセンス dongle の AmsNetId (AMS ネットワーク ID) (T_AmsNetId)

- ・ USB Dongle: TwinCAT コンピュータの AmsNetId。ローカルコンピュータで実行する場合、空の文字列を入力できます。
- ・ EL6070: EtherCAT マスタの AmsNetId (EL6070 の InfoData の AdsAddr.netId を参照)

nPort: ライセンス dongle の AMS ポート

- ・ USB: ESB デバイスの ADS ポート (USB Dongle の [ESB Device] タブの ADS ポートを参照、デフォルトは 16#7100)

- ・ EL6070: EtherCATターミナルのADSポート (EL6070のInfoDataのAdsAddr.portを参照)

bExecute: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

dwPassCode: ファイルアクセス用のパスコード (特に保護されたファイル用のみ)

tTimeout: コマンドが実行されるときに、超過してはならないタイムアウト時間。

VAR_OUTPUT

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

bBusy: ファンクションブロックがアクティブな間はTRUEです。

bError: コマンド実行中にエラーが発生するとTRUEとなります。

nErrorId: bError出力がセットされると、ADSエラー番号を返します。

nFileEntries: ライセンスドングル上のファイル数

VAR_IN_OUT

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

stStorageInfo: ライセンスドングルのStorageInfo (ST_LicStorageInfo)

要件

開発環境	ターゲットシステム	必要なPLCライブラリ
TwinCAT v3.1.4022	PCまたはCX (x64, x86)	Tc2_Utilities、3.3.26.0以降

3.40 FB_LicFileRead



このファンクションブロックは、bExecuteでの立ち上がりエッジによってライセンスドングルからバッファ (pDestBuffおよびcbReadLen)へファイルを読み込みします。このバッファは十分に大きくなければいけません。バッファが十分でない場合、ファイルの最初の部分しか読み込みできません。

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: ライセンス dongle の AmsNetId (AMS ネットワーク ID) (T_AmsNetId)

- ・ USB Dongle: TwinCAT コンピュータの AmsNetId。ローカルコンピュータで実行する場合、空の文字列を入力できます。
- ・ EL6070: EtherCAT マスタの AmsNetId (EL6070 の InfoData の AdsAddr.netId を参照)

nPort: ライセンス dongle の AMS ポート

- ・ USB: ESB デバイスの ADS ポート (USB Dongle の [ESB Device] タブの ADS ポートを参照、デフォルトは 16#7100)
- ・ EL6070: EtherCAT ターミナルの ADS ポート (EL6070 の InfoData の AdsAddr.port を参照)

sFileName: 読み込むファイル名

pDestBuff: 読み込むバッファアドレス

cbReadLen: 読み込むバイトカウント

bExecute: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

dwPassCode: ファイルアクセス用のパスワード

tTimeout: コマンドが実行される時に、超過してはならないタイムアウト時間。

VAR_OUTPUT

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

bBusy: ファンクションブロックがアクティブな間は TRUE です。

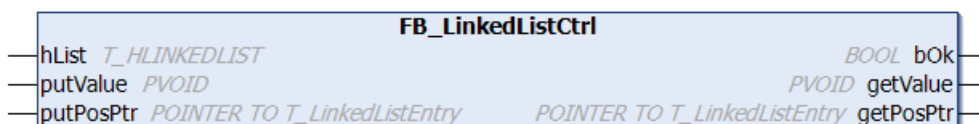
bError: コマンド実行中にエラーが発生すると TRUE となります。

nErrorId: bError 出力がセットされると、ADS エラー番号を返します。

要件

開発環境	ターゲットシステム	必要な PLC ライブラリ
TwinCAT v3.1.4022	PG または CX (x64, x86)	Tc2_Utilities、3.3.26.0 以降

3.41 FB_LinkedListCtrl



ファンクションブロック FB_LinkedListCtrl を使用して、PLC プロジェクトにリンクされたリストを実装できます。ダブルリンクされたリストが作成されます。リンクされたリストにより、(ノードとして既知の) 値を保存できます。後ろから前、またはその逆に繰り返すことが可能です。ノードはすぐに追加、または削除できます。

実行時にノードの最大数を変更することはできません。コンパイル前に指定する必要があります。タイプ T_LinkedListEntry の配列は、「ノードプール」として使用されます。ノードの追加/削除/検出は、アクションのコールで制御されます。ファンクションブロックには、次のタスクがあります。

- ・ **A_AddHeadValue** (リストの一番上に値 *putValue* をもつ新しいノードを追加します。同じ値は複数回追加できます。成功した場合、*getPosPtr* がアドレスを返し、一方で *getValue* が新しいノードの値を返します。)
- ・ **A_AddTailValue** (値 *putValue* をもつ新しいノードをリストの最後に追加します。同じ値は複数回追加できます。成功した場合、*getPosPtr* がアドレスを返し、一方で *getValue* が新しいノードの値を返します。)
- ・ **A_FindNext** (値が *putValue* と同じ次のノード (*putPosPtr* に相対的) を検索します。成功した場合、*getPosPtr* がアドレスを返し、一方で *getValue* がノードの値を返します。)
- ・ **A_FindPrev** (値が *putValue* と同じ以前のノード (*putPosPtr* と相対的に) と同じ以前のノードを検索します。成功した場合、*getPosPtr* がアドレスを返し、一方で *getValue* がノードの値を返します。)
- ・ **A_GetNext** (次のノードへナビゲートします (*putPosPtr* に相対的)。アドレス *putPosPtr* は、前のノードを指している必要があります。 *putValue* の値は使用されていません。)
- ・ **A_GetPrev** (*A_GetNext* と反対方向で以前のノード (*putPosPtr* に相対的) にナビゲートします。アドレス *putPosPtr* は、前のノードを指している必要があります。 *putValue* の値は使用されていません。)
- ・ **A_GetHead** (開始ノードを読み込みします。成功した場合、*getPosPtr* はノードのアドレスを返し、一方、*getValue* は関連した値を返します。 *putValue* と *putPosPtr* の値は使用されていません。)
- ・ **A_GetTail** (最後のノードを読み込みします。成功した場合、*getPosPtr* はノードのアドレスを返し、一方、*getValue* は関連した値を返します。 *putValue* と *putPosPtr* の値は使用されていません。)
- ・ **A_RemoveHeadValue** (リストの一番上からノードを削除します。成功した場合、*getPosPtr* がアドレスを返し、一方で *getValue* がノードの値を返します。 *putValue* と *putPosPtr* の値は使用されていません。)
- ・ **A_RemoveTailValue** (リストの最後からノードを削除します。成功した場合、*getPosPtr* がアドレスを返し、一方で *getValue* がノードの値を返します。 *putValue* と *putPosPtr* の値は使用されていません。)
- ・ **A_RemoveValueAtPosPtr** (アドレス *putPosPtr* をもつノードを検索して削除します。成功した場合、*getPosPtr* がアドレスを返し、一方で *getValue* がノードの値を返します。 *putValue* の値は使用されていません。)
- ・ **A_GetIndexAtPosPtr** (アドレス *putPosPtr* のノードの配列インデックス(「ノードプール」から)を返します。成功した場合、*getValue* はヌルベースの配列インデックスを返します。 *putValue* の値は使用されません。 *getValue* の値はノードインデックスで、ノードの値ではないことに注意してください。)
- ・ **A_SetValueAtPosPtr** (アドレス *putPosPtr* で *putValue* をもつノードの値を更新/セットします。成功した場合、*getPosPtr* がアドレスを返し、一方で *getValue* がノードの値を返します。)
- ・ **A_Reset** (すべてのリストエレメントを削除し、リストをリセットします。)

VAR_IN_OUT

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

hList: リンクされたリストハンドル ([T_HLINKEDLIST](#) [[▶ 304](#)])。ハンドルは使用する前に、ファンクション [F_CreateLinkedListHnd](#) [[▶ 246](#)] で初期化する必要があります。ハンドル変数の関連したインスタンスを作成し、各リンクされたリスト用に初期化する必要があります。

VAR_INPUT

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

putValue: 値/データエレメント(入力パラメータ、32/64ビット、符号なし数またはポインタ)。

putPosPtr: ノードエレメントのアドレス(入力パラメータ、[T_LinkedListEntry](#) [[▶ 305](#)])。

VAR_OUTPUT

```

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

```

bOk: 最後にコールされたアクションの結果。新しいノードエレメントをリストに追加、リストから削除、リストで検出できた場合にTRUEを返します。検索対象のノードエレメントを検出できなかった、リストが空、またはオーバーフローした(空のノードエレメントがもうない)場合、FALSEを返します。

getValue: 値/データエレメント(出力パラメータ、32/64ビット、符号なし数またはポインタ)。

getPosPtr: ノードエレメントのアドレス(出力パラメータ、[T_LinkedListEntry](#) [[▶ 305](#)])。

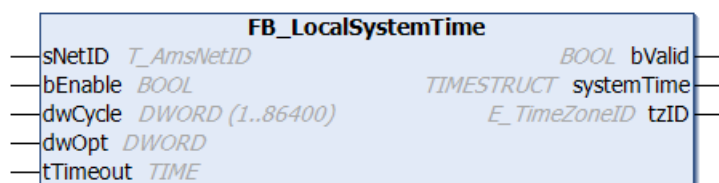
例:

参照: [例: リンクされたリスト\(FB_LinkedListCtrl\)](#) [[▶ 327](#)]。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.42 FB_LocalSystemTime



アプリケーションによっては、ローカルWindowsシステム時間をSNTPタイムサーバまたは電波時計を使用して同期しています。多くの場合、ローカルWindowsシステム時間はPLCで使用されます(たとえば、HMIへのタイムスタンプのログメッセージの形式で)。ローカルWindowsシステム時間は、タスクバーに表示されます。そのようなアプリケーションでは、FB_LocalSystemTimeファンクションブロックは有用です。

このファンクションブロックは、内部的に次のファンクションブロックの機能を結合します。[RTC_EX2](#) [[▶ 131](#)]、[NT_GetTime](#) [[▶ 113](#)]、[FB_GetTimeZoneInformation](#) [[▶ 69](#)]および[NT_SetTimeToRTCTime](#) [[▶ 116](#)]などです。たとえば、RTC_EX2ブロックは、ログの出力用にタイムスタンプを生成するのに使用できます。ただし、このブロックはその時間がローカルWindowsシステム時間と同期していないという欠点があり、周期的にNT_GetTimeファンクションブロック経由で再度、同期させる必要があります(マニュアルのRTCブロック例を参照)。内部時間(systemTime出力)の周期同期は、既にファンクションブロックに実装されています。サイクルタイムは、dwCycle入力によって設定できます。また、ファンクションブロックはタイムゾーン情報を提供します(夏時間/冬時間)。

FB_LocalSystemTimeファンクションブロックは、周期的にコールする必要があります(たとえば、1秒ごと、またはPLCサイクルごと)。これは、同期の間の時間を計算できるようにするために必要です。

● ジッタ

i ローカルWindowsシステム時間は、非周期サービス(ADS機能ブロック)を用いて読み込まれます。システム特性のために、ADSコマンドのランタイムは指定/推定できません。コマンドランタイムの差は、OS、同期間隔、およびPLCサイクルタイムに応じて、systemTime出力でのタイムジッタを引き起こすことがあります。このため、ブロックによって提供される時間は、たとえばビルディングオートメーションアプリケーションには十分である傾向がありますが、より高い精度を必要とする計測タスクに限られた範囲でのみ適しています。

夏時間と標準時間間の切り替え

ファンクションブロックは、夏時間から標準時間、またはその逆の切り替え時には、正確にその時間にコールすることができません。複雑な計算を避けるために、以下の実装が選択されました(例で説明)。

当社の例では、ファンクションブロックは自己の時間とローカルWindowsシステム時間(灰色)を60秒ごとに同期させます。

PLCアプリケーションは、ファンクションブロックの時間を必要とし、読み込みします(30秒ごとなど、青)。当社の例では、夏時間と標準時間間の切り替えは、15秒の遅延で検出されます。この挙動は、多くのアプリケーションでは問題がないはずで

標準時間から夏時間への切り替え

- ...
- 30-03-2008-01:58:10、tzID = 標準時間
- 30-03-2008-01:58:15、内部の同期後
- 30-03-2008-01:58:40、tzID = 標準時間
- 30-03-2008-01:59:10、tzID = 標準時間
- 30-03-2008-01:59:15、内部の同期後
- 30-03-2008-01:59:40、tzID = 標準時間
- 30-03-2008-02:00:00、OSが時間を2 amから3 amに変更
- 30-03-2008-02:00:10、tzID = 標準時間 (静止)
- 30-03-2008-03:00:15、内部同期後、それに続くtzID = 夏時間
- 30-03-2008-03:00:40、tzID = 夏時間
- 30-03-2008-03:01:10、tzID = 夏時間
- 30-03-2008-03:01:15、内部の同期後
- 30-03-2008-03:01:40、tzID = 夏時間
- ...

夏時間から標準時間への切り替え

- ...
- 26-10-2008-02:58:10、tzID = 夏時間
- 26-10-2008-02:58:15、内部の同期後
- 26-10-2008-02:58:40、tzID = 夏時間
- 26-10-2008-02:59:10、tzID = 夏時間
- 26-10-2008-02:59:15、内部の同期後
- 26-10-2008-02:59:40、tzID = 夏時間
- 26-10-2008-03:00:00、OSが時間を3 amから2 amに変更
- 26-10-2008-03:00:10、tzID = 夏時間(静止)
- 26-10-2008-02:00:15、内部の同期後、それに続くtzID = 標準時間
- 26-10-2008-02:00:40、tzID = 標準時間
- 26-10-2008-02:01:10、tzID = 標準時間
- 26-10-2008-02:01:15、内部の同期後
- 26-10-2008-02:01:40、tzID = 標準時間
- ...

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: その時間が同期のために使用するTwinCATコンピュータのネットワークアドレスの文字列 (T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

bEnable: この入力での立ち上がりで、ローカルWindowsシステム時間の内部時間の即時同期をトリガします。出力**bValid**は、同期が完了するまでFALSEがセットされた状態が継続します。最初の立ち上がりが、周期同期を有効にします。その後、周期同期が自動的に実行されます。多くの場合、アプリケーションはこの入力を一度、TRUEにセットしなければいけません。

dwCycle: ファンクションブロックが自己の時間を再同期させる間のサイクルタイム(秒単位)。周期同期は、**bEnable**入力での最初の立ち上がりの後に有効になります。デフォルト: 同期は5秒ごとです。

dwOpt: 追加のオプションパラメータ。以下のパラメータが現在、使用可能です。

- ・ビット0: これがセットされていると、Windowsシステム時間は追加でハードウェアクロックと追加で周期的に同期します(RTC) (ファンクションNT_SetTimeToRTCTime)に対応して)。デフォルト: 有効になっています。
このオプションは、Windows CEシステムとは無関係です。

tTimeout: 内部ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```

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

```

bValid: この出力がFALSEの場合、*systemTime*出力での時間は無効です。TRUE (すなわち、最低1度はローカルWindows時間で同期されています)の場合、時間は有効です。

systemTime: ローカルWindowsシステム時間 (TIMESTRUCT [▶ 306])。

tzID: タイムゾーン情報(夏時間、冬時間) (E_TimeZoneID [▶ 288])。

例:

例では、FB_LocalSystemTimeファンクションブロックはプログラムのスタートアップで有効になります (**bEnable**入力の立ち上がり)。時間が同期すると (**bValid** = TRUE)、PLCはメッセージをTwinCATシステムのLogviewに500 msごとに書き込みます。内部同期は秒ごとに実行されます。

```

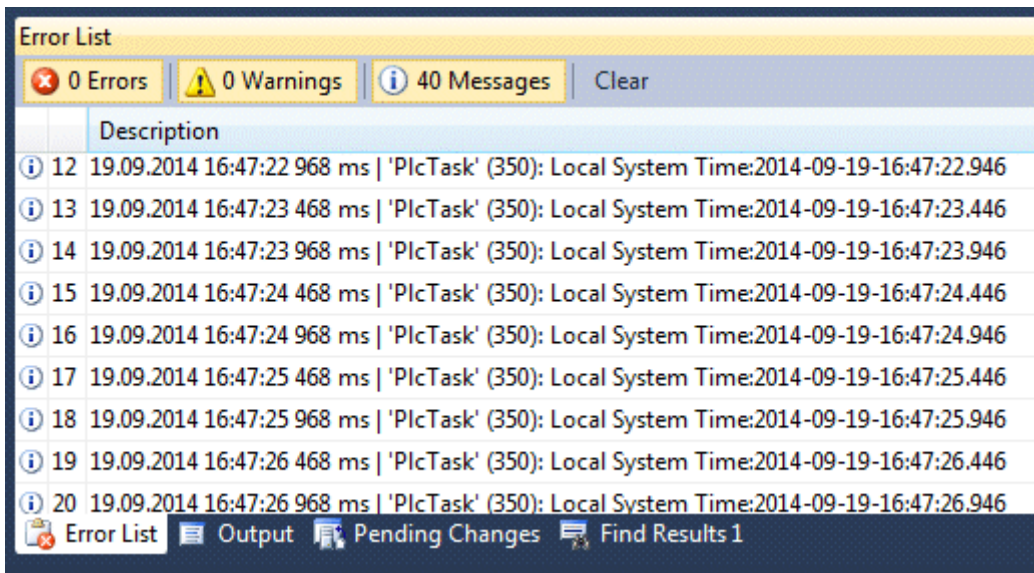
PROGRAM MAIN
VAR
  fbTime      : FB_LocalSystemTime := ( bEnable := TRUE, dwCycle := 1 );
  logTimer    : TON := ( IN := TRUE, PT := T#500ms );
END_VAR

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

```

書き込まれるメッセージされるメッセージは、TwinCAT Logviewで表示できます。



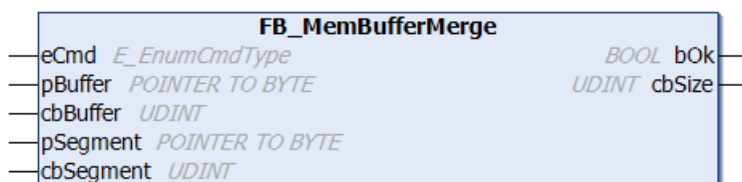
時間とタイムゾーンのその他のファンクションとファンクションブロック

- ・ [FB_TzSpecificLocalTimeToSystemTime \[▶_108\]](#)
- ・ [FB_TzSpecificLocalTimeToFileTime \[▶_106\]](#)
- ・ [FB_SystemTimeToTzSpecificLocalTime \[▶_104\]](#)
- ・ [FB_FileTimeToTzSpecificLocalTime \[▶_52\]](#)
- ・ [FB_GetTimeZoneInformation \[▶_69\]](#)
- ・ [FB_SetTimeZoneInformation \[▶_101\]](#)
- ・ [NT_SetLocalTime \[▶_115\]](#)
- ・ [NT_GetTime \[▶_113\]](#)
- ・ [NT_SetTimeToRTCTime \[▶_116\]](#)
- ・ [F_TranslateFileTimeBias \[▶_144\]](#)

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.43 FB_MemBufferMerge



このファンクションブロックは、大きいデータセグメントを構築するために個々の小さなデータセグメントを追加します。ターゲットバッファは、入力パラメータとしてブロックへ送信されなければいけません。追加されるセグメントが残っている空のバッファサイズを超過する場合、データバイトはさらに追加されません。

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: 列挙型ブロックの制御パラメータ (E_EnumCmdType [▶ 283])。eEnumCmd_Firstは最初のセグメントを追加し、eEnumCmd_Nextは次のセグメントを追加します。その他のパラメータは使用されません。

pBuffer: ターゲットバッファ変数のアドレス (ポインタ) アドレスは、ADR演算子で判定できます。

cbBuffer: ターゲットバッファ変数の最大使用可能サイズ (バイト単位)。サイズは、SIZEOF演算子で判定できます。

pSegment: 追加される次のデータセグメントのアドレス (ポインタ) (ゼロの場合があります)。また、アドレスはADR演算子で判定できます。

cbSegment: 追加される次のデータセグメントのサイズ (ゼロの場合があります)。また、サイズはSIZEOF演算子で判定できます。

VAR_OUTPUT

```

VAR_OUTPUT
  bOk      : BOOL;
  cbSize   : UDINT;
END_VAR

```

bOk: TRUE = 成功、FALSE = バッファオーバーフロー、または欠陥のある入力パラメータ。

cbSize: 現在のバッファ入カステータス (バッファのデータバイト数)。

例:

以下の例では、小さなデータセグメントが追加された後で大きなデータセグメントが16進数文字列に変換されます (テスト用)。

```

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
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.44 FB_MemBufferSplit



このファンクションブロックは、メモリ領域(データバッファ)を必要に応じて特定の最大長のいくつかの小さなセグメントに分割します。ファンクションブロックは、最後のセグメント長が必要な大きさよりも小さい場合、より小さい部分セグメントを返します。

VAR_INPUT

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

eCmd: ファンクションブロックの制御パラメータ (E_EnumCmdType [▶ 283])。eEnumCmd_Firstは最初のセグメントを返し、eEnumCmd_Nextは次のセグメントを返します。その他のパラメータは使用されません。

pBuffer: 分割されるデータバッファのアドレス(ポインタ)。アドレスは、ADR演算子で判定できます。

cbBuffer: 分割されるデータバッファ長。長さは、sizeof演算子で判定できます。

cbSize: データバッファが分割される最大セグメントサイズ。

VAR_OUTPUT

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

bOk: TRUE = 成功、FALSE = エラー、無効なパラメータ値または、使用可能なセグメントがもうありません。

pSegment: 次のデータセグメントへのアドレス(ポインタ)。

cbSegment: 次のデータセグメント長(バイト)。

bEOS: セグメントの終わり。TRUE = 最後のセグメント。FALSE = さらにセグメントが続きます。

例:

以下の例では、buffer変数は5バイトのセグメントに分割されます。返されたセグメントは、テストのために16進数文字列に変換されます。

```

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;

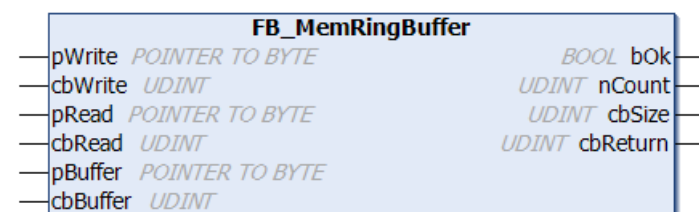
  REPEAT
    fbSplit( pBuffer := ADR(buffer), cbBuffer := SIZEOF(buffer), cbSize := 5 );
    IF fbSplit.bOk THEN
      sHex := DATA_TO_HEXSTR( pData := fbSplit.pSegment, cbData := fbSplit.cbSegment, FALSE );
      fbSplit.eCmd := eEnumCmd_Next;
    END_IF
  UNTIL NOT fbSplit.bOk
  END_REPEAT
END_IF

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.45 FB_MemRingBuffer



ファンクションブロックFB_MemRingBufferを使用して、異なる長さをもつデータセットをリングバッファに書き込んだり、リングバッファから以前に書き込まれたデータセットを読み込んだりすることができます。書き込まれたデータセットは、FIFO原理(すなわち、最も古いエントリが最も早く読み込まれる)に基づいて以前にリングバッファファイルに書き込まれているのと同じ順序で読み込まれます。バッファメモリは、*pBuffer/cbBuffer*入力変数によってファンクションブロックで使用可能になります。データセット書き込み/読み込みは、アクションのコールで制御されます。ファンクションブロックには、次のタスクがあります。

- ・ **A_AddTail** (新しいデータセットをリングバッファに書き込みます。)
- ・ **A_GetHead** (リングバッファの最も古いデータセットを読み込みますが、削除しません。)
- ・ **A_RemoveHead** (リングバッファから最も古いデータセットを読み込みし、削除します。)
- ・ **A_reset** (リングバッファのすべてのデータセットを削除します。)

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: PLC変数のアドレス、または書き込まれる値データを含むバッファ変数のアドレス。アドレスは、ADR演算子で判定できます。*cbWrite*データバイトを取得できるバッファ変数のサイズは、プログラマが決定する必要があります。

cbWrite: 書き込まれる値データバイト数。(文字列変数の場合には、これは最後のヌルを含みます)。

pRead: 読み込まれる値データがコピーされるPLC変数、またはバッファ変数のアドレス。アドレスは、ADR演算子で判定できます。*cbRead*データバイトを受け入れることができるバッファ変数のサイズは、プログラマが決定する必要があります。バッファ変数のサイズ(バイト単位)は、読み込まれるデータレコードのサイズ以上でなければいけません。

cbRead: 読み込まれる値データバイト数。バッファサイズが小さすぎる場合、データはコピーされません。ファンクションブロックはバッファアンダーフローエラー(*bOk* = FALSE)を報告し、読み込まれる次のデータレコードに必要なバッファサイズを*cbReturn*出力で返します。

pBuffer: ファンクションブロックによってバッファメモリとして使用するPLC変数のアドレス(例: RRAY[...] OF BYTES)。アドレスは、ADR演算子で判定できます。

cbBuffer: バッファメモリとして使用するPLC変数の最大バイトサイズ。サイズは、SIZEOF演算子で判定できます。

VAR_OUTPUT

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

bOk: 新しいデータセットが正常に追加されたり、削除されたりした場合、TRUEを返します。バッファオーバーフローの場合、またはバッファでエントリがこれ以上利用できない場合に、FALSEを返します。

nCount: キューのデータセットの現在数を返します。

cbSize: バッファで割り当てられたデータバイトの現在数を返します。割り当てられたデータバイト数は、常に書き込まれた値された値データの実数の数よりも大きくなります。各データセットは追加情報で補完されます。そのため、後で位置を特定可能です。

cbReturn: 正常に読み込んだ値データバイト数。読み込みバッファのアンダーフローエラーが発生した場合、この出力は読み込みバッファサイズをバイト単位で返します。この場合、*cbRead* 長は小さすぎます。

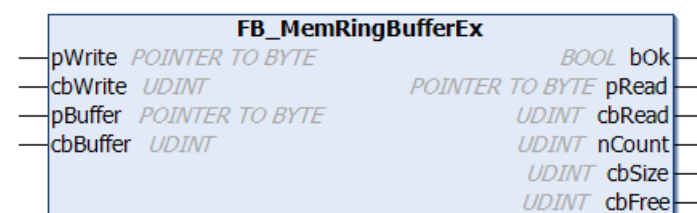
例:

参照: [例: メモリリングFiFo \(FB_MemRingBuffer\)](#)。 [▶ 319]

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.46 FB_MemRingBufferEx



ファンクションブロックFB_MemRingBufferEXを使用して、異なる長さをもつデータセットをリングバッファに書き込んだり、リングバッファから以前に書き込まれたデータセットを読み込んだりすることができます。書き込まれたデータセットは、FIFO原理(すなわち、最も古いエントリが最も早く読み込まれる)に基づいて以前にリングバッファファイルに書き込まれているのと同じ順序で読み込まれます。バッファメモリは、*pBuffer/cbBuffer*入力変数によってファンクションブロックで使用可能になります。データセット書き込み/読み込みは、アクションのコールで制御されます。

このファンクションブロックの機能は、[FB_MemRingBuffer](#) [▶ 87] ファンクションブロックの機能と同様です。データセットの読み込み中に、FB_MemRingBufferはデータを外部のバッファ変数にコピーします。FB_MemRingBufferExは、データセット用の参照のみを提供します(アドレスポインタ/アドレス長)。アプリケーションは、その後の処理のためにデータ自体をコピーしなければいけません。

ファンクションブロックには、次のタスクがあります。

- ・ **A_AddTail** (新しいデータセットをリングバッファに書き込みます。)
- ・ **A_GetHead** (参照を返します。リングバッファの最も古いデータセットのアドレスポインタ/アドレス長を返しますが、削除しません。)
- ・ **A_FreeHead** (リングバッファから最も古いデータセットを読み込みし、削除します。返されたアドレスポインタ/アドレス長は、ゼロです。新しいデータセット用に空のメモリセグメントは解放されません。)
- ・ **A_reset** (リングバッファのすべてのデータセットを削除します。)
- ・ **A_GetFreeSize** (バッファの最大の空のメモリセグメントのバイトサイズを返します。)

VAR_INPUT

```
VAR_INPUT
  pWrite   : POINTER TO BYTE;
  cbWrite  : UDINT;
  pBuffer  : POINTER TO BYTE;
  cbBuffer : UDINT;
END_VAR
```

pWrite: PLC変数のアドレス、または書き込まれる値データを含むバッファ変数のアドレス。アドレスは、ADR演算子で判定できます。*cbWrite*データバイトを取得できるバッファ変数のサイズは、プログラマが決定する必要があります。

cbWrite: 書き込まれる値データバイト数。(文字列変数の場合には、これは最後のヌルを含みます)。サイズは、SIZEOF演算子で判定できます。

pBuffer: ファンクションブロックによってバッファメモリとして使用するPLC変数のアドレス(例: RRAY[...] OF BYTES)。アドレスは、ADR演算子で判定できます。

cbBuffer: バッファメモリとして使用するPLC変数の最大バイトサイズ。サイズは、SIZEOF演算子で判定できます。

VAR_OUTPUT

```
VAR_OUTPUT
  bOk      : BOOL;
  pRead    : POINTER TO BYTE;
  cbRead   : UDINT;
  nCount   : UDINT;
  cbSize   : UDINT;
  cbFree   : UDINT;
END_VAR
```

bOk: 新しいデータセットが正常に追加されたり、削除されたりした場合、TRUEを返します。バッファオーバーフローの場合、またはバッファでエントリがこれ以上利用できない場合に、FALSEを返します。

pRead: アクションA_GetHeadのコール後、成功した場合(bOk=TRUE)、この変数はリングバッファの最も古いデータセット用の参照(アドレスポインタ)を返します。リングバッファでこれ以上データセットが利用できない場合、ゼロを返します。

cbRead: アクションのコール後、成功した場合 (`bOk=TRUE`)、`A_GetHead`変数はリングバッファの最も古いデータセットの長さを返します。リングバッファでこれ以上データセットが利用できない場合、ゼロを返します。

nCount: キューのデータセットの現在数を返します。

cbSize: バッファで割り当てられたデータバイトの現在数を返します。割り当てられたデータバイト数は、常に書き込まれた値データの数よりも大きくなります。各データセットは追加情報で補完されます。そのため、後で位置を特定可能です。

cbFree: アクションのコール後、`A_GetFreeSize`は、バッファの中で最大の空のメモリセグメントのバイトサイズを返します。ファンクションブロックがデータセットの参照を返すので、データセットはバッファメモリの連続したアドレスを使用しなければいけません。これは、自動的にバッファの終わりでのセグメンテーションの原因となります。新しいデータセットがバッファの終わりの空のセグメントよりも大きい場合、このメモリは使用できません。

例:

参照: [例: メモリリングFiFo \(FB_MemRingBufferEx\) \[▶ 321\]](#)。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.47 FB_MemStackBuffer



ファンクションブロック `FB_MemStackBuffer` を使用して、異なる長さをもつデータセットをバッファに書き込んだり、以前に書き込まれたデータセットをバッファから読み込んだりすることができます。データセットはLIFO原理(最後に入れたものを最初に出す)に基づいて読み込まれます。すなわち、バッファに書き込まれたされたのと逆の順序で読み込まれます。つまり、最新のエンタリは、最初に読み込まれます。バッファメモリは、`pBuffer`および`cbBuffer`入力変数によってファンクションブロックで使用可能になります。データセット書き込み/読み込みは、アクションのコールで制御されます。ファンクションブロックには、次のタスクがあります。

- ・ `A_Push()`: 新しいデータセットをバッファに書き込みます。
- ・ `A_Top()`: バッファから最後に追加された/最新のデータセットを読み込みしますが、削除しません。
- ・ `A_Pop()`: バッファから最後に追加された/最新のデータセットを読み込みし、削除します。
- ・ `A_Reset()`: バッファからすべてのデータセットを削除します。

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: PLC変数のアドレス、または書き込まれる値データを含むバッファ変数のアドレス。アドレスは、ADR演算子で判定できます。*cbWrite*データバイトを取得できるバッファ変数のサイズは、プログラマが決定する必要があります。

cbWrite: 書き込まれる値データバイト数。(文字列変数の場合には、これは最後のヌルを含みます)。

pRead: 読み込まれる値データがコピーされるPLC変数、またはバッファ変数のアドレス。アドレスは、ADR演算子で判定できます。*cbRead*データバイトを受け入れることができるバッファ変数のサイズは、プログラマが決定する必要があります。バッファ変数のサイズ(バイト単位)は、読み込まれるデータレコードのサイズ以上でなければいけません。

cbRead: 読み込まれる値データバイト数。バッファサイズが小さすぎる場合、データはコピーされません。ファンクションブロックはバッファアンダーフローエラー(*bOk* = FALSE)を報告し、読み込まれる次のデータレコードに必要なバッファサイズを*cbReturn*出力で返します。

pBuffer: ファンクションブロックによってバッファメモリとして使用するPLC変数のアドレス(例: RRAY[...] OF BYTES)。アドレスは、ADR演算子で判定できます。

cbBuffer: バッファメモリとして使用するPLC変数の最大バイトサイズ。サイズは、SIZEOF演算子で判定できます。

VAR_OUTPUT

```
VAR_OUTPUT
```

```
  bOk      : BOOL;
  nCount   : UDINT;
  cbSize   : UDINT;
  cbReturn : UDINT;
```

```
END_VAR
```

bOk: 新しいデータセットが正常に追加されたり、削除されたりした場合、TRUEを返します。バッファオーバーフローの場合、またはバッファでエントリがこれ以上利用できない場合に、FALSEを返します。

nCount: キューのデータセットの現在数を返します。

cbSize: バッファで割り当てられたデータバイトの現在数を返します。割り当てられたデータバイト数は、常に書き込まれた値データの実際の数よりも大きくなります。各データセットは追加情報で補完されます。そのため、後で位置を特定可能です。

cbReturn: 正常に読み込んだ値データバイト数。読み込みバッファのアンダーフローエラーが発生した場合、この出力は読み込みバッファサイズをバイト単位で返します。この場合、*cbRead* 長は小さすぎます。

例:

以下の例で、ファンクションブロックの簡単なアプリケーションを説明します。異なる長さの文字列がバッファリングされます。*bReset*の立ち上がりで、バッファをクリアします。*bAdd*がTRUEにセットされると、10個の新しい文字列をバッファに書き込みます。*bRemove*がTRUEの場合、最後に書き込まれた文字列はバッファから削除されます。*bGet*の立ち上がりで、最後に書き込まれた文字列が読み込まれますが、削除されません。

宣言部分:

```
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
```

プログラムコード:

```

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

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.48 FB_RegQueryValue



システムレジストリは階層構造化ツリーです。ツリーのノードは、キーとして参照されます。各キーは、サブキーとデータ値を含むことができます。ファンクションブロック「FB_RegQueryValue」を使用して、事前定義されたハンドルHKEY_LOCAL_MACHINEをもつブランチから個々のシステムレジストリ値を読み込みできま

す。成功した場合、*cbData*データバイトは、アドレス*pData*をもつバッファにコピーされます。ファンクションブロックを使用して、任意の値タイプ(REG_DWORD、REG_SZなど)、または無制限のバイト長(REG_BINARY)をもつバイナリデータを読み込みできます。

コメント:

文字列*sSubKey*と*sValueName* は、空でない可能性があります

● 64ビットOS用のHKEY_LOCAL_MACHINE¥SOFTWARE¥



64ビットWindowsOSでは、32ビットアプリケーションのすべてのレジストリのエントリは、KEY_LOCAL_MACHINE¥SOFTWARE¥の下ではなく、HKEY_LOCAL_MACHINE¥SOFTWARE¥WOW6432Node¥の下に保存されます。
ファンクションブロックFB_RegQueryValueおよびFB_RegSetValueは、SOFTWAREフォルダの下のレジストリエントリ が選択されている場合、すべての32ビットアプリケーションのように自動的にWOW6432Nodeフォルダの下で動作します。出力先の変更は、OSによって自動的に実行されます。

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: そのシステムレジストリが読み込まれるTwinCATコンピュータのネットワークアドレスをもつ文字列を入力できます(T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

sSubKey: サブキー名をもつ文字列(T_MaxString)。

sValName: 値名をもつ文字列(T_MaxString)。

cbData: 読み込まれる値データバイトの数。

pData: 値データがコピーされるデータバッファ/変数のアドレス。アドレスは、ADR演算子で判定できません。cbDataデータバイトを格納できるデータバッファのサイズは、プログラマが決定する必要があります。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットしません。

nErrId: *bError*出力のセット時にADSエラー番号 [▶ 342]、またはコマンド固有のエラーコード(テーブル)を返します。

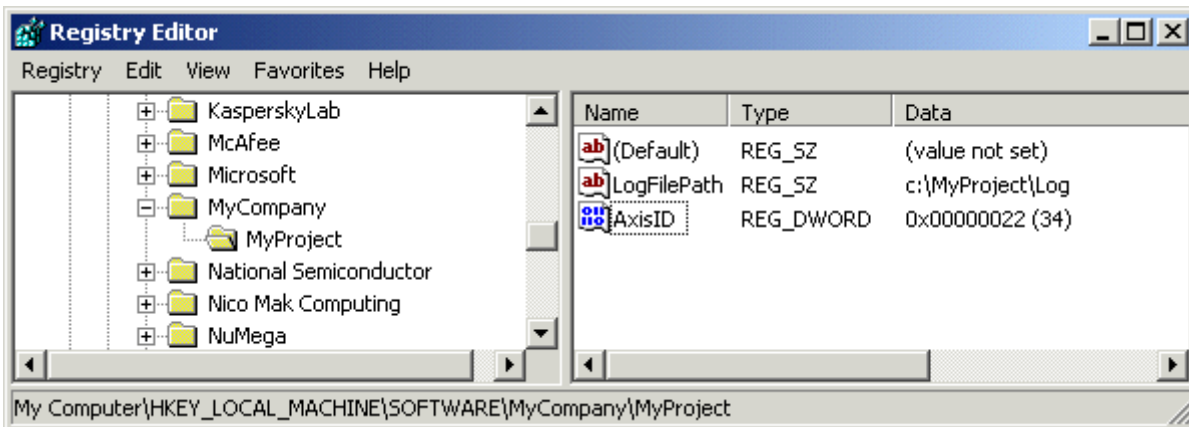
cbRead: 正常に読み込んだ値データバイト数。

エラーコード	エラー説明
0x00	エラーはありません

エラーコード	エラー説明
0x01	名前がsSubKeyのキーは開くことができず、検出できません。
0x02	名前が sValNameのキー値は開くことができず、検出できません。

例:

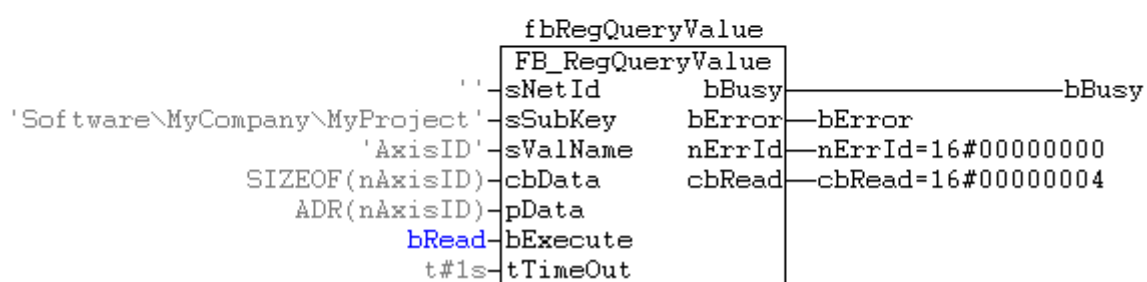
値AxisIdとLogFilePath は、システムレジストリから読み込まれます。



```
PROGRAM MAIN
VAR
    fbRegQueryValue : FB_RegQueryValue;
    bRead           : BOOL;
    bBusy          : BOOL;
    bError         : BOOL;
    nErrId         : UDINT;
    cbRead         : UDINT;
    sValData       : STRING;
    nAxisID        : DWORD;
END_VAR
```

Read REG_DWORD-Value:

```
fbRegQueryValue
bRead = TRUE
bBusy = FALSE
bError = FALSE
nErrId = 16#00000000
cbRead = 16#00000004
nAxisID = 16#00000022
```

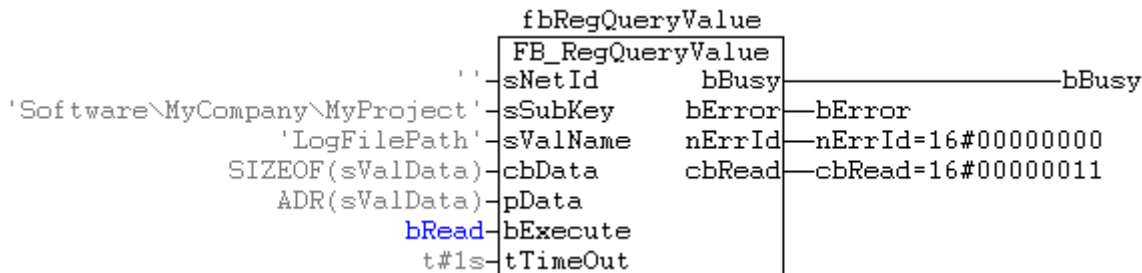


レジストリの値0x22は、PLC変数nAxisIdで読み込みましたしました。

Read REG_SZ-Value :

```

fbRegQueryValue
-----
bRead = TRUE
bBusy = FALSE
bError = FALSE
nErrId = 16#00000000
cbRead = 16#00000011
sValData = 'c:\MyProject\Log'
    
```



レジストリの文字列「c:\MyProject\Log」は、PLC変数sValDataで読み込みました。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.49 FB_RegSetValue



システムレジストリは階層構造化ツリーです。ツリーのノードは、キーとして参照されます。各キーは、サブキーとデータ値を含むことができます。

ファンクションブロック「FB_RegSetValue」を使用して、事前定義されたハンドルHKEY_LOCAL_MACHINEによるブランチの個々のキー値、または新しいキー名や値(サブキー+値)の書き込みや生成ができます。任意の数の値の型(REG_DWORD、REG_SZなど)や最大500 バイトのバイナリデータ(REG_BINARY)をシステムレジストリに書き込みできます。また、キー値が存在しない場合、自動的に作成されます。

コメント:

文字列sSubKeyとsValueName は、空でない可能性があります

● 64ビットOS用のHKEY_LOCAL_MACHINE\SOFTWARE

i 64ビットWindowsOSでは、32ビットアプリケーションのすべてのレジストリのエントリは、KEY_LOCAL_MACHINE\SOFTWAREの下ではなく、HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Nodeの下に保存されます。ファンクションブロックFB_RegQueryValueおよびFB_RegSetValueは、SOFTWAREフォルダの下のレジストリエントリが選択されている場合、すべての32ビットアプリケーションのように自動的にWOW6432Nodeフォルダの下で動作します。出力先の変更は、OSによって自動的に実行されます。

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: そのシステムレジストリが書き込まれるTwinCATコンピュータのネットワークアドレスをもつ文字列を入力できます (T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

sSubKey: サブキー名をもつ文字列 (T_MaxString)。

sValName: 値名をもつ文字列 (T_MaxString)。

eValType: REG_DWORDやREG_SZなどの書き込まれるレジストリデータのデータ型のフォーマット (E_RegValueType [▶ 286])。

cbData: 書き込まれる値データバイト数。(文字列変数の場合には、これは最後のヌルを含みます)。

pData: 値データを含むデータバッファ/PLC変数のアドレス。アドレスは、ADR演算子で判定できます。
cbDataデータバイトを読み込むことのできるデータバッファのサイズは、プログラマが決定する必要があります。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットしません。

nErrId: *bError*出力のセット時にADSエラー番号 [▶ 342]、またはコマンド固有のエラーコード(テーブル)を返します。

cbWrite: 正常に書き込まれた値データバイト数。

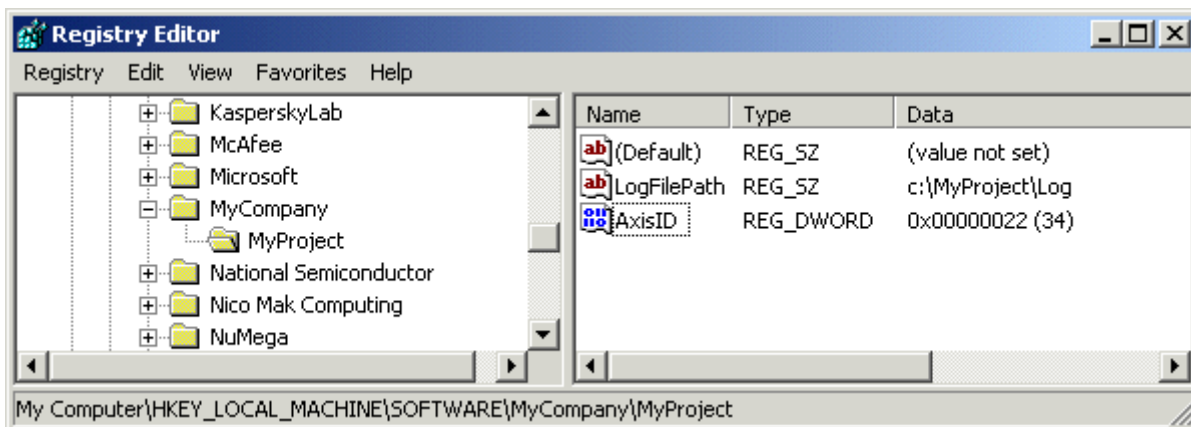
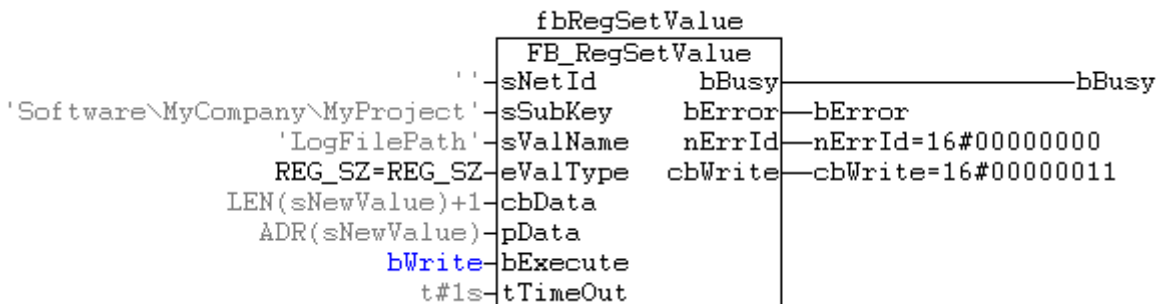
エラーコード	エラー説明
0x00	エラーはありません
0x01	名前が sSubKey のキーは開くことができず、検出できません。
0x02	名前が sValName のキー値は開くことができず、検出できません。

例:

事前定義されたハンドルHKEY_LOCAL_MACHINEがあるブランチでは、キー名が「LogFile \backslash Name」のサブキー「SOFTWARE \backslash MyCompany \backslash MyProject」、タイプREG_SZ、

および値「c: \backslash MyProject \backslash Log」が作成され、セットされます。

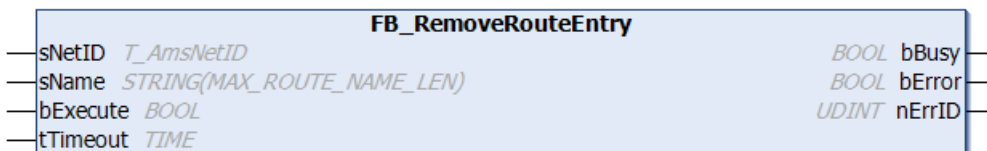

```
PROGRAM MAIN
VAR
    fbRegSetValue      : FB_RegSetValue;
    bBusy              : BOOL;
    bError             : BOOL;
    nErrId             : UDINT;
    cbWrite            : UDINT;
    bWrite             : BOOL;
    sNewValue          : STRING := 'c:\MyProject\Log';
END_VAR
```



要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3. 1. 0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3. 50 FB_RemoveRouteEntry



このファンクションブロックを使用して、AMSルータ接続のリストからTwinCATシステムへの既存の接続を削除できます(リモートルート)。

VAR_INPUT

```
VAR_INPUT
    sNetID      : T_AmsNetID;
    sName       : STRING(MAX_ROUTE_NAME_LEN);
```

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

sNetID: ここでは、AMSルータ接続を削除するTwinCATコンピュータのネットワークアドレスを文字列で指定できます(T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

sName: 削除する接続名です。最大文字列長は、定数によって制限されます(デフォルト: 31文字)。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

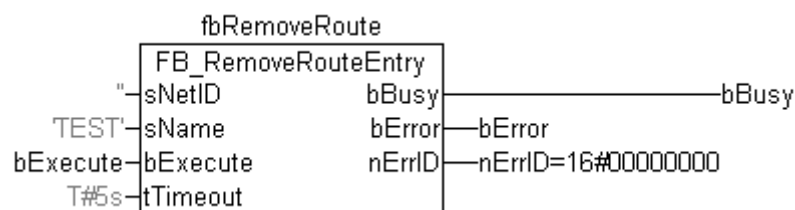
bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットします。

nErrId: *bError*出力がセットされると、ADSエラー番号 [▶ 342]を返します。

例:

名前が「TEST」の接続が、ローカルTwinCAT PCシステムのAMSルータ接続リストから削除されます。立ち上がりが**bExecute**変数で検出された場合、接続は削除されます。

```
PROGRAM P_TEST2
VAR
  fbRemoveRoute : FB_RemoveRouteEntry;
  bExecute      : BOOL;
  bBusy         : BOOL;
  bError        : BOOL;
  nErrID        : UDINT;
END_VAR
```



要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.51 FB_ScopeServerControl

[このマニュアルは準備段階のものであり、変更されることがあります]

FB_ScopeServerControl		
sNetId	T_AmsNetId	BOOL bBusy
eReqState	E_ScopeServerState	BOOL bDone
sConfigFile	STRING	BOOL bError
sSaveFile	STRING	UDINT nErrorId
tTimeout	TIME	

ファンクションブロック「FB_ScopeServerControl」により、PLCはTwinCATスコープ2で後で表示するためにデータを収集できます。

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: このパラメータを使用して、TwinCATターゲットシステムのネットワークアドレスを含む文字列を指定できます(T_AmsNetId)。ローカルコンピュータの場合、この文字列は空のことがあります。

eReqState: リクエストされたスコープサーバステータス (E_ScopeServerState [[▶ 288](#)])。

sConfigFile: 構成ファイル名付きのフルパス (例: 「C:¥TwinCAT¥TwinCATScope2¥First.sv2」)。

sSaveFile: データファイル名付きのフルパス (例: 「C:¥TwinCAT¥TwinCATScope2¥First.svd」)。

tTimeout: 内部ADSコマンドの最大許容時間

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまでセットされた状態が継続します。

bDone: リクエストされたステータスが有効な場合、bDoneをセットします。

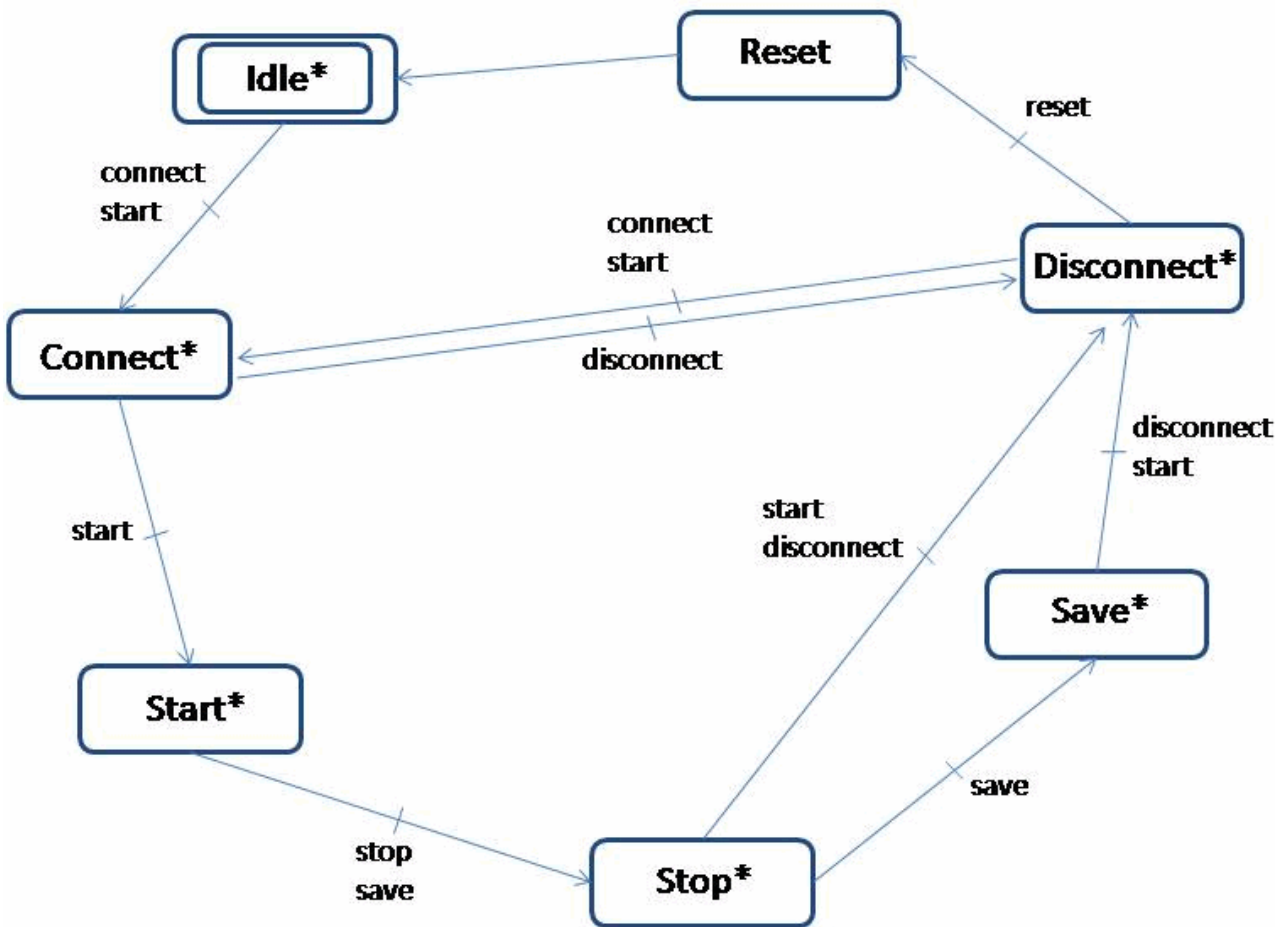
bError: コマンド送信中にエラーが発生した場合、bBusy出力がリセットされるとこの出力をセットしません。

nErrorId: エラー出力bErrorがセットされると、エラー番号を表示します。一般に、これは、ADSエラー番号 [[▶ 342](#)]、このライブラリの固有のエラーコード [[▶ 341](#)]です。

Notice

1つのターゲットシステムのみが、構成ファイル(*.sv2)に対して許可されます。

状態図:



*) 状態の変更をリセットすることは、いつでも可能です。

この状態図は、eReqStateの起こる可能性のある遷移を示します。他に状態の変更がある場合、リクエストされたbErrorをセットします。

タイプ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
);
  
```

例:

宣言部分:

```

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

```

FB_ScopeServerSampleの実装

```

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);

```

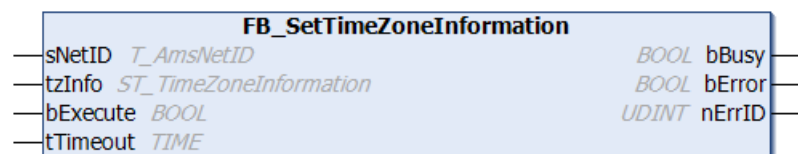
この例は、スコープサーバによる長期の記録を示します。

既存のコンフィグレーション(Test.sv2)は、このために読み込まれます。この例では、Test.sv2はリングバッファを実行するために保存されました。データロギングは、FB_ScopeServerControlが有効になるまで終了しません。内部のトリガイベント(エラーイベントなど)が発生すると、タイマーが起動し、10秒後にデータはTest.svdに保存されます。このようにして、データファイルはトリガイベントの前後の情報を格納します。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.52 FB_SetTimeZoneInformation



このファンクションブロックを使用して、OSのタイムゾーン設定を変更したり、セットしたりできます。

● 時間設定

新しいタイムゾーン設定がセットされた後で、OSはいくつかの時間設定を変更できます。
時間は、ファンクションブロック `NT_SetLocalTime` [▶ 115] でリセットできます。

VAR_INPUT

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

sNetID: そのタイムゾーン設定が変更されるTwinCATコンピュータのネットワークアドレスをもつ文字列を入力できます(T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

tzInfo: セットされる新しいタイムゾーン情報をもつ構造体(ST_TimeZoneInformation [▶ 300])。

bExecute: ブロックは、この入力での立ち上がりエッジによって有効になります。

tTimeout: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

bBusy: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

bError: コマンド送信中にエラーが発生した場合、*bBusy*出力がリセットされるとこの出力をセットしません。

nErrId: *bError*出力がセットされると、ADSエラー番号 [▶ 342] を返します。

サンプル:

ローカルTwinCAT PCシステムでのタイムゾーン: 「Western Europe Standard Time (西ヨーロッパ標準時間)」をセットします。サンプルとして、適切なパラメータ値をもつ定数WEST_EUROPE_TZIは、既にPLCライブラリで宣言されました。他のタイムゾーンを設定するには、ファンクションブロックの*tzInfo*入力を、対応する値に割り当てる必要があります(ST_TimeZoneInformation [▶ 300]構造体の説明を参照)。

```
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
```

宣言セクション:

```
PROGRAM MAIN
VAR
  fbGet      : FB_GetTimeZoneInformation;
  fbSet      : FB_SetTimeZoneInformation;
```

```

tzi_get : ST_TimeZoneInformation;
tzID    : E_TimeZoneID;
bGet    : BOOL := TRUE;
bSet    : BOOL := FALSE;

```

```
END_VAR
```

*bSet*変数の立ち上がりで、必要なタイムゾーン設定をセットします。検証のために、現在の設定は*bGet*変数の立ち上がりで読み込みできます。

```

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

```

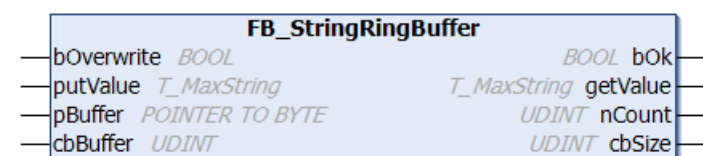
その他の時間とタイムゾーンファンクションおよびファンクションブロック:

- ・ [FB_TzSpecificLocalTimeToSystemTime](#) [[▶ 108](#)]
- ・ [FB_TzSpecificLocalTimeToFileTime](#) [[▶ 106](#)]
- ・ [FB_SystemTimeToTzSpecificLocalTime](#) [[▶ 104](#)]
- ・ [FB_FileTimeTimeToTzSpecificLocalTime](#) [[▶ 52](#)]
- ・ [FB_GetTimeZoneInformation](#) [[▶ 69](#)]
- ・ [NT_SetLocalTime](#) [[▶ 115](#)]
- ・ [NT_GetTime](#) [[▶ 113](#)]
- ・ [NT_SetTimeToRTCTime](#) [[▶ 116](#)]
- ・ [F_TranslateFileTimeBias](#) [[▶ 144](#)]
- ・ [FB_LocalSystemTime](#) [[▶ 81](#)]

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.53 FB_StringRingBuffer



ファンクションブロックFB_StringRingBufferを使用して、文字列変数をリングバッファに書き込んだり、リングバッファから以前に書き込んだ文字列変数を読み込んだりすることができます。書き込まれた文字列は、FIFO原理(すなわち、最も古いエントリが最も早く読み込まれる)に基づいて以前にリングバッファファイルに書き込まれているのと同じ順序で読み込まれます。バッファメモリは、*pBuffer/cbBuffer*入力変数によってファンクションブロックで使用可能になります。文字列の書き込み/読み込みはアクションのコールで制御されます。ファンクションブロックには、次のタスクがあります。

- ・ **A_AddTail** (新しい文字列をリングバッファに書き込みます。)

- ・ **A_GetHead** (リングバッファの最も古い文字列を読み込みますが、削除しません。)
- ・ **A_RemoveHead** (リングバッファから最も古い文字列を読み込みし、削除します。)
- ・ **A_Reset** (リングバッファからすべての文字列を削除します。)

VAR_INPUT

```
VAR_INPUT
  bOverwrite : BOOL;
  putValue   : T_MaxString := '';
  pBuffer    : POINTER TO BYTE;
  cbBuffer   : UDINT;
END_VAR
```

bOverwrite : TRUEの場合、最も古いエントリはバッファオーバーフローの際に上書きされます。FALSEの場合、バッファオーバーフローの際にエラーが報告されます (bOk = FALSE)。

putValue: リングバッファに書き込まれる文字列される文字列 (T_MaxString)。

pBuffer: ファンクションブロックによってバッファメモリとして使用するPLC変数のアドレス (例: RRAY[...] OF BYTES)。アドレスは、ADR演算子で判定できます。

cbBuffer: バッファメモリとして使用するPLC変数の最大バイトサイズ。サイズは、SIZEOF演算子で判定できます。

VAR_OUTPUT

```
VAR_OUTPUT
  bOk       : BOOL;
  getValue  : T_MaxString := '';
  nCount    : UDINT;
  cbSize    : UDINT;
END_VAR
```

bOk: 新しい文字列が正常に追加されたり、削除されたりした場合、TRUEを返します。バッファオーバーフローの場合、またはバッファでエントリがこれ以上利用できない場合に、FALSEを返します。

getValue: この出力は、リングバッファから読み込んだ最後の文字列を返します (T_MaxString)。

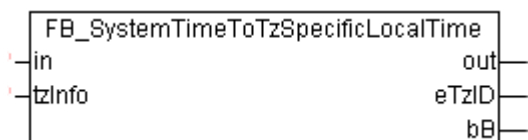
nCount: キューの文字列の現在数を返します。

cbSize: バッファで割り当てられたデータバイトの現在数を返します。割り当てられたデータバイト数は、常に書き込まれた値データの実際の数よりも大きくなります。各文字列は追加情報で補完されます。そのため、後で位置の特定が可能です。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.54 FB_SystemTimeToTzSpecificLocalTime



このファンクションブロックは、指定したタイムゾーン情報を考慮に入れてUTC時間 (構造体システム時間の形式) を現地時間 (構造体システム時間の形式) に変換します。ファンクションブロック **FB_FileTimeToTzSpecificLocalTime** [▶ 52]には同様の機能がありますが、異なる時間の形式 (ファイル時間の形式) を使用します。

ブロックは、**連続的なUTCタイムスタンプ情報の変換にのみ適しています**。ファンクションブロックはタイムゾーン情報を使用して、現地時間で必要な時間ステップ（夏時間/冬時間の切り替え）を計算します。UTC入力時間での時間ステップは、不正な変換になるため許可されていません。理由：ブロックは最後に変換された時間を内部的に保存し、そのため現地時間が変更されたときにUTC入力時間と保存された値からB時間（下記参照）を検出できます。

ブロックはアクションと関連があります：A_Reset()。このアクションがブロック出力をコールし、ローカルに保存された（最後に変換された）時間がゼロにリセットされます。

VAR_INPUT

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

in: 変換するUTC時間（構造体システム時間の形式）（[TIMESTRUCT \[▶ 306\]](#)）。

tzInfo: OSの現在のタイムゾーン情報をもつ構造体変数（[ST_TimeZoneInformation \[▶ 300\]](#)）。

VAR_OUTPUT

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

out: 変換された現地時間（構造体システム時間の形式）（[TIMESTRUCT \[▶ 306\]](#)）。

eTzID: 追加の夏時間/冬時間情報（[E_TimeZoneID \[▶ 288\]](#)）。

bB: TRUE → B時間（例：02:05:00 CET B）、FALSE → 他の時間（例：02:05:00 GEST A）。この出力は現地時間がジャンプバックするとセットされ、複製時間が経過するとリセットされます。

例:

```
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 );
```

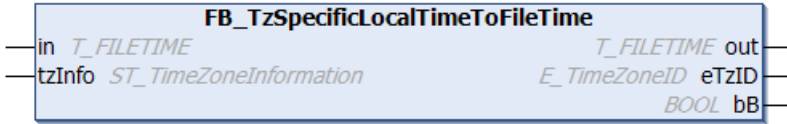
その他の時間とタイムゾーンファンクションおよびファンクションブロック:

- [FB_TzSpecificLocalTimeToSystemTime \[▶ 108\]](#)
- [FB_TzSpecificLocalTimeToFileTime \[▶ 106\]](#)
- [FB_FileTimeTimeToTzSpecificLocalTime \[▶ 52\]](#)
- [FB_GetTimeZoneInformation \[▶ 69\]](#)
- [FB_SetTimeZoneInformation \[▶ 101\]](#)
- [NT_SetLocalTime \[▶ 115\]](#)
- [NT_GetTime \[▶ 113\]](#)
- [NT_SetTimeToRTGTime \[▶ 116\]](#)
- [F_TranslateFileTimeBias \[▶ 144\]](#)
- [FB_LocalSystemTime \[▶ 81\]](#)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.55 FB_TzSpecificLocalTimeToFileTime

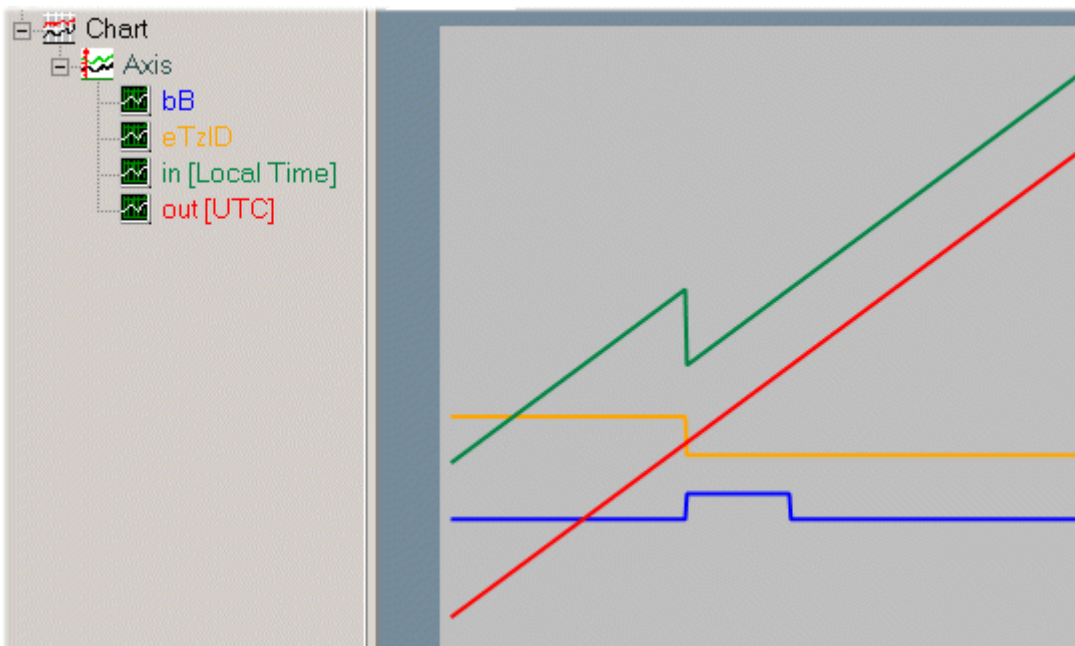


指定したタイムゾーン情報を考慮して、ファンクションブロックは現地時間(ファイル時間の形式)をUTC時間(ファイル時間の形式)に変換します。ファンクションブロックFB_TzSpecificLocalTimeToSystemTime [▶_108]には同様の機能がありますが、異なる時間の形式(構造体システム時間の形式)を使用します。

ブロックは、**連続的な**ローカルタイムスタンプ情報の変換にのみ適しています。夏時間/冬時間の切り替えによる現地時間のステップ変化は許可され、ブロックにより正確に検出されます。現地時間の任意の変更は、不正確な変換になります。理由: 現地時間がリセットされるときに、夏時間/冬時間情報とB時間(下記を参照)を識別するために、最後に変換された時間が内部的にブロックに保存されます。ブロックはアクションと関連があります: A_Reset()。このアクションがブロック出力をコールし、ローカルに保存された(最後に変換された)時間がゼロにリセットされます。

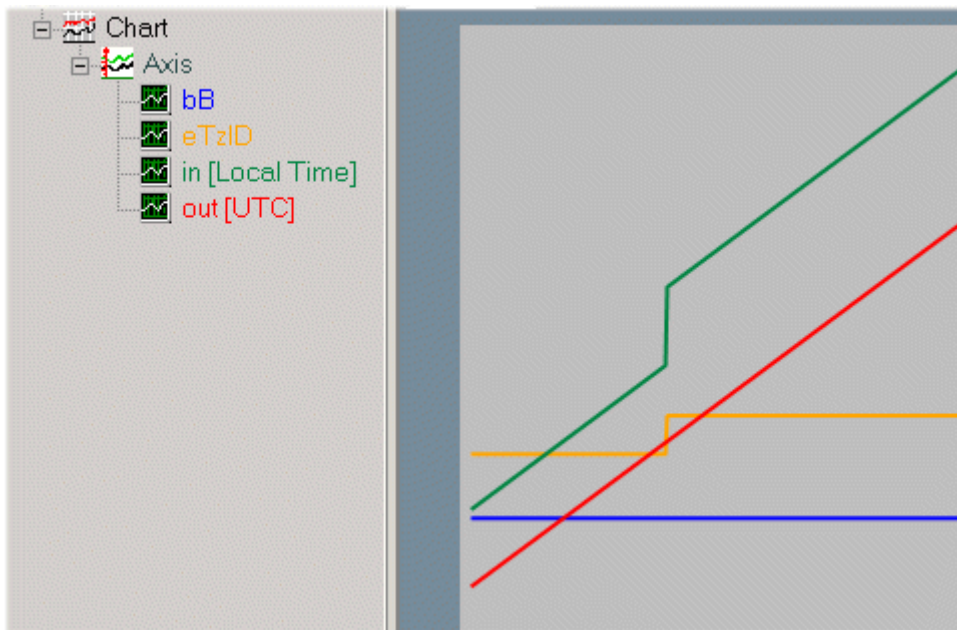
リニアUTC時間に変換する必要があるため、現地時間のステップ変化には問題があります。そのため、タイムスタンプタスクのために(連続的な) UTC時間を使用し、表示のために(視覚化など)それぞれの現地時間に時間を変換することを推奨します。

1. 夏時間から冬時間への切り替えのグラフ表示(tzInfo = WEST_EUROPE_TZI):



現地時間(緑)のジャンプバック。UTC出力時間(赤)が連続。現地時間: 02h:59m:59s:999ms. . の後にすぐに 02h:00m:00s:000ms. . が続きます。2時と3時の間の時間が2度発生します。切り替え前の複製時間が02:05:00 CEST A として参照されます。たとえば、切り替え後の時間は02:05:00 CET Bとして参照されます。出力変数**bB** は、最初または2番目のパスかどうかを示します。2番目のパスの間に、**bB**出力変数(青)はTRUEにセットされます。複製時間が経過すると、**bB**出力変数は自動的にリセットされます。タイムゾーンID (オレンジ)は、*eTimeZoneID_Daylight* (夏時間)から*eTimeZoneID_Standard* (冬時間)に変わります。

2. 冬時間から夏時間(tzInfo = WEST_EUROPE_TZI)への切り替えのグラフ表示。



現地時間（緑）ジャンプフォワードします。UTC出力時間（赤）が連続。現地時間：2h:59m:59s:999ms.. の後にすぐに：3h:00m:00s:000ms.. が続きます。タイムゾーンID（オレンジ）は、*eTimeZoneID_Standard*（冬時間）から *eTimeZoneID_Daylight*（夏時間）に変化します。

VAR_INPUT

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

in: 変換する現地時間（ファイル時間の形式）（[T_FILETIME](#) [[▶ 302](#)]）。

tzInfo: OSの現在のタイムゾーン情報をもつ構造体変数（[ST_TimeZoneInformation](#) [[▶ 300](#)]）。

VAR_OUTPUT

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

out: 変換されたUTC時間（ファイル時間の形式）（[T_FILETIME](#) [[▶ 302](#)]）。

eTzID: 追加の夏時間/冬時間情報（[E_TimeZoneID](#) [[▶ 288](#)]）。

bB: TRUE → B時間（例：02:05:00 CET B）、FALSE → 他の時間（例：02:05:00 GEST A）。この出力は現地時間がジャンプバックするとセットされ、複製時間が経過するとリセットされます。

例:

現地時間：DT#2011-09-02-11:01:31がUTC時間：DT#2011-09-02-09:01:31に変換されます。

```
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 );
```

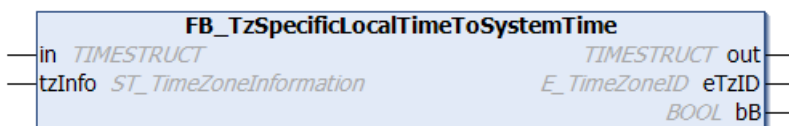
時間とタイムゾーンのその他のファンクションとファンクションブロック

- ・ [FB_TzSpecificLocalTimeToSystemTime](#) [[▶ 108](#)]
- ・ [FB_SystemTimeToTzSpecificLocalTime](#) [[▶ 104](#)]
- ・ [FB_FileTimeToTzSpecificLocalTime](#) [[▶ 52](#)]
- ・ [FB_GetTimeZoneInformation](#) [[▶ 69](#)]
- ・ [FB_SetTimeZoneInformation](#) [[▶ 101](#)]
- ・ [NT_SetLocalTime](#) [[▶ 115](#)]
- ・ [NT_GetTime](#) [[▶ 113](#)]
- ・ [NT_SetTimeToRTCTime](#) [[▶ 116](#)]
- ・ [F_TranslateFileTimeBias](#) [[▶ 144](#)]
- ・ [FB_LocalSystemTime](#) [[▶ 81](#)]

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.56 FB_TzSpecificLocalTimeToSystemTime



このファンクションブロックは、指定したタイムゾーン情報を考慮に入れて現地時間(構造体システム時間の形式)をUTC時間(構造体システム時間の形式)に変換します。ファンクションブロック

[FB_TzSpecificLocalTimeToFileTime](#) [[▶ 106](#)]には同様の機能がありますが、異なる時間の形式(ファイル時間の形式)を使用します。

ブロックは、**連続的な**ローカルタイムスタンプ情報の変換にのみ適しています。夏時間/冬時間の切り替えによる現地時間のステップ変化は許可され、ブロックにより正確に検出されます。現地時間の任意の変更は、不正確な変換になります。理由：現地時間がリセットされるときに、夏時間/冬時間情報とB時間(下記を参照)を識別するために、最後に変換された時間が内部的にブロックに保存されます。ブロックはアクションと関連があります：A.Reset()。このアクションがブロック出力をコールし、ローカルに保存された(最後に変換された)時間がゼロにリセットされます。

リニアUTC時間に変換する必要があるので、現地時間のステップ変化には問題があります。そのため、タイムスタンプタスクのために(連続的な) UTC時間を使用し、表示のために(視覚化など)それぞれの現地時間に時間を変換することを推奨します。

[FB_TzSpecificLocalTimeToFileTime](#) [[▶ 106](#)]ファンクションブロックについての詳細情報は、マニュアルに記載されています。

VAR_INPUT

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

in: 変換する現地時間(構造体システム時間の形式) ([TIMESTRUCT](#) [[▶ 306](#)])。

tzInfo: OSの現在のタイムゾーン情報をもつ構造体変数 ([ST_TimeZoneInformation](#) [[▶ 300](#)])。

VAR_OUTPUT

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

out: 変換されたUTC時間(構造体システム時間の形式) ([Timestruct](#) [[▶ 306](#)])。

eTzID: 追加の夏時間/冬時間情報([E_TimeZoneID](#) [[▶ 288](#)])。

bB: TRUE → B時間(例: 02:05:00 CET B)、FALSE → 他の時間(例: 02:05:00 CET A)。この出力は現地時間がジャンプバックするとセットされ、複製時間が経過するとリセットされます。

例:

```
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 );
```

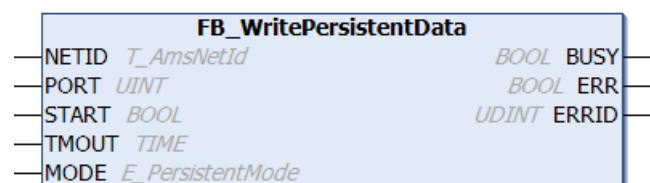
時間とタイムゾーンのその他のファンクションとファンクションブロック

- ・ [FB_TzSpecificLocalTimeToFileTime](#) [[▶ 106](#)]
- ・ [FB_SystemTimeToTzSpecificLocalTime](#) [[▶ 104](#)]
- ・ [FB_FileTimeToTzSpecificLocalTime](#) [[▶ 52](#)]
- ・ [FB_GetTimeZoneInformation](#) [[▶ 69](#)]
- ・ [FB_SetTimeZoneInformation](#) [[▶ 101](#)]
- ・ [NT_SetLocalTime](#) [[▶ 115](#)]
- ・ [NT_GetTime](#) [[▶ 113](#)]
- ・ [NT_SetTimeToRTCTime](#) [[▶ 116](#)]
- ・ [F_TranslateFileTimeBias](#) [[▶ 144](#)]
- ・ [FB_LocalSystemTime](#) [[▶ 81](#)]

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.57 FB_WritePersistentData



ファンクションブロックFB_WritePersistentDataは、[WritePersistentData \[▶ 137\]](#) ファンクションブロックの拡張バージョンです。[永続データを書き込むためのするためのシステム動作 \[▶ 338\]](#) (データの整合性/タスクサイクルタイムの超過)は、MODEパラメータによって影響を与えることができます。

VAR_INPUT

```
VAR_INPUT
    NETID      : T_AmsNetId;
    PORT       : UINT;
    START      : BOOL;
    TMOUT      : TIME := DEFAULT_ADS_TIMEOUT;
    MODE       : E_PersistentMode;
END_VAR
```

NETID: このパラメータを使用して、永続データが保存されるTwinCATコンピュータのAmsNetIDを指定できません (T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

PORT: PORTパラメータは、永続データが保存されるランタイムシステムを指定します。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

MODE : 永続データが書き込まれるモードされるモード ([E_PersistentMode \[▶ 286\]](#))。

VAR_OUTPUT

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

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットしません。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

例:

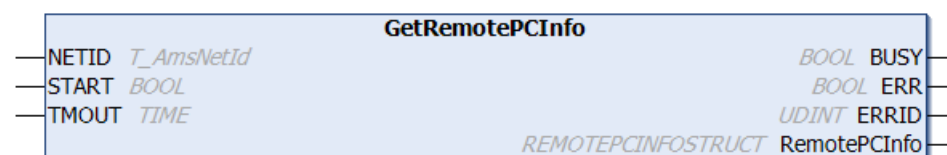
参照: [永続データの書き込み: システム動作 \[▶ 338\]](#)。

参照: [WritePersistentDataファンクションブロックのマニュアルでの例 \[▶ 137\]](#)。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3. 58 GetRemotePCInfo



ファンクションブロック「GetRemotePCInfo」を使用して、TwinCATルータ内の設定されたリモートPCの情報を読み込みできます。正常に実行されると、構造体「RemotePCInfo」は、TwinCATルータに保存された順序で、NetIDとリモートPCの名前を文字列として含みます。ファンクションブロックは、ローカル、またはリモートTwinCATシステムのいずれかに関連したルータ情報を読み込むことができますようにします。

VAR_INPUT

```
VAR_INPUT
  NETID      : T_AmsNetId;
  START      : BOOL;
  TMOUT      : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: このパラメータを使用して、そのルータ情報が設定されたリモートPCによって読み込まれるTwinCATコンピュータのAmsNetIDを指定できます(T_AmsNetID)。ローカルTwinCAT PCシステムのリモートPCを判定するために、空の文字列を指定できます。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY       : BOOL;
  ERR        : BOOL;
  ERRID      : UDINT;
  RemotePCInfo : REMOTEPCINFOSTRUCT;
END_VAR
```

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

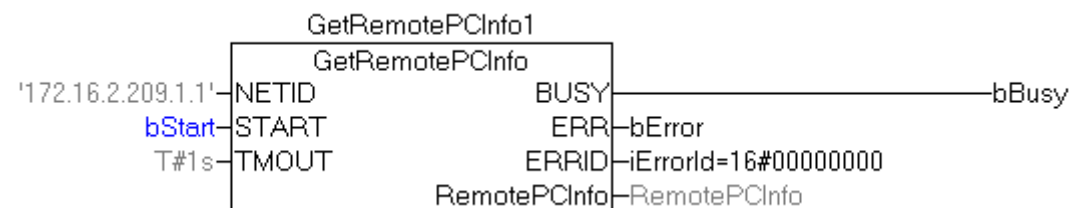
ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットします。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

RemotePCInfoStruct: 設定されたリモートPCについての情報を格納する構造体 ([REMOTEPCINFOSTRUCT \[▶ 291\]](#))。

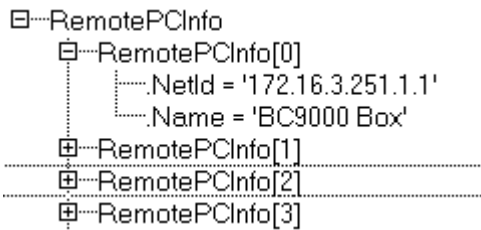
例:

```
PROGRAM MAIN
VAR
  GetRemotePCInfo1 : GetRemotePCInfo;
  RemotePCInfo     : REMOTEPCINFOSTRUCT;
  bBusy            : BOOL;
  bError           : BOOL;
  iErrorId         : UDINT;
  bStart           : BOOL;
END_VAR
```



オンラインビュー:

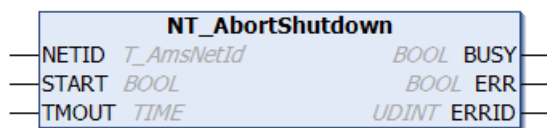
設定されたリモートPCのNetIDと名前。



要件

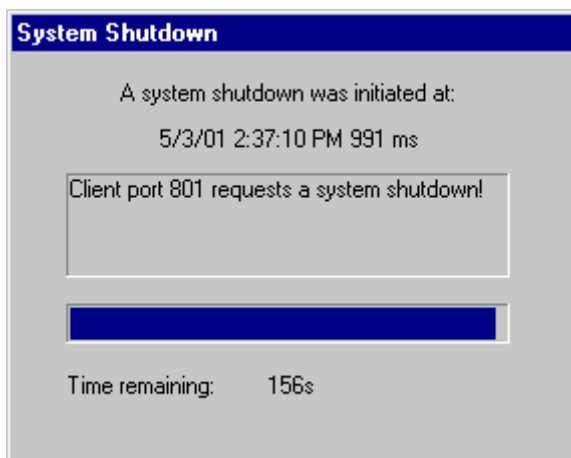
開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3. 59 NT_AbortShutdown



「NT_AbortShutdown」ファンクションブロックを使用して、NT_Shutdown [▶_117]ファンクションブロックで以前にコールしたシャットダウンコマンドを中止できます。

NT_Shutdown [▶_117]ファンクションブロックをコールする場合、遅延期間をパラメータとして指定できません。残りの時間は、メッセージウィンドウに表示されます。



遅延時間が経過した後で、OSはシャットダウンします。この間、シャットダウンプロセスは、「NT_AbortShutdown」ファンクションブロックを用いてPLCから割り込みできます。

VAR_INPUT

```

VAR_INPUT
    NETID   : T_AmsNetId;
    START   : BOOL;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
    
```

NETID: シャットダウンプロセスが中止されるTwinCATコンピュータのAmsNetIdを返します (T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

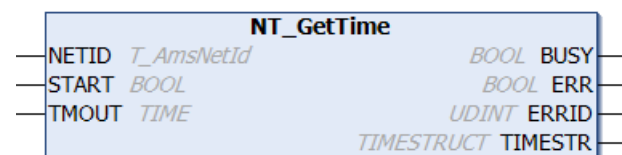
ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットしません。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.60 NT_GetTime



ファンクションブロック「NT_GetTime」を使用して、TwinCATシステムのローカルWindowsシステム時間を判定できます(ローカルWindowsシステム時間はタスクバーに表示されます)。年、月、日、曜日、時間、分、秒、およびミリ秒は、構造体[TIMESTRUCT \[▶ 306\]](#)の変数に保存されます。

VAR_INPUT

```
VAR_INPUT
  NETID  : T_AmsNetId;
  START  : BOOL;
  TMOUT  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: このパラメータを使用して、そのローカルWindowsシステム時間が決定されるTwinCATコンピュータのAmsNetIDを指定できます(T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できません。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY   : BOOL;
  ERR    : BOOL;
  ERRID  : UDINT;
  TIMESTR : TIMESTRUCT;
END_VAR
```

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットしません。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

TIMESTR: ローカルWindowsシステム時間をもつ構造体 ([TIMESTRUCT \[▶ 306\]](#))。

例:

参照: [例: ソフトウェアクロック \(RTC、RTC EX、RTC EX2\) \[▶ 336\]](#)。

その他の時間とタイムゾーンファンクションおよびファンクションブロック:

- ・ [FB_TzSpecificLocalTimeToSystemTime \[▶ 108\]](#)
- ・ [FB_TzSpecificLocalTimeToFileTime \[▶ 106\]](#)
- ・ [FB_SystemTimeToTzSpecificLocalTime \[▶ 104\]](#)
- ・ [FB_FileTimeTimeToTzSpecificLocalTime \[▶ 52\]](#)
- ・ [FB_GetTimeZoneInformation \[▶ 69\]](#)
- ・ [FB_SetTimeZoneInformation \[▶ 101\]](#)
- ・ [NT_SetLocalTime \[▶ 115\]](#)
- ・ [NT_SetTimeToRTCTime \[▶ 116\]](#)
- ・ [F_TranslateFileTimeBias \[▶ 144\]](#)
- ・ [FB_LocalSystemTime \[▶ 81\]](#)

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.61 NT_Reboot



Windows NTOSは、「NT_Reboot」ファンクションブロックで再起動できます。ファンクションは、ほぼ、Windowsタスクバーの再起動コマンドに相当します。再起動コマンドの実行前の遅延は、DELAYパラメータで定義できます。

VAR_INPUT

```

VAR_INPUT
    NETID    : T_AmsNetId;
    DELAY    : DWORD;
    START    : BOOL;
    TMOUT    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
  
```

NETID: 再起動するTwinCATコンピュータのAmsNetIdを返します (T_AmsNetID)。ローカルコンピュータで再起動を実行する場合、空の文字列を入力できます。

DELAY: 再起動コマンドが実行される前の秒単位の遅延時間。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットしません。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.62 NT_SetLocalTime



ファンクションブロック「NT_SetLocalTime」を使用して、TwinCATシステムのローカルWindowsシステム時間をセットできます(ローカルWindowsシステム時間はタスクバーに表示されます)。

VAR_INPUT

```
VAR_INPUT
  NETID : T_AmsNetId;
  START : BOOL;
  TIMESTR : TIMESTRUCT;
  TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: このパラメータを使用して、そのローカルWindowsシステム時間がセットされるTwinCATコンピュータのAmsNetIDを指定できます(T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TIMESTR: 新しいローカルWindowsシステム時間をもつ構造体(TIMESTRUCT [\[▶ 306\]](#))。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットしません。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

時間とタイムゾーンのその他のファンクションとファンクションブロック

- ・ [FB_TzSpecificLocalTimeToSystemTime \[▶ 108\]](#)
- ・ [FB_TzSpecificLocalTimeToFileTime \[▶ 106\]](#)
- ・ [FB_SystemTimeToTzSpecificLocalTime \[▶ 104\]](#)
- ・ [FB_FileTimeToTzSpecificLocalTime \[▶ 52\]](#)
- ・ [FB_GetTimeZoneInformation \[▶ 69\]](#)
- ・ [FB_SetTimeZoneInformation \[▶ 101\]](#)
- ・ [NT_GetTime \[▶ 113\]](#)
- ・ [NT_SetTimeToRTCTime \[▶ 116\]](#)
- ・ [F_TranslateFileTimeBias \[▶ 144\]](#)
- ・ [FB_LocalSystemTime \[▶ 81\]](#)

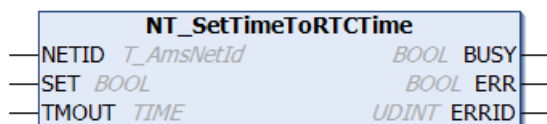
要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.63 NT_SetTimeToRTCTime



この機能は、Windows CEのPLCランタイムシステムでは使用できません



ファンクションブロック「Nt_SetTimeToRtcTime」を使用して、ローカルWindowsシステム時間をPC (BIOSのRTC時間)のリアルタイムクロックと同期できます (タスクバーに表示)。

コメント

ファンクションブロックがコールされるとき、TwinCAT PCのリアルタイムクロックはローカルWindowsシステム時間と比較され、ローカルWindowsシステム時間は差を訂正されます。タイムゾーンと夏時間が考慮されます。計測中、またはログブックのエントリ中にタイムジャンプが発生する可能性があることに注意してください。

ローカルWindowsシステム時間がセットされると、OSは自動的にRTC時間を新しいローカルWindowsシステム時間にセットします。変換と遅延のために、新しいRTC時間は必然的に小さなエラーを被りがちです。このエラーはミリ秒の範囲です。すなわち、NT_SetTimeToRTCTimeがコールされるたびに、小さなエラーがリアルタイムクロックに取り込まれます。長い間のずれを最小化するために、たとえば、各PLCサイクルの間よりもむしろ24時間ごとに補正を実行してください。

VAR_INPUT

```

VAR_INPUT
  NETID   : T_AmsNetId;
  SET     : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR

```

NETID: このパラメータを使用して、そのローカルWindowsシステム時間が同期されるTwinCATコンピュータのAmsNetIDを指定できます(T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できません。

SET: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```

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

```

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットしません。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

時間とタイムゾーンのその他のファンクションとファンクションブロック

- ・ [FB TzSpecificLocalTimeToSystemTime \[▶ 108\]](#)
- ・ [FB TzSpecificLocalTimeToFileTime \[▶ 106\]](#)
- ・ [FB SystemTimeToTzSpecificLocalTime \[▶ 104\]](#)
- ・ [FB FileTimeToTzSpecificLocalTime \[▶ 52\]](#)
- ・ [FB GetTimeZoneInformation \[▶ 69\]](#)
- ・ [FB SetTimeZoneInformation \[▶ 101\]](#)
- ・ [NT SetLocalTime \[▶ 115\]](#)
- ・ [NT GetTime \[▶ 113\]](#)
- ・ [F TranslateFileTimeBias \[▶ 144\]](#)
- ・ [FB LocalSystemTime \[▶ 81\]](#)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.64 NT_Shutdown

この機能はWindows CEでは使用できません



Windows NTOSは、「NT_Shutdown」ファンクションブロックを使用してシャットダウンできます。ファンクションは、ほぼ、Windowsタスクバーのシャットダウンコマンドに相当します。シャットダウンコマンドの実行前の遅延は、DELAYパラメータで定義できます。

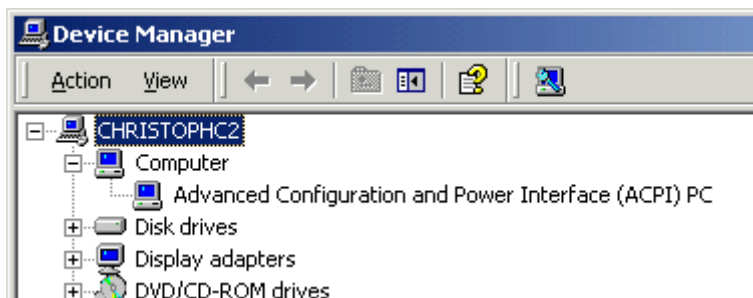
注記:

新しいOS (Windows 2000など) では、「NT_Shutdown」ファンクションブロックで、「Shutdown with Power OFF」を実行します(コンピュータがその電源をオフにします)。このファンクションは、ACPI (電力制御インターフェイス) 準拠のシステムでのみ使用できます。ACPIファンクションは、OSのインストールの前にBIOSで有効にしてください。

ACPIファンクションでは、マザーボードとPCの電源が対応していなければいけません。後で変更しても、OSは認識しません。PCがACPIに対応しているかどうかは、次の手順でWindows 2000などでチェックできます。

1. [システムマネージャ]で、フォルダ[system]を開きます。
2. タブ[ハードウェア]で、[デバイスマネージャ]を選択します。

デバイスのナビゲーションツリーの[コンピュータ]で、[電力制御インターフェイス (ACPI) PC]を読み込みできます。



デフォルトのTwinCAT設定では、シャットダウンして電源をオフにします。Windowsのレジストリで電源オフファンクションを無効にできます。以下のエントリを追加してください。

「DisableACPIPowerOff」 「REG_DWORD = 0x00000001」
 (「HKEY_LOCAL_MACHINE\SOFTWARE\Beckhoff\TwinCAT\System」 の下のレジストリ)

VAR_INPUT

```
VAR_INPUT
NETID   : T_AmsNetId;
DELAY   : DWORD;
START   : BOOL;
TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: OSがシャットダウンされるTwinCATコンピュータのAmsNetIdを返します(T_AmsNetID)。ローカルコンピュータをシャットダウンする場合、空の文字列を入力できます。

DELAY: シャットダウンコマンドが実行される前の秒単位の遅延時間。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```

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

```

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

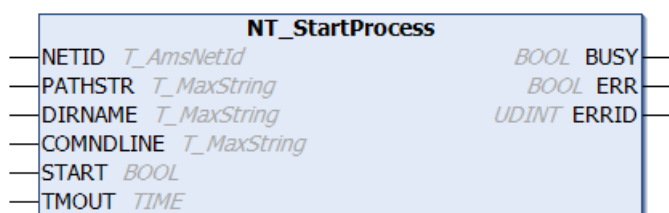
ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットします。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.65 NT_StartProcess



「NT_StartProcess」ファンクションブロックを使用して、PLCからWindowsアプリケーションを起動できます。また、ファンクションブロックはリモートPCでアプリケーションを実行できます。

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: アプリケーションが起動するTwinCATコンピュータのAmsNetIdを返します(T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

PATHSTR: 実行されるアプリケーションのパス全体の文字列(T_MaxString、例: 「C:\\$WINNT\\$NOTEPAD.EXE」)。

DIRNAME: 実行されるアプリケーションの作業ディレクトリの文字列(T_MaxString、例: 「C:\\$WINNT」)。

COMNDLINE: コマンドラインパラメータ(T_MaxString、例: 「win.ini」)。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

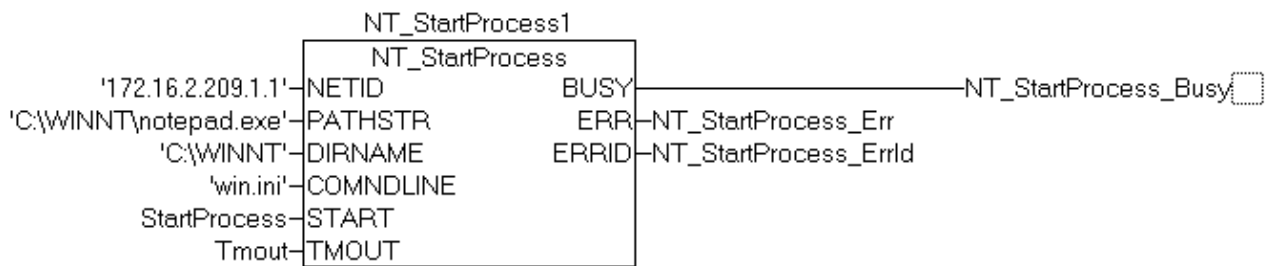
BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットしません。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)、またはWin32エラーコード [[▶ 345](#)]を返しません(プラットフォームSDK: Win32 API)。

例:

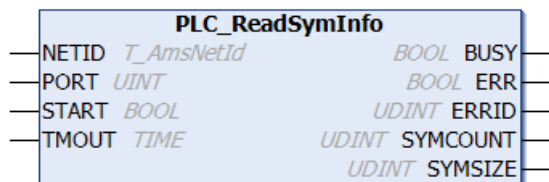
```
NT_StartProcess1 : NT_StartProcess;
NT_StartProcess_Busy : BOOL;
NT_StartProcess_Err : BOOL;
NT_StartProcess_ErrId : UDINT;
StartProcess : BOOL;
Tmout : TIME;
```



要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3. 66 PLC_ReadSymInfo



「PLC_ReadSymInfo」ファンクションブロックは、PLCランタイムシステムのシンボル(変数)に関する情報を許可します。

VAR_INPUT

```
VAR_INPUT
    NETID : T_AmsNetId;
    PORT  : T_AmsPort;
    START : BOOL;
    TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```


NETID: そのシンボル情報が検出されるTwinCATコンピュータのAmsNetIdを返します(T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

PORT: PLCランタイムシステムのポート番号(T_AmsPort)。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY      : BOOL;
    ERR       : BOOL;
    ERRID     : UDINT;
    SYMCOUNT  : UDINT;
    SYMSIZE   : UDINT;
END_VAR
```

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットします。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

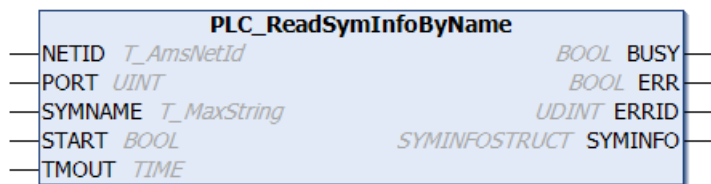
SYMCOUNT: PLCランタイムシステムのシンボル数。

SYMSIZE: シンボル情報が保存されるデータ長(バイト単位)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.67 PLC_ReadSymInfoByName



ファンクションブロック「PLC_ReadSymInfoByName」は、PLCシンボル変数についての追加情報(データ型ID、インデックスグループ、インデックスオフセット、コメント...など)をシンボル名を使用して読み込みできるようにします。正常に実行されると、データはそのタイプがSYMINFOSTRUCTであるSymInfoデータ構造体で使用できます。ファンクションブロックにパラメータとして送信されるシンボル名の最大長は、255文字に制限されます。

VAR_INPUT

```
VAR_INPUT
    NETID      : T_AmsNetId;
    PORT       : T_AmsPort;
    SYMNAME    : T_MaxString;
    START      : BOOL;
    TMOUT      : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: ファンクションが実行されるTwinCATコンピュータのAmsNetIdを入力できます(T_AmsNetID)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

PORT: シンボル変数が属するPLCランタイムシステムのポート番号(T_AmsPort)。

SYMNAME: 情報が読み込まれるPLC変数のシンボル名(T_MaxString、「MAIN.INIT_TASK.VARINT」などのパス全体を含む最大255文字)。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY      : BOOL;
  ERR       : BOOL;
  ERRID     : UDINT;
  SymInfo   : SYMINFOSTRUCT;
END_VAR
```

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

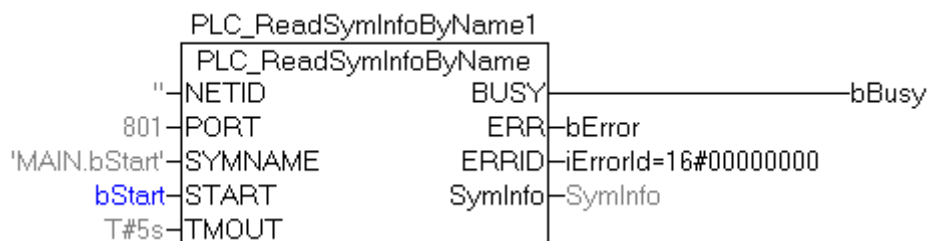
ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットします。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

SymInfo: シンボル変数についての追加情報をもつ構造体([SYMINFOSTRUCT \[▶ 301\]](#))。

例:

```
PROGRAM MAIN
VAR
  PLC_ReadSymInfoByName1 : PLC_ReadSymInfoByName;
  bStart AT%QX0.5        : BOOL; (*Starts FB execution*)
  bBusy                  : BOOL;
  bError                 : BOOL;
  iErrorId               : UDINT;
  SymInfo                : SYMINFOSTRUCT;(*Structure with sombol information*)
END_VAR
```



オンラインビュー:

```
-----SymInfo-----
.....symEntryLen = 16#00000043
.....idxGroup = 16#0000F031
.....idxOffset = 16#00000005
.....byteSize = 16#00000001
.....adsDataType = ADST_BIT
.....symDataType = 'BOOL'
.....symComment = 'STARTS FB EXECUTION'
```

このようにして取得したデータは、次の意味があります。

symEntryLen = 16#43: シンボルテーブルのエントリの実際長は67バイトです。

idxGroup = 16#F031: 物理的出力のPLCプロセスイメージからの変数です。

idxOffset = 16#5: 変数は、バイトオフセットゼロおよびビットオフセット5の位置にあります。

byteSize = 16#1: 変数の値は、1バイトのメモリを占有します。

adsDataType = ADST_BIT: ADSデータ型ID。

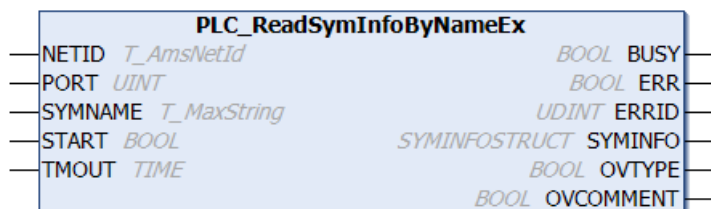
symDataType = BOOL: PLCのデータ型ID。

symComment = 'STARTS FB EXECUTION': ユーザが変数の定義行に追加したコメント。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3. 68 PLC_ReadSymInfoByNameEx



ファンクションブロック「PLC_ReadSymInfoByNameEx」には、ファンクションブロック [PLC_ReadSymInfoByName](#) [▶ 121]と同様の機能があります。両方のファンクションブロックは、シンボル名でシンボル情報を読み込みできます。2つのブロックの相違は、ここに記載したブロックは使用可能なバッファサイズを超過して、不完全な情報を出力した可能性がある場合でも、エラーを報告しないことです。この場合、コメントやデータ型IDが切り捨てられた可能性があります。2つの追加の出力変数、すなわち *OVTYPE*および *OVCOMMENT*はこうした現象が発生したことを示します。そのため、アプリケーションは適切にレスポンスできます。

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  PORT    : T_AmsPort;
  SYMNAME : T_MaxString;
  START   : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: ファンクションが実行されるTwinCATコンピュータのAmsNetIdを入力できます(T_AmsNetId型)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

PORT: シンボル変数が属するPLCランタイムシステムのポート番号(T_AmsPort型)。

SYMNAME: 情報が読み込まれるPLC変数のシンボル名(T_MaxString型、「MAIN.INIT_TASK.VARINT」などのパス全体を含む最大255文字)。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY    : BOOL;
  ERR     : BOOL;
  ERRID   : UDINT;
```

```
SymInfo : SYMINFOSTRUCT;
OVRTYPE : BOOL;
OVCOMMENT : BOOL;
END_VAR
```

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットしません。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

SymInfo: シンボル変数についての追加情報をもつ構造体 ([SYMINFOSTRUCT \[▶ 301\]](#)型)。

OVRTYPE: データ型IDをもつ文字列がオーバーフロー (TRUE) を起こしたかどうかを示します。データ型IDをもつ文字列が、切り捨てられた可能性があります。

OVCOMMENT: シンボルコメントをもつ文字列がオーバーフロー (TRUE) を起こしたかどうかを示します。コメントのある文字列が、切り捨てられた可能性があります。

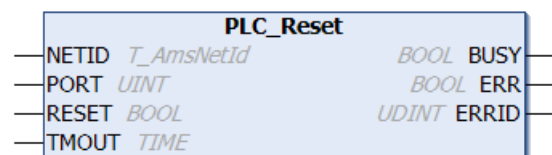
例:

ファンクションブロック [PLC_ReadSymInfoByName \[▶ 121\]](#) についてのマニュアルを参照してください。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3. 1. 0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3. 69 PLC_Reset



「PLC_Reset」ファンクションブロックを使用して、PLCランタイムシステムをリセットできます。PLCがリセットされると、PLC変数は初期値で満たされ、PLCプログラムの実行は停止します。

VAR_INPUT

```
VAR_INPUT
NETID : T_AmsNetId;
PORT : T_AmsPort;
RESET : BOOL;
TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: PLCランタイムシステムがリセットされるTwinCATコンピュータのAmsNetIdを返します (T_AmsNetID型)。PLCリセットをローカルコンピュータで実行する場合、空の文字列を入力できます。

PORT: リセットされるPLCランタイムシステムのADSポート番号を格納します (T_AmsPort型)。

RESET: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

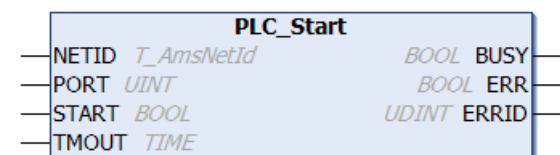
ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットしません。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.70 PLC_Start



ファンクションブロック「PLC_Start」を使用して、TwinCATコンピュータのPLCランタイムシステムを起動できます。たとえば、ファンクションブロックを使用して、リモートPCのPLCを起動できます。

VAR_INPUT

```
VAR_INPUT
  NETID : T_AmsNetId;
  PORT  : T_AmsPort;
  START : BOOL;
  TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: このパラメータを使用して、PLCが起動するTwinCATコンピュータのAmsNetIDを指定できます(T_AmsNetID型)。PLCをローカルコンピュータで起動する場合、空の文字列を入力できます。

PORT: 起動するPLCランタイムシステムのADSポート番号を格納します(T_AmsPort型)。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットしません。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.71 PLC_Stop



ファンクションブロック「PLC_Stop」を使用して、TwinCATコンピュータのPLCランタイムシステムをt停止できます。たとえば、このファンクションブロックを使用して、リモートまたはローカルPCでのPLCを停止できます。

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  PORT    : T_AmsPort;
  STOP    : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: このパラメータを使用して、PLCが停止されるTwinCATコンピュータのAmsNetIDを指定できます (T_AmsNetID型)。PLCをローカルコンピュータで停止する場合、空の文字列を入力できます。

PORT: 停止するPLCランタイムシステムのADSポート番号を格納します (T_AmsPort型)。

STOP: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットします。

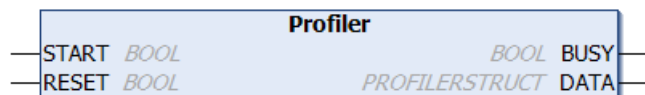
ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.72 プロファイラ

i この機能は、Windows CE環境のPLCでは使用できません



「Profiler」ファンクションブロックを使用して、PLCコードの実行時間を計測できるようにします。内部的に、GETCPUACCOUNTファンクションブロックのインスタンスがコールされます。計測はSTART入力の立ち上がりで開始し、立ち下がりで停止します。計測は内部的に評価され、タイプPROFILERSTRUCT [▶ 290]の構造体のDATA出力でさらに処理するために利用できます。現在の実行時間、最小実行時間および最大実行時間と共に、ファンクションブロックは最後の10個の計測値の平均実行時間を計算します。平均計測値の数は、2～100の間でグローバル変数MAX_AVERAGE_MEASURES [▶ 309]を使用して設定できます。計測時間はマイクロ秒単位です。出力変数DATA.MeasureCycle [▶ 290]は、既に実行された計測数についての情報を返します。PLCプログラムの特定セグメントの実行時間を計測するためには、計測されるセグメントが開始するSTART入力の立ち上がりで計測を開始し、セグメントの終わりのSTART入力の立ち下がりで計測を停止しなければいけません。STARTでの立ち上がりと同じ時間で立ち上がりがRESET入力で生成される場合、DATA出力でのすべての値はリセットできます。新しい計測が開始し、ファンクションブロックのその後のコールから再計算される場合、古い計測値はリセットされます。

コメント:

ある特定の時間がGETCPUACCOUNTファンクションブロックをコールするためにのみ必要なので、計測時間は実際の値と異なっていることがあります。この時間は個別のコンピュータによって異なり、検出される時間に含まれます。

VAR_INPUT

```
VAR_INPUT
  START   : BOOL;
  RESET   : BOOL;
END_VAR
```

START: この入力の立ち上がりで、実行時間の計測を開始します。この入力の立ち下がりには計測を停止し、現在、最小、最大、平均実行時間を再計算させます。変数DATA.MeasureCycle [▶ 290]は、同時にインクリメントします。

RESET: START入力での立ち上がりと同時にこの入力での立ち上がりで生成される場合、DATA出力のすべての変数はリセットされます。現在、最小、最大、および平均実行時間の古い値はリセットされ、次の計測のために再計算されます。

VAR_OUTPUT

```
VAR_OUTPUT
  BUSY    : BOOL;
  DATA   : PROFILERSTRUCT;
END_VAR
```

BUSY: 計測プロセスの開始時にこの入力にセットされると、計測が完了するまでセットされたままの状態になります。BUSY出力がリセットされると、最新の時間がDATA出力で使用可能になります。

DATA: 計測時間[マイクロ秒単位]をもつタイプPROFILERSTRUCT [▶ 290]の構造体。

例1:

```
PROGRAM ProfilerTest_ST
VAR
  Profiler1   : PROFILER;
```

```

ProfilerData : PROFILERSTRUCT;
a            : LREAL;
END_VAR

```

計測時間のオンライン表示:

```

+ 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)));           a = 0.370490944866529

Profiler1(Start=FALSE);
ProfilerData:=Profiler1.Data;

```

例2:

```

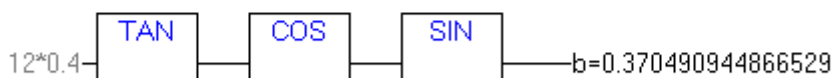
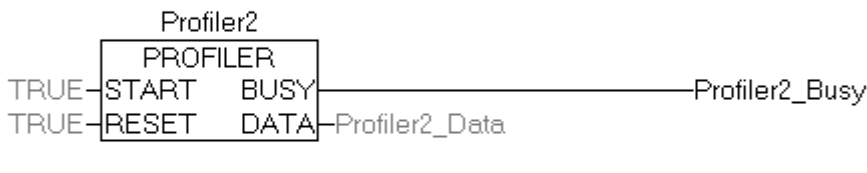
PROGRAM ProfilerTest_FUP
VAR
  Profiler2      :PROFILER;
  Profiler2_Busy :BOOL;
  Profiler2_Data :PROFILERSTRUCT;
  b              :LREAL;
END_VAR

```

計測時間のオンライン表示:


```

    ▣ Profiler2
      Profiler2_Busy = FALSE
    ▣ Profiler2_Data
      LastExecTime = 16#00000002
      MinExecTime = 16#00000002
      MaxExecTime = 16#00000002
      AverageExecTime = 16#00000002
      MeasureCycle = 16#00001FBB
      b = 0.370490944866529
  
```



要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3. 1. 0	PCまたはCX (x86、x64)	Tc2_Utilities (システム)

3. 73 RTC



ファンクションブロック「RTC」(リアルタイムクロック)を使用して、TwinCAT PLCで内部ソフトウェアクロックを実現できます。クロックは、開始日付と時間で初期化する必要があります。初期化の後で、日時はファンクションブロックのそれぞれのコールで更新されます。CPUシステムクロックを使用して、現在の時間と日付を計算します。現在の時間を計算できるように、ファンクションブロックはPLCサイクルごとにコールする必要があります。現在の時間と日付は、ファンクションブロックの出力で通常DATE_AND_TIME (DT)形式で利用できます。RTCファンクションブロックの複数のインスタンスは、1つのPLCプログラムで読み込みできます。

● RTC時間は、基準時間とは異なることがあります。

I システムの動作方法のために、RTC時間が基準時間と異なることがあることを意味します。PLCのサイクルタイム、基準のシステムティックの値、使用しているハードウェアにより時間のずれは異なります。

大きく時間がずれることを避けるために、RTCインスタンスは周期的に同期(電波時計やローカルWindowsシステム時間と)させる必要があります。ローカルWindowsシステム時間は、SNTPプロトコルで基準時間と同期できます。

VAR_INPUT

```
VAR_INPUT
  EN   : BOOL;
  PDT  : DATE_AND_TIME;
END_VAR
```

EN: ファンクションブロックは、この入力での立ち上がりで指定した日付および時間で再度初期化されません。

PDT: (プリセットされた日付と時間) ファンクションブロックの日付と時間の初期化値。EN入力の立ち上がりによってファンクションブロックはこの値を採用します。

VAR_OUTPUT

```
VAR_OUTPUT
  Q     : BOOL;
  CDT  : DATE_AND_TIME;
END_VAR
```

Q: ファンクションブロックが少なくとも一度初期化された場合に、この出力をセットします。出力がセットされると、PDT出力での日付と時間の値が有効になります。

CDT: RTCインスタンスの現在の日付と時間。CDT出力は、ファンクションブロックがコールされる場合にのみ更新されます。このため、ファンクションブロックのインスタンスは、各PLCサイクルで1回コールする必要があります。

例:

参照: [例: ソフトウェアクロック \(RTC、RTC EX、RTC EX2\) \[▶ 336\]](#)。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

3.74 RTC_EX

「RTC_EX」(拡張リアルタイムクロック)ファンクションブロックは、内部ソフトウェアクロックをTwinCAT PLCに実装できます。クロックは、開始日付と時間で初期化する必要があります。初期化の後で、日時はファンクションブロックのそれぞれのコールで更新されます。CPUシステムクロックを使用して、現在の時間と日付を計算します。現在の時間を計算できるように、ファンクションブロックはPLCサイクルごとにコールする必要があります。現在の時間と日付は、ファンクションブロックの出力で通常DATE_AND_TIME (DT)形式で利用できます。[RTC \[▶ 129\]](#) ファンクションブロックとは対照的に、RTC_EXの精度は1ミリ秒です。RTC_EXファンクションブロックの複数のインスタンスは、1つのPLCプログラム内で作成できます。

● RTC_EX時間は基準時間と異なることがあります。

i システムの動作方法のために、RTC_EX時間が基準時間と異なることがあることを意味します。PLCのサイクルタイム、基準のシステムティックの値、使用しているハードウェアにより時間のずれは異なります。

大きく時間がずれることを避けるために、RTC_EXインスタンスは周期的に同期(電波時計やローカルWindowsシステム時間と)させる必要があります。ローカルWindowsシステム時間は、SNTPプロトコルで基準時間と同期できます。

VAR_INPUT

```
VAR_INPUT
  EN      : BOOL;
  PDT     : DATE_AND_TIME;
  PMSEK   : DWORD;
END_VAR
```

EN: RTC_EXファンクションブロックは、この入力での立ち上がりで指定した日付、時間およびミリ秒で再度初期化されます。

PDT: (プリセットされた日付と時間) ファンクションブロックの日付と時間の初期化値。EN入力の立ち上がりによってファンクションブロックはこの値を採用します。

PMSEK: (プリセットされたミリ秒) ミリ秒の初期化値。EN入力の立ち上がりによってファンクションブロックはこの値を採用します。

VAR_OUTPUT

```
VAR_OUTPUT
  Q       : BOOL;
  CDT     : DATE_AND_TIME;
  CMSEK   : DWORD;
END_VAR
```

Q: ファンクションブロックが少なくとも一度初期化された場合に、この出力をセットします。出力がセットされると、PDTとCMSEK出力での日付、時間、ミリ秒が有効になります。

CDT: RTC_EXインスタスの現在の日付と時間。CDT出力は、ファンクションブロックがコールされる場合のみ更新されます。このため、ファンクションブロックのインスタスは、各PLCサイクルで1回コールする必要があります。

CMSEK: (現在のミリ秒) ミリ秒の出力。

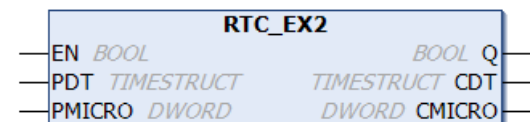
例:

参照: [例: ソフトウェアクロック \(RTC、RTC EX、RTC EX2\) \[▶ 336\]](#)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PGまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.75 RTC_EX2



「RTC_EX2」(拡張リアルタイムクロック)ファンクションブロックは、内部ソフトウェアクロックをTwinCAT PLCに実装できます。クロックは、開始日付と時間で初期化する必要があります。初期化の後で、日時はファンクションブロックのそれぞれのコールで更新されます。CPUシステムクロックを使用して、現在の時間と日付を計算します。現在の時間を計算できるように、ファンクションブロックはPLCサイクルごととにコールされる必要があります。ファンクションブロック出力で、現在の日付と時間はWindowsシステム時間の形式で使用できます。[RTC \[▶ 129\]](#)ファンクションブロックとは対照的に、RTC_EX2の精度はマイクロ秒です。RTC_EX2ファンクションブロックの複数のインスタスは、1つのPLCプログラム内で作成できません。

● RTC_EX2時間は、基準時間とは異なることがあります。

i システムの動作方法のために、RTC_EX2時間が基準時間と異なることがあることを意味します。PLCのサイクルタイム、基準のシステムティックの値、使用しているハードウェアにより時間のずれは異なります。

大きく時間がずれることを避けるために、RTC_EX2インスタンスは周期的に同期（電波時計やローカルWindowsシステム時間と）させる必要があります。ローカルWindowsシステム時間は、SNTPプロトコルで基準時間と同期できます。

VAR_INPUT

```
VAR_INPUT
  EN      : BOOL;
  PDT     : TIMESTRUCT;
  PMICRO  : DWORD;
END_VAR
```

EN: RTC_EX2ファンクションブロックはこの入力での立ち上がりによって、指定した日付、時間、およびミリ秒で再初期化されます。

PDT: (プリセットされた日付と時間) ファンクションブロックの日付と時間の初期化値 (TIMESTRUCT [▶ 306]型)。EN入力の立ち上がりによってファンクションブロックはこの値を採用します。

PMICRO: (プリセットされたミリ秒) ミリ秒の初期化値。EN入力の立ち上がりによってファンクションブロックはこの値を採用します。

VAR_OUTPUT

```
VAR_OUTPUT
  Q       : BOOL;
  CDT     : TIMESTRUCT;
  CMICRO  : DWORD;
END_VAR
```

Q: ファンクションブロックが少なくとも一度初期化された場合に、この出力をセットします。出力がセットされると、PDTとCMICRO出力での日付、時間、ミリ秒が有効になります。

CDT: RTC_EX2インスタンスの現在の日付と時間 (TIMESTRUCT [▶ 306]型)。CDT出力は、ファンクションブロックがコールされる場合にのみ更新されます。このため、ファンクションブロックのインスタンスは、各PLCサイクルで1回コールする必要があります。

CMICRO: (現在のミリ秒) ミリ秒の出力。

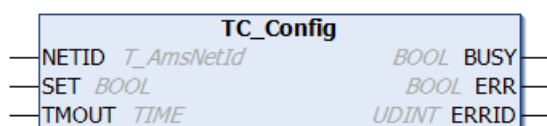
例:

参照: 例: ソフトウェアクロック (RTC、RTC EX、RTC EX2) [▶ 336]。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.76 TC_Config



RUNモードのTwinCATシステム(緑のTwinCATシステムのアイコン)は、ファンクションブロック「TC_Config」によってCONFIGモード(青いTwinCATシステムのアイコン)へ切り替えることができます。システムが既にCONFIGモードの場合、最初にSTOPモード(赤いTwinCATシステムのアイコン)に切り替わり、それからCONFIGモードに切り替わります。

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  SET     : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: このパラメータを使用して、CONFIGモードに切り替わるTwinCATコンピュータのAmsNetIDを指定できます(T_AmsNetID型)。ローカルTwinCAT PCコンピュータで実行する場合、空の文字列を入力できます。

SET: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

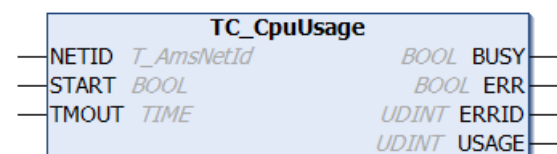
ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットします。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.77 TC_CpuUsage



「TC_CpuUsage」ファンクションブロックは、TwinCATシステムの現在のCPU負荷を判定できるようにします。このファンクションは、リアルタイム設定でTwinCATシステムメニューのCPU負荷の表示に相当します。

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  START   : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: CPU負荷を判定するTwinCATコンピュータのAmsNetIDを返します(T_AmsNetID型)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY : BOOL;
    ERR : BOOL;
    ERRID : UDINT;
    USAGE : UDINT;
END_VAR
```

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

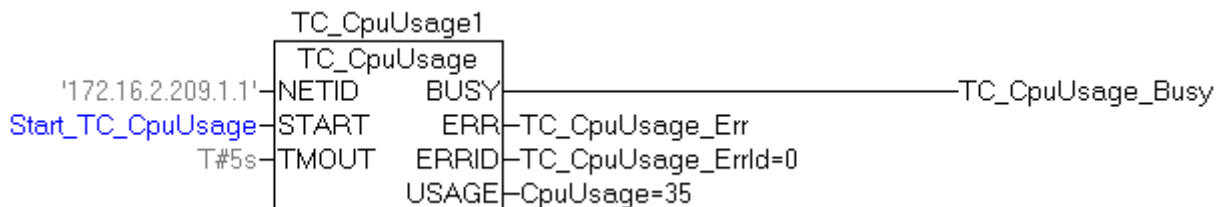
ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットします。

ERRID: ERR出力がセットされると、ADSエラー番号 [▶ 342]を返します。

USAGE: TwinCATシステムの現在のCPU負荷(%)。

例:

```
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
```



例では、TwinCATシステムは、使用可能なCPU計算時間の合計の35%を使用しています。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3. 78 TC_Restart



ファンクションブロック「TC_Restart」は、TwinCATシステムを再起動できます。ファンクションは、TwinCATシステムメニューの再起動コマンド(Windowsタスクバーの右)に相当します。TwinCATシステムの再起動では、最初にTwinCATシステムが停止し、その後、直ちに再度起動します。

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  RESTART : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: TwinCATシステムが再起動されるTwinCATコンピュータのAmsNetIdを返します(T_AmsNetID型)。ローカルコンピュータで再起動を実行する場合、空の文字列を入力できます。

RESTART: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットします。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.79 TC_Stop

ファンクションブロック「TC_Stop」を使用して、TwinCATシステムを停止できます。ファンクションは、TwinCATシステムメニューの停止コマンド(Windowsタスクバーの右)に相当します。

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  STOP    : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: TwinCATシステムが停止するTwinCATコンピュータのAmsNetIdを返します(T_AmsNetID型)。TwinCATシステムをローカルコンピュータで停止する場合、空の文字列を入力できます。

STOP: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

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

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットします。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

3.80 TC_SysLatency



ファンクションブロック「TC_SysLatency」を使用して、TwinCATシステムの 現在および最大遅延時間を判定できます。このファンクションは、リアルタイム設定でTwinCATシステムメニューのTwinCATの遅延時間の表示に相当します。

VAR_INPUT

```
VAR_INPUT
    NETID : T_AmsNetId;
    START : BOOL;
    TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: このパラメータを使用して、その遅延時間が判定されるTwinCATコンピュータのAmsNetIDを指定できます(T_AmsNetID型)。ローカルコンピュータで実行する場合、空の文字列を入力できます。

START: ブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドの実行で、超過してはならないタイムアウトの長さを表します。

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY : BOOL;
    ERR : BOOL;
    ERRID : UDINT;
    ACTUAL : UDINT;
    MAXIMUM : UDINT;
END_VAR
```

BUSY: この出力はファンクションブロックの実行中にセットされます。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットしません。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

ACTUAL: TwinCATシステムの現在の遅延時間(マイクロ秒単位)。

MAXIMUM: TwinCATシステムの最大遅延時間(マイクロ秒単位) (TwinCATシステムが最後に起動してからの最大遅延時間)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

3.81 WritePersistentData



永続変数がPLCランタイムシステムで定義されているとき、TwinCATシステムを停止/シャットダウンする場合、その現在値は通常TwinCAT¥Bootフォルダの.bootdataファイルに保存されます(最後のPLCサイクルの後で)。現在の永続データをファイルに書き込まれる前に、システムの古い.bootdataファイルの名前を.bootdata-oldに変更して、古い永続データのバックアップを行います。

ファイルは、設定されるランタイムシステムごとに作成されます。

システムの次回起動時に、.bootdataファイルは読み込まれられ、ランタイムシステムの永続変数はそのファイルからの値で初期化されます。

永続データを格納するファイル(.bootdata)が存在しない場合、永続データのこのバックアップファイル(.bootdata-old)がシステムの起動時に読み込まれます。これは例外的な処理ですが、UPSのないIPCに電源故障が発生してTwinCATが適切にシャットダウンできなかった場合などには起こり得ます。

WritePersistentDataファンクションブロックを使用して、PLCプログラムから永続データの保存を開始でき、永続データが保存されている最新の.bootdataファイルが利用できます。PORT入力パラメータは、永続データが保存されているランタイムシステムを指定します。

VAR_INPUT

```
VAR_INPUT
  NETID   : T_AmsNetId;
  PORT    : T_AmsPort;
  START   : BOOL;
  TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: ADSコマンドが実行されるTwinCATコンピュータのネットワークアドレス(T_AmsNetID型)。空の文字列をローカルコンピュータに入力できます。

PORT: 永続データが保存されるPLCランタイムシステムのADSポート番号(T_AmsPort型)。

START: ファンクションブロックは、この入力での立ち上がりエッジによって有効になります。

TMOUT: ADSコマンドを実行するときに、超過してはならないタイムアウト時間。

VAR_OUTPUT

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

BUSY: この出力はファンクションブロックの実行中にセットします。フィードバックを受信するまで、この出力はセットされたままです。

ERR: コマンドの送信中にADSエラーが発生した場合、BUSY出力がリセットされた後でこの出力をセットしません。

ERRID: ERR出力がセットされると、[ADSエラー番号 \[▶ 342\]](#)を返します。

例:

```
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;
```

付録 > [永続データを書き込む場合する場合のシステム動作 \[▶ 338\]](#)も参照してください。

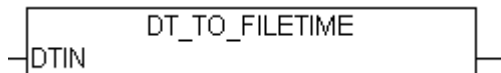
要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4 ファンクション

4.1 タイムファンクション

4.1.1 DT_TO_FILETIME



ファンクション「DT_TO_FILETIME」を使用して、DATE_AND_TIME (DT) 形式のPLC変数をFILETIME形式 (64ビット) に変換できます。

ファンクションDT_TO_FILETIME: [T_FILETIME](#) [[▶ 302](#)]

VAR_INPUT

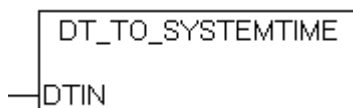
```
VAR_INPUT
  DTIN : DT;
END_VAR
```

DTIN: DATE_AND_TIME形式で変換する日付と時間。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.1.2 DT_TO_SYSTEMTIME



「DT_TO_SYSTEMTIME」ファンクションは、DATE_AND_TIME形式 (DT) のPLC変数をWindowsシステム時間構造体に変換します。システム時間は1ミリ秒の精度で、一方DATE_AND_TIMEの精度は1秒です。そのため、システム時間構造体の「wMilliseconds」変数は、常に、値ゼロを返します。

FUNCTION DT_TO_SYSTEMTIME: [Timestruct](#) [[▶ 306](#)]

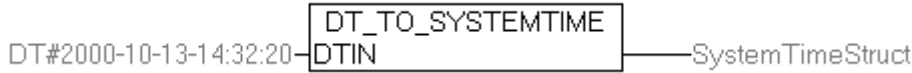
VAR_INPUT

```
VAR_INPUT
  DTIN : DT;
END_VAR
```

DTIN: DATE_AND_TIME形式で変換する日付と時間。

例:

```
PROGRAM SystemTimeTest
VAR
  SystemTimeStruct : Timestruct;
END_VAR
```



要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.1.3 F_GetDayOfMonthEx



このファンクションは、ある年のある月の1番目、2番目などの週の曜日の日付を計算します (例：2011年の1月の第2月曜日の日付)。

ファンクション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: 年 (1601~30827)。

wMonth: 月 (1~12)。

wWOM: 月の週 (1/ 5)。値1は第1週に相当し、2は第2週、および5は最後の週 (その月が5週ない場合でも)。

wDOW: 曜日 (0~6)。0 = 日曜日、1 = 月曜日~6 = 土曜日。

戻りパラメータ	説明
0	エラー、誤った、または無効なファンクションパラメータ
> 0	エラーはありません。月の特定の日

例:

例は、2011年8月の第2月曜日の日付を判定します。結果は8です。

```

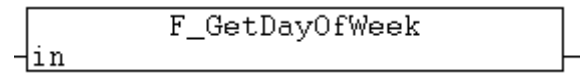
PROGRAM P_Dok_F_GetDayOfMonthEx
VAR
  wYear  : WORD := 2011;
  wMonth : WORD := 8;
  wWOM   : WORD (1..5) := 2; (* Week of month: 2 = Second week *)
  wDOW   : WORD (0..6) := 1; (* Day of week 1 = Monday *)
  wDay   : WORD; (* Day of month *)
END_VAR

wDay := F_GetDayOfMonthEx ( wYear, wMonth, wWOM, wDOW );
    
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.1.4 F_GetDayOfWeek



このファンクションは、DIN 1355 / ISO 8601に準拠して曜日を返します。この規格に準拠すると、曜日は次のような数字になります。月曜日 = 1、火曜日 = 2～日曜日 = 7。

ファンクションF_GetDayOfWeek: WORD

VAR_INPUT

```
VAR_INPUT
  in : DT;
END_VAR
```

in: 曜日の番号を判定する日付。

例:

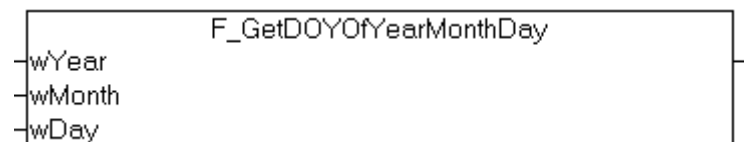
```
PROGRAM MAIN
VAR
  dtFirst : DT := DT#2008-01-01-00:00;
  dayOfWeek : WORD;
END_VAR
dayOfWeek := F_GetDayOfWeek(dtFirst);
```

結果は2 (火曜日)です。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.1.5 F_GetDOYOfYearMonthDay



このファンクションは、日付がその年の1月1日から何日目かを計算します。

ファンクションF_GetDOYOfYearMonthDay: WORD

VAR_INPUT

```
VAR_INPUT
  wYear : WORD;
  wMonth : WORD;
  wDay : WORD;
END_VAR
```

wYear: 年(0 ~ 2999)。

wMonth: 月(1 ~ 12)。

wDay: 日(1~31)。

戻りパラメータ	説明
0	エラー、誤ったwYear、wMonth、またはwDayパラメータ値
> 0	エラーはありません。年の日数(1 ~ 366)

例:

```
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 *)

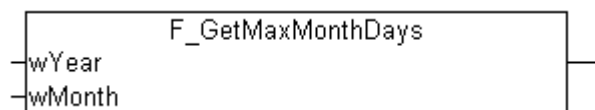
wYear := 2009;
wMonth := 2;
wDay := 1;
wDOY := F_GetDOYOfYearMonthDay( wYear, wMonth, wDay );(* wDOY = 32 *)

wYear := 2009;
wMonth := 3;
wDay := 1;
wDOY := F_GetDOYOfYearMonthDay( wYear, wMonth, wDay );(* wDOY = 60 *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.1.6 F_GetMaxMonthDays



このファンクションは、ある年のある月の最大日数を返します。

ファンクションF_GetMaxMonthDays: WORD

VAR_INPUT

```
VAR_INPUT
    wYear : WORD;
    wMonth : WORD;
END_VAR
```

wYear: 年。

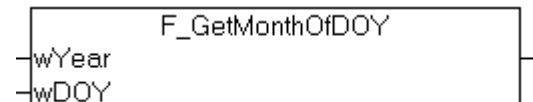
wMonth: 月(1~12)。

戻りパラメータ	説明
0	エラー、誤ったwMonthパラメータ値
> 0	エラーはありません。月の最大日数。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.1.7 F_GetMonthOfDOY



このファンクションは、年の日数値に基づいて月を計算します。

ファンクションF_GetMonthOfDOY: WORD

VAR_INPUT

```
VAR_INPUT
  wYear  : WORD;
  wDOY   : WORD;
END_VAR
```

wYear: 年(0 ~ 2999)。

wDOY: 月を判定する指定した年の日数値(1 ~ 366)。

戻りパラメータ	説明
0	エラー、誤ったwYear、またはwDOYパラメータ値。
> 0	エラーはありません。月(1 ~ 12)。

例:

```
PROGRAM P_TEST_DOY
VAR
  wYear  : WORD;
  wDOY   : WORD;
  wMonth : WORD;
END_VAR

wYear   := 2009;
wDOY    := 31;
wMonth  := F_GetMonthOfDOY( wYear, wDOY ); (* wMonth = 1 *)

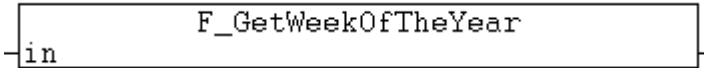
wYear   := 2009;
wDOY    := 32;
wMonth  := F_GetMonthOfDOY( wYear, wDOY ); (* wMonth = 2 *)

wYear   := 2009;
wDOY    := 60;
wMonth  := F_GetMonthOfDOY( wYear, wDOY ); (* wMonth = 3 *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.1.8 F_GetWeekOfTheYear



このファンクションは、DIN 1355 / ISO 8601規格に準拠して指定した日付の暦週数(週番号)を返します。

- ・ 最初の暦週は、**最低限、新年の4日間(DIN 1355 / ISO 8601)**を含む最初の週として定義されています。
- ・ 暦週は月曜日から始まります。各暦週は7日間です。
- ・ 最初の暦週の戻り値は、数字の1です。
- ・ 12月の29、30、および31日は、次の年の最初の暦週に属することがあります。
- ・ 1月1日、2日、および3日は前の年の最後の暦週に属することがあります。

ファンクションF_GetWeekOfTheYear: WORD

VAR_INPUT

```
VAR_INPUT
    in : DT;
END_VAR
```

in: その日の暦週を判定。

例:

```
PROGRAM MAIN
VAR
    dtNow      : DT := DT#2008-03-17-12:00;
    weekOfYear : WORD;
END_VAR
```

<code />

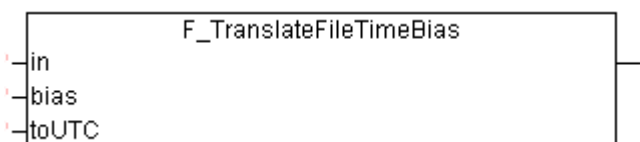
```
weekOfYear := F_GetWeekOfTheYear(dtNow);
```

結果は12です。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.1.9 F_TranslateFileTimeBias



このファンクションは、指定したバイアス時間移動に基づいて他のタイムゾーンに入力時間を変換します。たとえば、このファンクションを使用して、現地時間をUTC時間(協定世界時)に変換でき、その逆もできます。

ファンクション `F_TranslateFileTimeBias`: `T_FILETIME` [▶ 302]

VAR_INPUT

```
VAR_INPUT
  in      : T_FILETIME;
  bias    : DINT;
  toUTC   : BOOL;
END_VAR
```

in: 変換する入力時間 (`T_FILETIME` [▶ 302]型)。

bias: 分単位のUTC時間と現地時間の間の差 (正または負の値が許可されています)。

toUTC: このパラメータを使用して、入力時間を変換する方向を指定できます。

toUTC	方向	内部式
FALSE	UTC → 現地時間	現地時間 := UTC - バイアス
TRUE	現地時間 → UTC	UTC := 現地時間 + バイアス

例:

*in*変数は、変換する時間を格納します。*bToUTC*変数は、変換方向を決定します。*bToUTC* = TRUEの場合、現地時間はUTC時間に変換されます。*bToUTC* = FALSEの場合、UTC時間は現地時間に変換されます。*WEST_EUROPE_TZI*定数は、西ヨーロッパのタイムゾーン情報を格納します。必要なバイアス値は、定数のタイムゾーン情報と現在のbDST設定(夏時間)から計算されます。また、TwinCATシステムの現在のタイムゾーン情報は、ファンクションブロック `FB_GetTimeZoneInformation` [▶ 69] で判定できます。

Important notice: オンラインモードの視覚制御オプションのために、データ型DTが入力用に選択されました。他の時間の形式への変換は、変換関数が非常に演算集約型なので必ずしも推奨しません。

```
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 ) );
```

要件

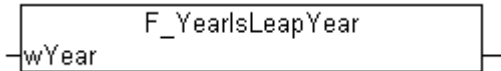
開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

これについて参照する

- 📖 `FB_TzSpecificLocalTimeToSystemTime` [▶ 108]
- 📖 `FB_TzSpecificLocalTimeToFileTime` [▶ 106]
- 📖 `FB_SystemTimeToTzSpecificLocalTime` [▶ 104]
- 📖 `FB_FileTimeToTzSpecificLocalTime` [▶ 52]
- 📖 `FB_GetTimeZoneInformation` [▶ 69]
- 📖 `FB_SetTimeZoneInformation` [▶ 101]

- NT_SetLocalTime [▶ 115]
- NT_GetTime [▶ 113]
- NT_SetTimeToRTCTime [▶ 116]
- FB_LocalSystemTime [▶ 81]

4.1.10 F_YearIsLeapYear



このファンクションは、ある年がうるう年かどうかを判定します。

ファンクションF_YearIsLeapYear: BOOL

VAR_INPUT

```
VAR_INPUT
    wYear : WORD;
END_VAR
```

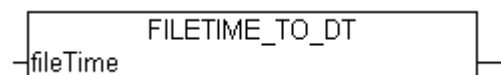
wYear: 年。

戻りパラメータ	説明
TRUE	切り替え年
FALSE	うるう年ではありません

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.1.11 FILETIME_TO_DT



ファンクション「FILETIME_TO_DT」は、FILETIME形式の時間をDATE_AND_TIME形式(DT)に変換します。DT形式はFILETIME形式よりも値範囲が小さく、秒精度のみを返します。このため、変換するFILETIME値は制限されます。最小許容値はDT#1970-01-01-00:00:00に相当し、最大許容値はDT#2106-02-06-06:28:15に相当します。

ファンクションFILETIME_TO_DT: DT

VAR_INPUT

```
VAR_INPUT
    fileTime : T_FILETIME;
END_VAR
```

fileTime: FILETIME形式に変換する時間(T_FILETIME [▶ 302]型)。

例:

```
PROGRAM MAIN
VAR
    fbSystemTime : GETSYSTEMTIME;
```

```

timeAsFileTime : T_FILETIME;
timeAsDT       : DT;
END_VAR

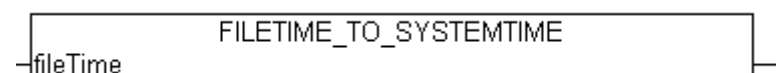
fbSystemTime( timeLoDW=>timeAsFileTime.dwLowDateTime, timeHiDW=>timeAsFileTime.dwHighDateTime );
timeAsDT := FILETIME_TO_DT( timeAsFileTime );

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.1.12 FILETIME_TO_SYSTEMTIME



ファンクション「FILETIME_TO_SYSTEMTIME」は、FILETIME形式の時間を「読み込み可能な」SYSTEMTIME形式に変換します。64ビットFILETIME変数の最上位ビットがセットされている場合、変換は失敗します。この場合、TIMESTRUCTメンバ変数の値はゼロです。

ファンクションFILETIME_TO_SYSTEMTIME: [TIMESTRUCT](#) [[▶ 306](#)]

VAR_INPUT

```

VAR_INPUT
fileTime : T_FILETIME;
END_VAR

```

fileTime: FILETIME形式に変換する時間 ([T_FILETIME](#) [[▶ 302](#)]型)。

例:

```

PROGRAM MAIN
VAR
fbSystemTime      : GETSYSTEMTIME;
timeAsFileTime    : T_FILETIME;
timeAsSystemTime  : TIMESTRUCT;
END_VAR

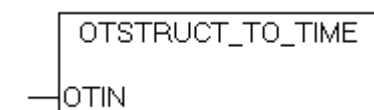
fbSystemTime( timeLoDW=>timeAsFileTime.dwLowDateTime, timeHiDW=>timeAsFileTime.dwHighDateTime );
timeAsSystemTime := FILETIME_TO_SYSTEMTIME( timeAsFileTime );

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.1.13 OTSTRUCT_TO_TIME



ファンクション「OTSTRUCT_TO_TIME」を使用して、分離されたミリ秒、秒、分、時間、日、週をもつ構造体をTIME変数に変換できます。

ファンクションOTSTRUCT_TO_TIME: TIME

VAR_INPUT

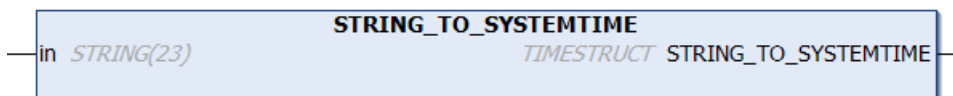
```
VAR_INPUT
    OTIN : OTSTRUCT;
END_VAR
```

OTIN: 変換する構造体 (OTSTRUCT [▶ 290]型)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.1.14 STRING_TO_SYSTEMTIME



ファンクションは、文字列をWindows SYSTEMTIME形式に変換します。

ファンクションSTRING_TO_SYSTEMTIME: Timestruct [▶ 306]

VAR_INPUT

```
VAR_INPUT
    in : STRING(23);
END_VAR
```

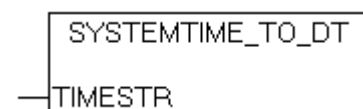
in: 変換する文字列。文字列は、以下のような「YYYY-MM-DD-hh:mm:ss.xxx」形式でなければいけません。

- ・ YYYY: 年 (1601~9999)
- ・ MM: 月 (01~12)
- ・ DD: 日 (01~31)
- ・ hh: 時間 (00~23)
- ・ mm: 分 (00~59)
- ・ ss: 秒 (00~59)
- ・ xxx: ミリ秒 (000~999)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.1.15 SYSTEMTIME_TO_DT



「SYSTEMTIME_TO_DT」ファンクションは、通常PLCでWindowsシステム時間構造体をDATE_AND_TIME形式(DT)に変換します。システム時間は1ミリ秒の精度で、一方DATE_AND_TIMEの精度は1秒です。システム時間のミリ秒は、返されたDATE_AND_TIME値の丸め方向を判定するために変換の最中に使用されます。丸めを無効にするには、Windowsシステム時間構造体のwMillisecondsエレメントをゼロにセットします。

ファンクションSYSTEMTIME_TO_DT: DT

VAR_INPUT

```
VAR_INPUT
    TIMESTR : Timestruct;
END_VAR
```

TIMESTR: 変換を必要とするWindowsシステム時間をもつ構造体 (Timestruct [▶ 306])。

例:

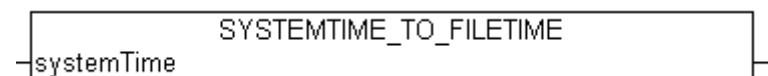
```
PROGRAM SystemTimeTest
VAR
    SystemTimeStruct : Timestruct;
    DTFromSystemTime : DT;
END_VAR
```



要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.1.16 SYSTEMTIME_TO_FILETIME



このファンクションを使用して、Windowsシステム時間構造体をFiletime形式に変換できます。システム時間変数の週wDayOfWeekの日は無視されます。システム時間の年は1601より大きく、30827未満です。

ファンクションSYSTEMTIME_TO_FILETIME: T_FILETIME [▶ 302]

VAR_INPUT

```
VAR_INPUT
    systemTime : Timestruct;
END_VAR
```

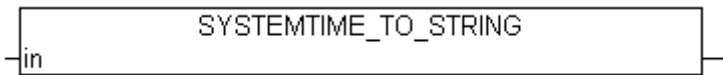
systemTime: 変換を必要とするWindowsシステム時間をもつ構造体 (Timestruct [▶ 306]型)。

戻りパラメータ	説明
0	エラー、誤ったシステム時間パラメータ値。
> 0	エラーはありません。ファイル時間。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.1.17 SYSTEMTIME_TO_STRING



このファンクションは、Windowsシステム時間構造体を以下の形式YYYY-MM-DD-hh:mm:ss.xxxをもつ文字列に変換します。

- ・ YYYY: 年 (1601~9999)
- ・ MM: 月 (01~12)
- ・ DD: 日 (01~31)
- ・ hh: 時間 (00~23)
- ・ mm: 分 (00~59)
- ・ ss: 秒 (00~59)
- ・ xxx: ミリ秒 (000~999)

ファンクションSYSTEMTIME_TO_STRING: STRING(24)

VAR_INPUT

```

VAR_INPUT
  in : TIMESTRUCT;
END_VAR
  
```

in: 変換を必要とするWindowsシステム時間をもつ構造体 (TIMESTRUCT [▶_306]型)。

例:

```

PROGRAM MAIN
VAR
  fbGetSystemTime : GETSYSTEMTIME;
  fileTime        : T_FILETIME;
  sTime           : STRING;
END_VAR
  
```

<code />

```

fbGetSystemTime(timeLoDW=>fileTime.dwLowDateTime, timeHiDW=>fileTime.dwHighDateTime );
sTime := SYSTEMTIME_TO_STRING( FILETIME_TO_SYSTEMTIME( fileTime ) );
  
```

オンラインビュー:

Expression	Type	Value	Prepared value	Address	Comment
fbGetSystemTime	GETSYSTEMTIME				
fileTime	T_FILETIME				
sTime	STRING	'2014-09-22-11:16:48.935'			

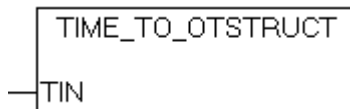
```

1 fbGetSystemTime(timeLoDW[16#83D07B70]->fileTime.dwLowDateTime[16#83D07B70], timeHiDW[16#01CFD656]->fileTime.dwHighDateTime[16#01CFD656]);
2 sTime[2014-09-22] := SYSTEMTIME_TO_STRING( FILETIME_TO_SYSTEMTIME( fileTime ) );
3 RETURN
  
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.1.18 TIME_TO_OTSTRUCT



ファンクション「TIME_TO_OTSTRUCT」を使用して、分離したミリ秒、秒、分、時間、日、週をもつ構造体に TIME定数または変数を変換します。

ファンクションTIME_TO_OTSTRUCT: [OTSTRUCT](#) [[▶ 290](#)]

VAR_INPUT

```
VAR_INPUT
    TIN : TIME;
END_VAR
```

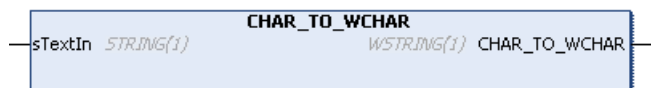
TIN: 変換するTIME変数。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.2 拡張STRINGファンクション

4.2.1 CHAR_TO_WCHAR



このファンクションは、データ型STRINGの変数をデータ型WSTRINGの変数に変換します(null終端付き)。

ファンクションCHAR_TO_WCHAR : WSTRING(1)

VAR_INPUT

```
VAR_INPUT
    sTextIn : STRING(1);
END_VAR
```

sTextIn: 変換するSTRING変数。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)、 3.3.21.0以降

4.2.2 CONCAT2



このファンクションは、任意の長さのデータ型STRINGの2つの文字列を連結し、生成される文字列が指定した出力文字列より長いかどうかをチェックします。長い場合、文字列は切り捨てられます。

このファンクションは、

- ・ 連結が成功した場合にTRUEを返します。
- ・ 生成される文字列が出力文字列より長く、所定の出力バッファに収まらない場合、FALSEを返します。その場合、生成される文字列に必要なメモリ容量は出力文字列に必要なメモリ容量よりも大きくなります。そのため、文字列は切り捨てられます。

ファンクションは、無限ループを避けるために10,000文字 (ParameterList.cMaxCharacters) 以降の変換を停止します。

ファンクションCONCAT2 : BOOL

VAR_INPUT

```
VAR_INPUT
  pSrcString1 : POINTER TO STRING;
  pSrcString2 : POINTER TO STRING;
  pDstString  : POINTER TO STRING;
  nDstSize   : UDINT;
END_VAR
```

pSrcString1: 連結される1番目のSTRING変数へのポインタ (入力文字列)。

pSrcString2: 連結される2番目のSTRING変数へのポインタ (入力文字列)。

pDstString: 連結後に生成されるSTRING変数へのポインタ (出力文字列)

nDstSize: 生成されるSTRING変数のサイズ (バイト単位) (出力文字列)。割り当てには、演算子SIZEOF () を使用できます。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.21.0以降

4.2.3 DELETE2



このファンクションは、文字列を位置nPosから開始してnLen文字分を削除します。

このファンクションは、

- ・ 文字の削除に成功した場合にTRUEを返します。
- ・ 生成される文字列が出力文字列より長く、所定の出力バッファに収まらない場合、FALSEを返します。その場合、生成される文字列に必要なメモリ容量は出力文字列に必要なメモリ容量よりも大きくなります。そのため、文字列は切り捨てられます。

このファンクションは、無限ループを避けるために10,000文字 (ParameterList.cMaxCharacters) 以降の変換を停止します。

ファンクションDELETE2 : BOOL

VAR_INPUT

```

VAR_INPUT
  pSrcString : POINTER TO STRING;
  pDstString : POINTER TO STRING;
  nDstSize   : UDINT;
  nLen       : UDINT;
  nPos       : UDINT;
END_VAR

```

pSrcString: STRING変数へのポインタ (入力文字列)

pDstString: 生成されるSTRING変数へのポインタ (出力文字列)

nDstSize: 生成されるSTRING変数のサイズ(バイト単位) (出力文字列)。割り当てには、演算子SIZEOF ()を使用できます。

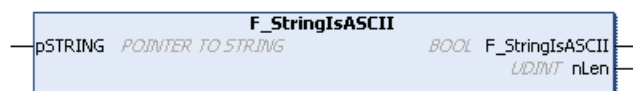
nLen: 削除する文字数

nPos: 削除する先頭文字の位置。それに続く文字も削除されます (nPos = 1 = 先頭文字)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.35.0以降

4.2.4 F_StringIsASCII



このファンクションは文字列がASCII文字 (0x000~0x7F) のみを格納しているかどうかチェックし、ASCII文字数を返します。文字列がASCII文字のみを格納している場合、文字列は直接UTF-8と互換性があります。

ファンクションは、無限ループを避けるために10,000文字 (Parameter list. cMaxCharacters) 以降の変換を停止します。

ファンクションF_StringIsASCII : BOOL

文字列がASCII文字のみを格納している場合、戻し値はTRUEです。

VAR_INPUT

```

VAR_INPUT
  pSTRING : POINTER TO STRING;
END_VAR

```

pString: STRING変数へのポインタ。

VAR_OUTPUT

```

VAR_OUTPUT
  nLen : UDINT;
END_VAR

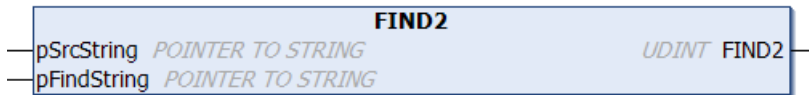
```

nLen: 文字列のASCII文字数

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.21.0以降

4.2.5 FIND2



このファンクションは、他の文字列で一度以上出現する文字列を検出します。

このファンクションは

- ・ 検出された最初の文字列の先頭文字の位置を返します。
- ・ 文字列が検出されなかった場合、値ゼロを返します。

このファンクションは、無限ループを避けるために10,000文字 (ParameterList.cMaxCharacters) 以降の変換を停止します。

FUNCTION FIND2 : UDINT

VAR_INPUT

```
VAR_INPUT
    pSrcString : POINTER TO STRING;
    pFindString : POINTER TO STRING;
END_VAR
```

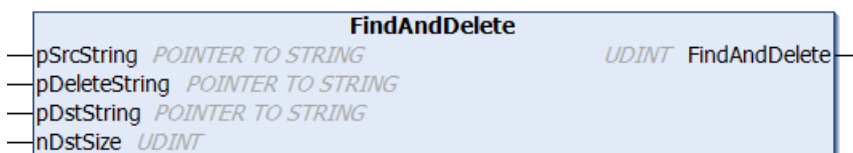
pSrcString: 文字列を検索するSTRING変数へのポインタ

pFindString: 文字列が検索中であるSTRING変数へのポインタ

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.35.0以降

4.2.6 FindAndDelete



このファンクションは、他の文字列で1回以上出現する文字列を検出し、削除します。

このファンクションは

- ・ 削除した文字列数を返します。
- ・ 文字列が検出されなかった場合、値ゼロを返します。

このファンクションは、無限ループを避けるために10,000文字 (ParameterList.cMaxCharacters) 以降の変換を停止します。

ファンクションFindAndDelete : UDINT

VAR_INPUT

```

VAR_INPUT
  pSrcString      : POINTER TO STRING;
  pDeleteString   : POINTER TO STRING;
  pDstString      : POINTER TO STRING;
  nDstSize        : UDINT;
END_VAR

```

pSrcString: 文字列を検索するSTRING変数へのポインタ (入力文字列)

pDeleteString: 文字列が検索され、削除されるSTRING変数へのポインタ (入力文字列)

pDstString: 生成されるSTRING変数へのポインタ (出力文字列)

nDstSize: 生成されるSTRING変数のサイズ(バイト単位) (出力文字列)。割り当てには、演算子SIZEOF()を使用できます。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.35.0以降

4.2.7 FindAndDeleteChar

```

FindAndDeleteChar
pSrcString POINTER TO STRING UDINT FindAndDeleteChar
sDeleteChar STRING(1)
pDstString POINTER TO STRING
nDstSize UDINT

```

このファンクションは、文字列で1回以上出現する文字を検出し、削除します。

このファンクションは

- ・ 削除した文字数を返します。
- ・ 文字が検出されなかった場合、値ゼロを返します。

このファンクションは、無限ループを避けるために10,000文字(ParameterList.cMaxCharacters)以降の変換を停止します。

ファンクションFindAndDeleteChar : UDINT

VAR_INPUT

```

VAR_INPUT
  pSrcString      : POINTER TO STRING;
  sDeleteChar     : STRING(1);
  pDstString      : POINTER TO STRING;
  nDstSize        : UDINT;
END_VAR

```

pSrcString: 文字列を検索するSTRING変数へのポインタ (入力文字列)

sDeleteChar: 削除する文字

pDstString: 生成されるSTRING変数へのポインタ (出力文字列)

nDstSize: 生成されるSTRING変数のサイズ(バイト単位) (出力文字列)。割り当てには、演算子SIZEOF()を使用できます。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)、 3.3.35.0以降

4.2.8 FindAndReplace

FindAndReplace	
pSrcString <i>POINTER TO STRING</i>	UDINT FindAndReplace
pDeleteString <i>POINTER TO STRING</i>	
pInsertString <i>POINTER TO STRING</i>	
pDstString <i>POINTER TO STRING</i>	
nDstSize <i>UDINT</i>	

このファンクションは、他の文字列で1回以上出現する文字列を検出し、他の文字列と置換します。

このファンクションは

- ・ 置換した文字列数を返します。
- ・ 文字列が検出されなかった場合、値ゼロを返します。

このファンクションは、無限ループを避けるために10,000文字 (Parameter list.cMaxCharacters) 以降の変換を停止します。

ファンクション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: 文字列を検索するSTRING変数へのポインタ (入力文字列)

pDeleteString: 文字列が置換されるSTRING変数へのポインタ (入力文字列)

pInsertString: 文字列が他の文字列を置換するSTRING変数へのポインタ (入力文字列)

pDstString: 生成されるSTRING変数へのポインタ (出力文字列)

nDstSize: 生成されるSTRING変数のサイズ(バイト単位) (出力文字列)。割り当てには、演算子SIZEOF()を使用できます。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)、 3.3.35.0以降

4.2.9 FindAndReplaceChar

FindAndReplaceChar		
pSrcString	POINTER TO STRING	UDINT FindAndReplaceChar
sDeleteChar	STRING(1)	
sInsertChar	STRING(1)	
pDstString	POINTER TO STRING	
nDstSize	UDINT	

ファンクションは、文字列で1回以上出現する文字を検出し、他の文字と置換します。

このファンクションは

- ・ 置換した文字数を返します。
- ・ 文字が検出されなかった場合、値ゼロを返します。

このファンクションは、無限ループを避けるために10,000文字 (Parameter list. cMaxCharacters) 以降の変換を停止します。

ファンクション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: 文字列を検索するSTRING変数へのポインタ (入力文字列)

sDeleteChar: 置換される文字

sInsertChar: 他の文字を置換する文字

pDstString: 生成されるSTRING変数へのポインタ (出力文字列)

nDstSize: 生成されるSTRING変数のサイズ(バイト単位) (出力文字列)。割り当てには、演算子SIZEOF()を使用できます。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.35.0以降

4.2.10 INSERT2

INSERT2		
pSrcString	POINTER TO STRING	BOOL INSERT2
pInsertString	POINTER TO STRING	
pDstString	POINTER TO STRING	
nDstSize	UDINT	
nPos	UDINT	

ファンクションは、文字列を他の文字列の位置nPosの後に挿入します。nPos = 0の場合、他の文字列の先頭文字の前に挿入されます。

このファンクションは

- ・ 文字列の挿入に成功した場合にTRUEを返します。

- ・ 生成される文字列が出力文字列より長く、所定の出力バッファに収まらない場合、FALSEを返します。その場合、生成される文字列に必要なメモリ容量は出力文字列に必要なメモリ容量よりも大きくなります。そのため、文字列は切り捨てられます。

このファンクションは、無限ループを避けるために10,000文字 (Parameter list. cMaxCharacters) 以降の変換を停止します。

ファンクション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: 文字列を検索するSTRING変数へのポインタ (入力文字列)

pInsertString: 文字列が他の文字列に挿入されるSTRING変数へのポインタ (入力文字列)

pDstString: 生成されるSTRING変数へのポインタ (出力文字列)

nDstSize: 生成されるSTRING変数のサイズ(バイト単位) (出力文字列)。割り当てには、演算子sizeof()を使用できます。

nPos: その後に文字列が挿入される文字の位置

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.35.0以降

4.2.11 LEN2



ファンクションは、文字列の文字数を返します (STRING長)。

ファンクションは、無限ループを避けるために10,000文字 (Parameter list. cMaxCharacters) 以降の変換を停止します。

ファンクションLEN2 : UDINT

VAR_INPUT

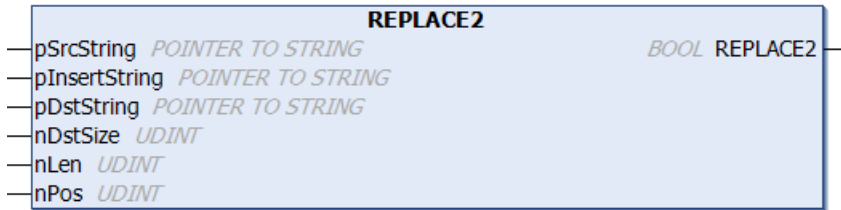
```
VAR_INPUT
    pSTRING : POINTER TO STRING;
END_VAR
```

pString: STRING変数へのポインタ

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.21.0以降

4. 2. 12 REPLACE2



ファンクションは、文字列のnLen文字を他の文字列の位置nPosから始まる文字と置換します。

このファンクションは

- ・ 文字の交換に成功した場合にTRUEを返します。
- ・ 生成される文字列が出力文字列より長く、所定の出力バッファに収まらない場合、FALSEを返します。その場合、生成される文字列に必要なメモリ容量は出力文字列に必要なメモリ容量よりも大きくなります。そのため、文字列は切り捨てられます。

このファンクションは、無限ループを避けるために10,000文字 (Parameter list. cMaxCharacters) 以降の変換を停止します。

ファンクション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: 文字列を検索するSTRING変数へのポインタ (入力文字列)

pInsertString: 文字列が文字を置換するSTRING変数へのポインタ (入力文字列)

pDstString: 生成されるSTRING変数へのポインタ (出力文字列)

nDstSize: 生成されるSTRING変数のサイズ(バイト単位) (出力文字列)。割り当てには、演算子sizeof()を使用できます。

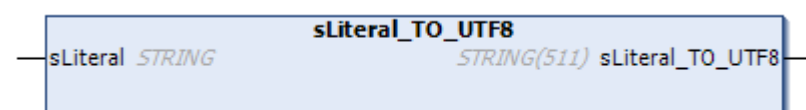
nLen: 置換される文字数

nPos: 削除する文字の位置。その後続く文字も削除されます (nPos = 1 = 先頭文字)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)、 3.3.35.0以降

4. 2. 13 sLiteral_TO_UTF8



このファンクションは、データ型STRINGの任意の文字列をUTF-8形式の文字列に変換します。このファンクションは、リテラルの割り当てに特に適しています。

リテラルをUTF-8文字列に割り当てるときのルールは以下のようになります。

- ・ ASCII文字のみを使用するリテラルは、直接割り当てることができます。
- ・ STRING文字セットを使用するリテラルは、`sLiteral_TO_UTF8()`によって割り当てできます。
- ・ WSTRING文字セットを使用するリテラルは、`wsLiteral_TO_UTF8()` [▶ 166]によって割り当てできません。

リテラルが使用可能な出力文字列より長い場合、空の文字列を返します。

ファンクション `sLiteral_TO_UTF8` : STRING(511)

VAR_INPUT

```
VAR_IN_OUT CONSTANT
    sLiteral    : STRING;
END_VAR
```

sLiteral: 変換するSTRING文字列。

例

```
{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 €.);
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.34.0以降

4.2.14 STRING_TO_UTF8

```
STRING_TO_UTF8
pDstUTF8  PVOID          BOOL STRING_TO_UTF8
pSrcSTRING POINTER TO STRING
nDstSize  UDINT
```

このファンクションは、データ型STRINGの変数の任意の文字列をUTF-8形式の文字列に変換します。

このファンクションは

- ・ 変換できた場合にTRUEを返します。
- ・ 特定の文字セットのために変換できなかった場合は、FALSEを返します。

入力文字列が出力文字列より長い場合、文字列は切り捨てられます。入力文字列が長すぎる場合、出力文字列へのコード化はできません。UTF-8への変換では、出力文字列に必要なメモリ容量は入力文字列に必要なメモリ容量よりも大きいことがあります。

ファンクションは、無限ループを避けるために10,000文字以降の変換を停止します (Parameter list. cMaxCharacters)。

ファンクション `STRING_TO_UTF8` : BOOL

VAR_INPUT

```
VAR_INPUT
    pDstUTF8    : PVOID;
    pSrcSTRING  : POINTER TO STRING;
    nDstSize    : UDINT;
END_VAR
```


pDstUTF8: UTF-8形式の文字列へのポインタ (出力文字列)

pSrcSTRING: 変換するSTRING変数へのポインタ (入力文字列)

nDstSize: 生成される変数のサイズ(バイト単位) (出力文字列)。割り当てには、演算子SIZEOF()を使用できます。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)、 3.3.21.0以降

4.2.15 STRING_TO_WSTRING2



このファンクションはデータ型STRINGの変数をデータ型WSTRINGの変数に変換し、入力文字列が出力文字列より長いかどうかをチェックします。長い場合、文字列は切り捨てられます。

このファンクションは

- ・ 文字列のすべてを変換できた場合にTRUEを返します。
- ・ 入力文字列が出力文字列より長く、所定の出力バッファに収まらない場合、FALSEを返します。出力文字列に必要なメモリ容量は入力文字列に必要なメモリ容量よりも大きくなります。そのため、文字列は切り捨てられます。

ファンクションは、無限ループを避けるために10,000文字以降の変換を停止します (Parameter list. cMaxCharacters)。

ファンクションSTRING_TO_WSTRING2 : BOOL

VAR_INPUT

```

VAR_INPUT
  pDstWSTRING : POINTER TO WSTRING;
  pSrcSTRING  : POINTER TO STRING;
  nDstSize    : UDINT;
END_VAR

```

pDstWSTRING: 変換するWSTRING変数へのポインタ (出力文字列)

pSrcSTRING: 変換するSTRING変数へのポインタ (入力文字列)

nDstSize: 生成されるWSTRING変数のサイズ(バイト単位) (出力文字列)。割り当てには、演算子SIZEOF()を使用できます。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)、 3.3.21.0以降

4.2.16 STRNCPY



ファンクションは、データ型STRINGの変数の文字列をコピーし、文字列をすべてコピーしたかどうかをチェックします。

このファンクションは

- ・ すべての文字列 (ソース配列の内容) をコピーできた場合にTRUEを返します。
- ・ コピー時に文字列が切り捨てられた場合、FALSEを返します。入力文字列が出力文字列よりも長い場合、出力文字列長に相当する文字数 (null 終端を含む) のみがコピーされます。

ファンクションは、無限ループを避けるために10,000文字以降 (Parameter list. cMaxCharacters) で停止します。(Parameter list. cMaxCharacters)。

ファンクションSTRNCOPY : BOOL

VAR_INPUT

```
VAR_INPUT
    pDst      : POINTER TO STRING;
    pSrc      : POINTER TO STRING;
    nDstSize  : UDINT;
END_VAR
```

pDst: コピーされたSTRING変数へのポインタ (入力文字列)

pSrc: コピーされるSTRING変数へのポインタ (出力文字列)

nDstSize: 生成されるSTRING変数のサイズ (バイト単位) (出力文字列)。割り当てには、演算子sizeof () を使用できます。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.21.0以降

4.2.17 UTF8_TO_STRING



ファンクションは、UTF8形式の文字列 (データ型PVOIDのポインタ変数) をデータ型STRINGの文字列 (変数) に変換します。

このファンクションは

- ・ 変換できた場合にTRUEを返します。
- ・ 特定の文字セットのために変換できなかった場合は、FALSEを返します。

入力文字列が出力文字列より長い場合、文字列は切り捨てられます。未知の文字はスキップされます。

ファンクションは、無限ループを避けるために10000文字以降の変換を停止します (Parameter list. cMaxCharacters)。

ファンクションUTF8_TO_STRING : BOOL

変換が成功した場合、戻り値はTRUEです。

VAR_INPUT

```
VAR_INPUT
  pDstSTRING : POINTER TO STRING;
  pSrcUTF8   : PVOID;
  nDstSize   : UDINT;
END_VAR
```

pDstSTRING: 変換されたSTRING変数へのポインタ (出力文字列)

pSrcUTF8: ポインタ変数 (入力文字列)

nDstSize: 生成されるSTRING変数のサイズ (バイト単位) (出力文字列)。割り当てには、演算子SIZEOF ()を使用できます。

VAR_OUTPUT

```
VAR_OUTPUT
  nDstLen      : UDINT;
END_VAR
```

nDstLen: 文字数での出力文字列の実際長。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.21.0以降

4.2.18 UTF8_TO_WSTRING

ファンクションは、UTF-8形式の文字列をデータ型WSTRINGの文字列 (変数) に変換します。

このファンクションは

- ・ 変換できた場合にTRUEを返します。
- ・ 特定の文字セットのために変換できなかった場合は、FALSEを返します。

入力文字列が出力文字列より長い場合、文字列は切り捨てられます。入力文字列が長すぎる場合、出力文字列へのコード化はできません。UTF-8への変換では、出力文字列に必要なメモリ容量は入力文字列に必要なメモリ容量よりも大きいことがあります。未知の文字はスキップされます。

ファンクションは、無限ループを避けるために10000文字以降の変換を停止します (Parameter list. cMaxCharacters)。

ファンクションFUNCTION UTF8_TO_WSTRING : BOOL

変換が成功した場合、戻り値はTRUEです。

VAR_INPUT

```
VAR_INPUT
  pDstWSTRING : POINTER TO WSTRING;
  pSrcUTF8   : PVOID;
  nDstSize   : UDINT;
END_VAR
```

pDstWSTRING: 変換するWSTRING変数へのポインタ (出力文字列)

pSrcUTF8: ポインタ変数 (入力文字列)

nDstSize: 生成されるWSTRING変数のサイズ(バイト単位) (出力文字列)。割り当てには、演算子SIZEOF()を使用できます。

VAR_OUTPUT

```
VAR_OUTPUT
    nDstLen : UDINT;
END_VAR
```

nDstLen: 文字数での出力文字列の実際長。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)、 3.3.21.0以降

4.2.19 UTF8Len



ファンクションは、UTF-8文字列の文字数を返します。

文字列がUTF-8形式に対応していない場合、ファンクションは値ゼロを返します。

さらに、ファンクションはすべての文字が有効なASCII文字かどうかをチェックし、結果をbASCII出力で出力します。

ファンクションは、無限ループを避けるために10000文字以降のチェックを停止します (Parameter list. cMaxCharacters)。

ファンクションUTF8Len : UDINT

戻り値は、UTF-8文字列で文字数を返します。

VAR_INPUT

```
VAR_INPUT
    pUTF8 : PVOID;
END_VAR
```

pUTF8: null終端UTF-8文字列へのポインタ

VAR_OUTPUT

```
VAR_OUTPUT
    bASCII : BOOL;
    nSize : UDINT;
END_VAR
```

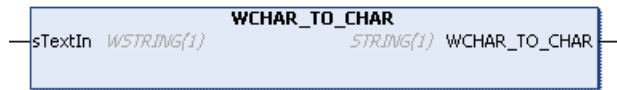
bASCII: UTF-8文字が有効なASCII文字の場合、TRUEです。

nSize: バイト単位の文字列のサイズ(ゼロ終端なし)。文字によって、バイト単位のサイズが文字列長よりも大きいことがあります。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)、 3.3.21.0以降

4. 2. 20 WCHAR_TO_CHAR



このファンクションは、データ型WSTRINGの変数をデータ型STRINGの変数(null終端付き)に変換します。変換は、WSTRING文字がSTRING文字に相当する場合のみ可能です。そうでない場合、文字が返されません。

ファンクションWCHAR_TO_CHAR : STRING(1)

VAR_INPUT

```
VAR_INPUT
    sTextIn : WSTRING(1);
END_VAR
```

sTextIn: 変換するWSTRING変数。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)、 3.3.21.0以降

4. 2. 21 WCONCAT2



このファンクションは、任意の長さのデータ型WSTRINGの2つの文字列を連結し、生成される文字列が指定した出力文字列より長いかどうかをチェックします。長い場合、文字列は切り捨てられます。

このファンクションは

- ・ 連結が成功した場合にTRUEを返します。
- ・ 生成される文字列が出力文字列より長く、所定の出力バッファに収まらない場合、FALSEを返します。その場合、生成される文字列に必要なメモリ容量は出力文字列に必要なメモリ容量よりも大きくなります。そのため、文字列は切り捨てられます。

ファンクションは、無限ループを避けるために10,000文字(Parameter list. cMaxCharacters)以降の変換を停止します。

ファンクションWCONCAT2 : BOOL

VAR_INPUT

```
VAR_INPUT
    pSrcWString1 : POINTER TO WSTRING;
    pSrcWString2 : POINTER TO WSTRING;
    pDstWString  : POINTER TO WSTRING;
    nDstSize     : UDINT;
END_VAR
```

pSrcWString1: 連結される最初のWSTRING変数へのポインタ(入力文字列)

pSrcWString2: 連結される2番目のWSTRING変数へのポインタ(入力文字列)

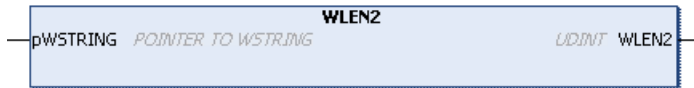
pDstString: 連結後に生成されるWSTRING変数へのポインタ(出力文字列)

nDstSize: 生成されるWSTRING変数のサイズ(バイト単位) (出力文字列)。割り当てには、演算子SIZEOF()を使用できます。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)、3.3.21.0以降

4.2.22 WLEN2



ファンクションは、データ型WSTRINGのUnicode文字列の文字数を返します(WSTRINGの長さ)。

ファンクションは、無限ループを避けるために10,000文字(Parameterlist.cMaxCharacters)以降の変換を停止します。

ファンクションWLEN2 : UDINT

VAR_INPUT

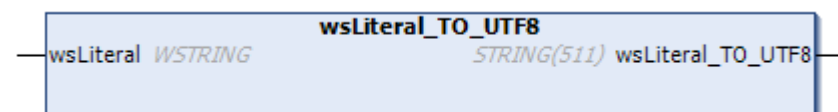
```
VAR_INPUT
    pWSTRING : POINTER TO WSTRING;
END_VAR
```

pWSTRING: WSTRING変数へのポインタ

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)、3.3.21.0以降

4.2.23 wsLiteral_TO_UTF8



このファンクションは、データ型WSTRINGの任意の文字列をUTF-8形式の文字列に変換します。このファンクションは、リテラルの割り当てに特に適しています。

リテラルをUTF-8文字列に割り当てるときのルールは以下ようになります。

- ・ ASCII文字のみを使用するリテラルは、直接割り当てることができます。
- ・ STRING文字セットを使用するリテラルは、[sLiteral_TO_UTF8\(\)](#) [▶ 159]によって割り当てできます。
- ・ WSTRING文字セットを使用するリテラルは、[wsLiteral_TO_UTF8\(\)](#)によって割り当てできます。

リテラルが使用可能な出力文字列より長い場合、空の文字列を返します。

ファンクション `wsLiteral_TO_UTF8` : `STRING(511)`

VAR_INPUT

```
VAR_IN_OUT CONSTANT
    wsLiteral    : WSTRING;
END_VAR
```

sLiteral: 変換するWSTRING文字列。

例

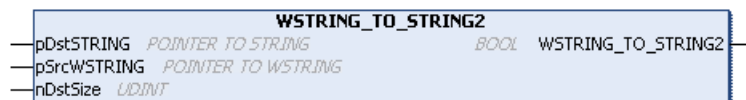
```
{attribute 'TcEncoding' := 'UTF-8'}
sMyText : STRING := wsLiteral_TO_UTF8("Hühner legen Eier.");

{attribute 'TcEncoding' := 'UTF-8'}
sMyText2 : STRING := wsLiteral_TO_UTF8("The dinner costs 30 €.");
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.34.0以降

4.2.24 WSTRING_TO_STRING2



ファンクションは、データ型WSTRINGの変数をデータ型STRINGの変数に変換します。

このファンクションは

- ・ 文字列のすべてを変換できた場合にTRUEを返します。
- ・ 入力文字列が出力文字列より長く、所定の出力バッファに収まらない場合、FALSEを返します。

変換できない文字は、変換中にスキップされます。

ファンクションは、無限ループを避けるために10,000文字以降の変換を停止します (Parameter list. cMaxCharacters)。

ファンクション `WSTRING_TO_STRING2` : `BOOL`

VAR_INPUT

```
VAR_INPUT
    pDstString : POINTER TO STRING;
    pSrcWString : POINTER TO WSTRING;
    nDstSize   : UDINT;
END_VAR
```

pDstSTRING: 変換されたSTRING変数へのポインタ (出力文字列)

pSrcWSTRING: 変換するWSTRING 変数へのポインタ (入力文字列)

nDstSize: 生成されるSTRING変数のサイズ(バイト単位) (出力文字列)。割り当てには、演算子 `sizeof()` を使用できます。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.21.0以降

4. 2. 25 WSTRING_TO_UTF8

```

WSTRING_TO_UTF8
pDstUTF8  PVOID          BOOL  WSTRING_TO_UTF8
pSrcWSTRING  POINTER TO WSTRING
nDstSize  UDINT

```

ファンクションは、データ型WSTRINGの変数の文字列をUTF-8形式の文字列に変換します。

このファンクションは

- ・ 変換できた場合にTRUEを返します。
- ・ 特定の文字セットのために変換できなかった場合は、FALSEを返します。

入力文字列が出力文字列より長い場合、文字列は切り捨てられます。入力文字列が長すぎる場合、出力文字列へのコード化はできません。(UTF-8への変換では、出力文字列に必要なメモリ容量は入力文字列に必要なメモリ容量よりも大きいことがあります)。未知の文字はスキップされます。

ファンクションは、無限ループを避けるために10,000文字以降の変換を停止します (Parameter list. cMaxCharacters)。

ファンクションWSTRING_TO_UTF8 : BOOL

VAR_INPUT

```

VAR_INPUT
  pDstUTF8      : PVOID;
  pSrcWSTRING   : POINTER TO WSTRING;
  nDstSize      : UDINT;
END_VAR

```

pDstUTF8: ポインタ変数(出力文字列)

pSrcWSTRING: WSTRING変数へのポインタ(入力文字列)

nDstSize: 生成される変数のサイズ(バイト単位)(出力文字列)。割り当てには、演算子sizeof()を使用できます。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.21.0以降

4. 2. 26 WSTRNCOPY

```

WSTRNCOPY
pDst  POINTER TO WSTRING  BOOL  WSTRNCOPY
pSrc  POINTER TO WSTRING
nDstSize  UDINT

```

ファンクションは、データ型WSTRINGの変数の文字列をコピーし、文字列をすべてコピーしたかどうかをチェックします。

このファンクションは

- ・ すべての文字列(ソース配列の内容)をコピーできた場合にTRUEを返します。
- ・ コピー時に文字列が切り捨てられた場合、FALSEを返します。入力文字列が出力文字列よりも長い場合、出力文字列長に相当する文字数(null終端を含む)のみがコピーされます。

ファンクションは、無限ループを避けるために10,000文字以降 (Parameter list. cMaxCharacters) で停止します。

ファンクションWSTRNCPY : BOOL

VAR_INPUT

```
VAR_INPUT
  pDst      : POINTER TO WSTRING;
  pSrc      : POINTER TO WSTRING;
  nDstSize  : UDINT;
END_VAR
```

pDst: コピーされたWSTRING変数へのポインタ (入力文字列)

pSrc: コピーされるWSTRING変数へのポインタ (出力文字列)

nDstSize: 生成されるWSTRING変数のサイズ (バイト単位) (出力文字列)。割り当てには、演算子SIZEOF () を使用できます。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.21.0以降

4.3 バイトオーダー変換機能

4.3.1 ホストバイトオーダー/ネットワークバイトオーダー

バイトオーダーはネットワークプロトコルで決まっています。これは、ネットワークバイトオーダーと呼ばれています。TwinCATシステムの通常のバイトオーダーは、ホストバイトオーダーと呼ばれます。多くの場合、必要なネットワークバイトオーダーはビッグエンディアン形式 (MOTOROLA) に相当します。ただし、TwinCAT PLCシステムはリトルエンディアン形式 (Intel) を使用します。TwinCAT PLCシステムと異種のプラットフォームの間でエラーがないデータ交換を行うために、アプリケーションプログラムのバイトオーダーを適切に変換する必要があります。

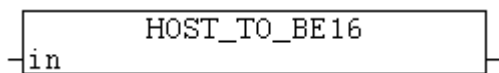
TwinCATシステム (ホスト) から外部システムへネットワークプロトコルで送信されるデータは、以下のファンクションを使用してネットワーク形式に変換できます。

- ・ [HOST TO BE16 \[▶ 170\]](#)
- ・ [HOST TO BE32 \[▶ 170\]](#)
- ・ [HOST TO BE64 \[▶ 170\]](#)
- ・ [HOST TO BE64EX \[▶ 171\]](#)
- ・ [HOST TO BE128 \[▶ 171\]](#)

逆に、受信したネットワークデータ (外部システム) は、以下のファンクションを使用してホスト形式 (TwinCATシステム) に変換できます。

- ・ [BE16 TO HOST \[▶ 172\]](#)
- ・ [BE32 TO HOST \[▶ 172\]](#)
- ・ [BE64 TO HOST \[▶ 173\]](#)
- ・ [BE64 TO HOSTEX \[▶ 173\]](#)
- ・ [BE128 TO HOST \[▶ 173\]](#)

4.3.2 HOST_TO_BE16



ファンクションは、16ビット数のホストバイトオーダーからネットワークバイトオーダーへの変換を実行します。[バイトオーダー](#) [[▶ 169](#)]も参照してください。

ファンクションHOST_TO_BE16: WORD

VAR_INPUT

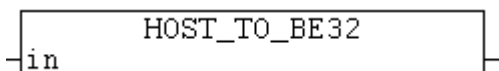
```
VAR_INPUT
    in : WORD;
END_VAR
```

in: 変換する数。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.3.3 HOST_TO_BE32



ファンクションは、32ビット数のホストバイトオーダーからネットワークバイトオーダーへの変換を実行します。[バイトオーダー](#) [[▶ 169](#)]も参照してください。

ファンクションHOST_TO_BE32: DWORD

VAR_INPUT

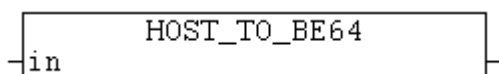
```
VAR_INPUT
    in : DWORD;
END_VAR
```

in: 変換する数。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.3.4 HOST_TO_BE64



ファンクションは、64ビット数のホストバイトオーダーからネットワークバイトオーダーへの変換を実行します(「レガシー」[T_ULARGE_INTEGER](#) [[▶ 306](#)]型)。[バイトオーダー](#) [[▶ 169](#)]も参照してください。

ファンクションHOST_TO_BE64: T_ULARGE_INTEGER

VAR_INPUT

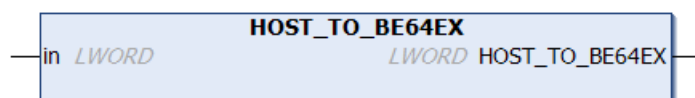
```
VAR_INPUT
  in : T_ULARGE_INTEGER;
END_VAR
```

in: 変換する数(T_ULARGE_INTEGER [▶ 306]型)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.3.5 HOST_TO_BE64EX



ファンクションは、64ビット数のホストバイトオーダーからネットワークバイトオーダーへの変換を実行します(「ネイティブ」LWORD)。[バイトオーダー \[▶ 169\]](#)も参照してください。

ファンクションHOST_TO_BE64EX: LWORD

VAR_INPUT

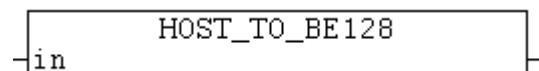
```
VAR_INPUT
  in : LWORD;
END_VAR
```

in: 変換する数。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.3.6 HOST_TO_BE128



ファンクションは、128ビット数のホストバイトオーダーからネットワークバイトオーダーへの変換を実行します(「レガシー」T_UHUGE_INTEGER [▶ 305])。[バイトオーダー \[▶ 169\]](#)も参照してください。

ファンクションHOST_TO_BE128: T_UHUGE_INTEGER

VAR_INPUT

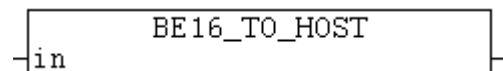
```
VAR_INPUT
  in : T_UHUGE_INTEGER;
END_VAR
```

in: 変換する数(タイプ:T_UHUGE_INTEGER [▶ 305])。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.3.7 BE16_TO_HOST



ファンクションは、16ビット数のネットワークバイトオーダーからホストバイトオーダーへの変換を実行します。[バイトオーダー](#) [[▶_169](#)]も参照してください。

ファンクションBE16_TO_HOST: WORD

VAR_INPUT

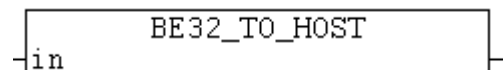
```
VAR_INPUT
  in : WORD;
END_VAR
```

in: 変換する数。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.3.8 BE32_TO_HOST



ファンクションは、32ビット数のネットワークバイトオーダーからホストバイトオーダーへの変換を実行します。[バイトオーダー](#) [[▶_169](#)]も参照してください。

ファンクションBE32_TO_HOST: DWORD

VAR_INPUT

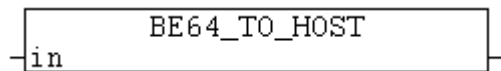
```
VAR_INPUT
  in : DWORD;
END_VAR
```

in: 変換する数。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.3.9 BE64_TO_HOST



ファンクションは、64ビット数のネットワークバイトオーダーからホストバイトオーダーへの変換を実行します(「レガシー」 T_ULARGE_INTEGER [▶ 306])。バイトオーダー [▶ 169]も参照してください。

ファンクションBE64_TO_HOST: T_ULARGE_INTEGER

VAR_INPUT

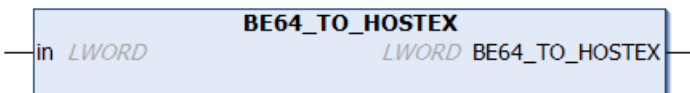
```
VAR_INPUT
    in : T_ULARGE_INTEGER;
END_VAR
```

in: 変換する数(タイプ:T_ULARGE_INTEGER [▶ 306])。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.3.10 BE64_TO_HOSTEX



ファンクションは、64ビット数のネットワークバイトオーダーからホストバイトオーダーへの変換を実行します(「ネイティブ」 LWORD)。バイトオーダー [▶ 169]も参照してください。

ファンクションBE64_TO_HOSTEX: LWORD

VAR_INPUT

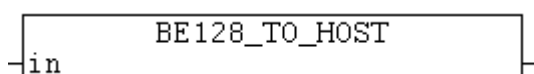
```
VAR_INPUT
    in : LWORD;
END_VAR
```

in: 変換する数。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.3.11 BE128_TO_HOST



ファンクションは、128ビット数のネットワークバイトオーダーからホストバイトオーダーへの変換を実行します(「レガシー」 T_UHUGE_INTEGER [▶ 305])。バイトオーダー [▶ 169]も参照してください。

ファンクションBE128_TO_HOST: T_UHUGE_INTEGER

VAR_INPUT

```
VAR_INPUT
  in : T_UHUGE_INTEGER;
END_VAR
```

in: 変換する数(T_UHUGE_INTEGER [▶_305])。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.4 FLOATファンクション

4.4.1 BOOL_TO_FLOAT

ファンクションBOOL_TO_FLOAT: LREAL

VAR_INPUT

```
VAR_INPUT
  in : BOOL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.4.2 DINT_TO_FLOAT

ファンクションDINT_TO_FLOAT: FLOAT

VAR_INPUT

```
VAR_INPUT
  in : DINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.4.3 FLOAT_TO_BOOL

ファンクションFLOAT_TO_BOOL: BOOL

VAR_INPUT

```
VAR_INPUT
  in      : LREAL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.4.4 FLOAT_TO_DINT

ファンクションFLOAT_TO_DINT: DINT

VAR_INPUT

```
VAR_INPUT
  in      : FLOAT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.4.5 FLOAT_TO_INT

ファンクションFLOAT_TO_INT: INT

VAR_INPUT

```
VAR_INPUT
  in      : FLOAT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.4.6 FLOAT_TO_SINT

ファンクションFLOAT_TO_SINT: SINT

VAR_INPUT

```
VAR_INPUT
  in      : FLOAT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.4.7 FLOAT_TO_STRING

ファンクションFLOAT_TO_STRING: STRING

VAR_INPUT

```
VAR_INPUT
    in          : FLOAT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.4.8 FLOAT_TO_TIME

ファンクションFLOAT_TO_TIME: TIME

VAR_INPUT

```
VAR_INPUT
    in          : FLOAT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.4.9 FLOAT_TO_UDINT

ファンクションFLOAT_TO_UDINT: UDINT

VAR_INPUT

```
VAR_INPUT
    in          : FLOAT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.4.10 FLOAT_TO_UINT

ファンクションFLOAT_TO_UINT: UINT

VAR_INPUT

```
VAR_INPUT
  in      : FLOAT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.4.11 INT_TO_FLOAT

ファンクションINT_TO_FLOAT: FLOAT

VAR_INPUT

```
VAR_INPUT
  in      : INT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.4.12 SINT_TO_FLOAT

ファンクションSINT_TO_FLOAT: FLOAT

VAR_INPUT

```
VAR_INPUT
  in      : SINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.4.13 TIME_TO_FLOAT

ファンクションTIME_TO_FLOAT: FLOAT

VAR_INPUT

```
VAR_INPUT
  in:      TIME;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.4.14 UDINT_TO_FLOAT

ファンクションUDINT_TO_FLOAT: FLOAT

VAR_INPUT

```
VAR_INPUT
    in    : UDINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.4.15 UINT_TO_FLOAT

ファンクションUINT_TO_FLOAT: FLOAT

VAR_INPUT

```
VAR_INPUT
    in    : UINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.4.16 LrealIsFinite

ファンクションLrealIsFinite: BOOL

VAR_INPUT

```
VAR_INPUT
    x    : REFERENCE TO LREAL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4020	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)、 3.3.16.0以降

4.4.17 LrealIsNaN

このファンクションは、値がNaN (非数) であるかどうかテストします。値がNaNの場合、戻り値はTRUEです。

```
FUNCTION LrealIsNaN : BOOL
VAR_INPUT
    x      : REFERENCE TO LREAL;
END_VAR
```

以下の点は、NaN値の主な特徴です。

- ・ NaNを入力データとして使用するすべての算術演算は、演算結果としてNaNを返します。
- ・ 少なくともオペランドの1つがNaNの場合、すべての関係演算子=、≠、> < >= <=は、必ず値Falseを返します。
- ・ 引数が値NaNの場合、スタンダードCファンクションisnan() _isnan()、またはPLCファンクション LrealIsNaN() [▶ 178] (Tc2_Uilitiesライブラリ)は値TRUEを返します。
- ・ 式isnan(a)は、式!(a == a) またはNOT(a = a)に相当します。

さらに計算で使用する場合、NaN値が自身を再生成するという事実は、無効な値が見過ごされることができないという点で 有利です。

⚠ 注意

ソフトウェアの誤作動

NaN値は明示的に承認される場合、NaN値をPLCライブラリでのみ使用できます。特に、モーション制御とドライブ制御のファンクションで制御値として使用できます。そうでない場合、NaN値は該当するソフトウェアに潜在的に危険な誤作動を引き起こすことがあります。

⚠ 注意

浮動小数点の例外処理

NaNが使用され、アプリケーションで処理される場合、FPの例外処理はオフにしなければいけません。オフでない場合、NaNとの比較が例外処理を引き起こし、ランタイムの停止と機械の損傷を引き起こします。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4020	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)、 3.3.16.0以降

4.5 LCOMPLEXファンクション

4.5.1 LcomplexIsNaN



LCOMPLEX型の引数が未定義の値(NaN)を保持している場合、ファンクションはTRUEを返します。

ファンクションLcomplexIsNaN : BOOL

VAR_INPUT

```
VAR_INPUT
    Z      : REFERENCE TO LCOMPLEX;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4020	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.16.0以降

4.5.2 LcomplexAbs



ファンクションは、送信される複素数の絶対値を返します。

ファンクションLcomplexAbs : LREAL

VAR_INPUT

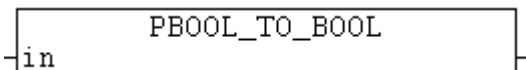
```
VAR_INPUT
  Z : LCOMPLEX;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)、 3.3.21.0以降

4.6 P[TYPE]_TO_[TYPE]変換ファンクション

4.6.1 PBOOL_TO_BOOL



ファンクションは、BOOLポインタ変数の内容を返します。

ファンクションPBOOL_TO_BOOL: BOOL

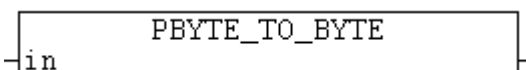
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO BOOL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.6.2 PBYTE_TO_BYTE



ファンクションは、BYTEポインタ変数の内容を返します。

ファンクションPBYTE_TO_BYTE: BYTE

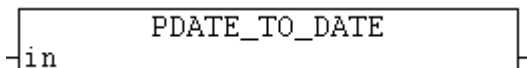
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO BYTE;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.3 PDATE_TO_DATE



ファンクションは、DATEポインタ変数の内容を返します。

ファンクションPDATE_TO_DATE: DATE

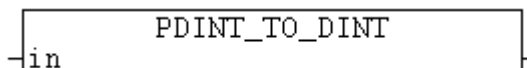
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO DATE;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.4 PDINT_TO_DINT



ファンクションは、DINTポインタ変数の内容を返します。

ファンクションPDINT_TO_DINT: DINT

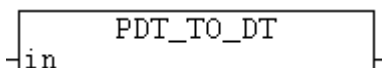
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO DINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.5 PDT_TO_DT



ファンクションは、DTポインタ変数の内容を返します。

ファンクションPDT_TO_DT: DT

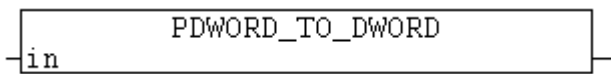
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO DT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.6 PDWORD_TO_DWORD



ファンクションは、DWORDポインタ変数の内容を返します。

ファンクションPDWORD_TO_DWORD: DWORD

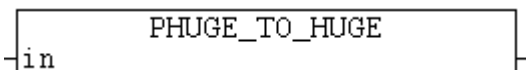
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO DWORD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.7 PHUGE_TO_HUGE



ファンクションは、T_HUGE_INTEGER [▶_304]ポインタ変数の内容を返します。

ファンクションPHUGE_TO_HUGE: T_HUGE_INTEGER

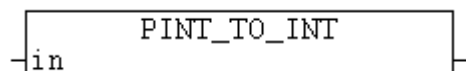
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO T_HUGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.8 PINT_TO_INT



ファンクションは、INTポインタ変数の内容を返します。

ファンクションPINT_TO_INT: INT

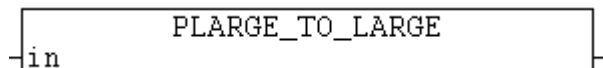
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO INT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.6.9 PLARGE_TO_LARGE



ファンクションは、T_LARGE_INTEGER [▶ 305]ポインタ変数の内容を返します。

ファンクションPLARGE_TO_LARGE: T_LARGE_INTEGER

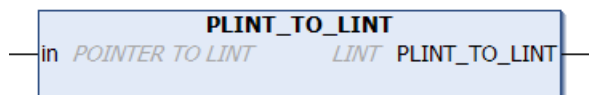
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO T_LARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.6.10 PLINT_TO_LINT



ファンクションは、LINTポインタ変数の内容を返します。

ファンクションPLINT_TO_LINT: LINT

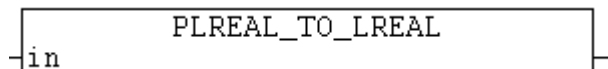
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO LINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.6.11 PLREAL_TO_LREAL



ファンクションは、LREALポインタ変数の内容を返します。

ファンクションPLREAL_TO_LREAL: LREAL

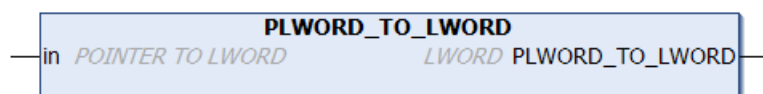
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO LREAL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.6.12 PLWORD_TO_LWORD



ファンクションは、LWORDポインタ変数の内容を返します。

ファンクションPLWORD_TO_LWORD: LWORD

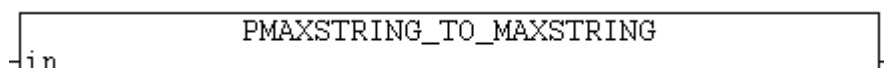
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO LWORD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.6.13 PMAXSTRING_TO_MAXSTRING



ファンクションは、T_MaxStringポインタ変数の内容を返します。

ファンクション PMAXSTRING_TO_MAXSTRING: T_MaxString

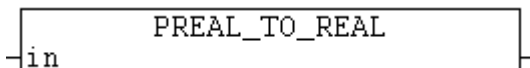
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO T_MaxString;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.14 PREAL_TO_REAL



ファンクションは、REALポインタ変数の内容を返します。

ファンクションPREAL_TO_REAL: REAL

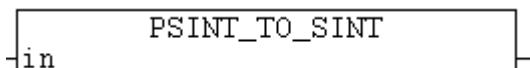
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO REAL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.15 PSINT_TO_SINT



ファンクションは、SINTポインタ変数の内容を返します。

ファンクションPSINT_TO_SINT: SINT

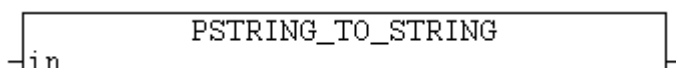
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO SINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.16 PSTRING_TO_STRING



ファンクションは、STRINGポインタ変数の内容を返します。

ファンクションPSTRING_TO_STRING: STRING

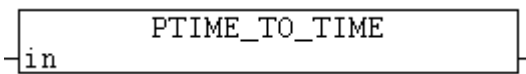
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO STRING;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.17 PTIME_TO_TIME



ファンクションは、TIMEポインタ変数の内容を返します。

ファンクションPTIME_TO_TIME: TIME

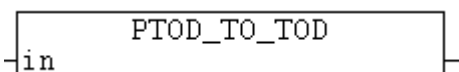
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO TIME;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.18 PTOD_TO_TOD



ファンクションは、TODポインタ変数の内容を返します。

ファンクションPTOD_TO_TOD: TOD

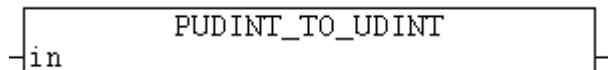
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO TOD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.19 PUDINT_TO_UDINT



ファンクションは、UDINTのポインタ変数の内容を返します。

ファンクションPUDINT_TO_UDINT: UDINT

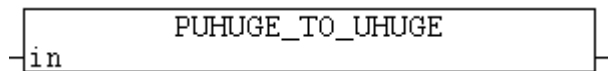
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO UDINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.20 PUHUGE_TO_UHUGE



ファンクションは、T_UHUGE_INTEGER [▶ 305]ポインタ変数の内容を返します。

ファンクションPUHUGE_TO_UHUGE: T_UHUGE_INTEGER

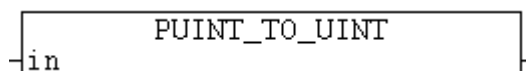
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO T_UHUGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.21 PUINT_TO_UINT



ファンクションは、UINTポインタ変数の内容を返します。

ファンクションPUINT_TO_UINT: UINT

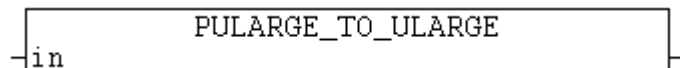
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO UINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.22 PULARGE_TO_ULARGE



ファンクションは、T_ULARGE_INTEGER [▶ 306]ポインタ変数の内容を返します。

ファンクションPULARGE_TO_ULARGE: T_ULARGE_INTEGER

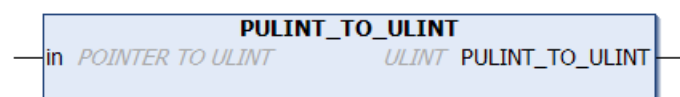
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.23 PULINT_TO_ULINT



ファンクションは、ULINTポインタ変数の内容を返します。

ファンクションPULINT_TO_ULINT: ULINT

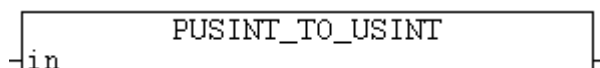
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO ULINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.24 PUSINT_TO_USINT



ファンクションは、USINTポインタ変数の内容を返します。

ファンクションPUSINT_TO_USINT: USINT

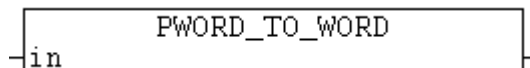
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO USINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.25 PWORD_TO_WORD



ファンクションは、WORDポインタ変数の内容を返します。

ファンクションPWORD_TO_WORD: WORD

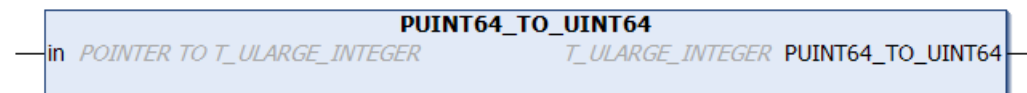
VAR_INPUT

```
VAR_INPUT
  in : POINTER TO WORD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.6.26 PUINT64_TO_UINT64



ファンクションPUINT64_TO_UINT64は、T_ULARGE_INTEGER [▶ 306]ポインタ変数の内容を返します。

ファンクションPUINT64_TO_UINT64: T_ULARGE_INTEGER

VAR_INPUT

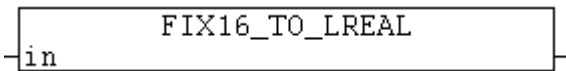
```
VAR_INPUT
  in : POINTER TO T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.7 16ビット固定小数点数ファンクション(符号付き)

4.7.1 FIX16_TO_LREAL



符号付き16ビット固定小数点数をタイプLREALの浮動小数点数に変換します。

ファンクションFIX16_TO_LREAL: LREAL

VAR_INPUT

```
VAR_INPUT
    in : T_FIX16;
END_VAR
```

in: 変換する固定小数点数 ([T_FIX16](#) [[▶ 302](#)])。

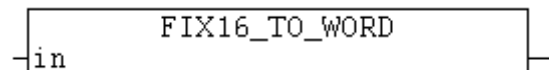
例:

[LREAL_TO_FIX16](#) [[▶ 194](#)] ファンクションの説明を参照してください。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.7.2 FIX16_TO_WORD



このファンクションは、16ビット固定小数点数をWORD型変数に変換します (WORD型変数は、固定小数点数のために数字と小数位を格納します)。

ファンクションFIX16_TO_WORD: WORD

VAR_INPUT

```
VAR_INPUT
    in : T_FIX16;
END_VAR
```

in: 変換する固定小数点数 ([T_FIX16](#) [[▶ 302](#)])。

例:

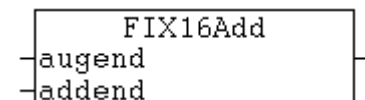
```
PROGRAM FIX_TO_WORD
VAR
    fp16 : WORD;
END_VAR
fp16 := FIX16_TO_WORD(LREAL_TO_FIX16(12.5, 8));
```

fp16 変数の値は、2#0000110010000000です。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.7.3 FIX16Add



このファンクションは、2つの符号付き16ビット固定小数点数を追加します。2つの数字は同じ精度(少数点以下の桁数)である必要はありません。少数点以下の桁数が多い数字の精度は加算の前に削減されます。すなわち、精度の高い方の数字の小数位は切り捨てられます。加算の結果は、符号付き16ビット固定小数点数です。

ファンクションFIX16Add: [T_FIX16](#) [[▶](#) [302](#)]

VAR_INPUT

```
VAR_INPUT
    augend : T_FIX16;
    addend : T_FIX16;
END_VAR
```

augend: 最初の被加数 ([T_FIX16](#) [[▶](#) [302](#)])。

addend: 2番目の被加数 ([T_FIX16](#) [[▶](#) [302](#)])。

例:

```
PROGRAM FIXADD
VAR
    a, b    : T_FIX16;
    result  : LREAL;
END_VAR

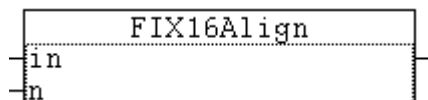
a := LREAL_TO_FIX16( 0.5, 8 );
b := LREAL_TO_FIX16( -0.25, 8 );

result := FIX16_TO_LREAL( FIX16Add( a, b ) );(* The result is: 0.25 *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.7.4 FIX16Align



このファンクションを使用して、符号付き16ビット固定小数点数の精度(少数点以下の桁数)を変更できます。ファンクションは、新しい固定小数点数を戻りパラメータとして返します。

ファンクションFIX16Align: [T_FIX16 \[▶ 302\]](#)

VAR_INPUT

```
VAR_INPUT
  in : T_FIX16;
  n  : BYTE(0..15);
END_VAR
```

in: 精度が変更される固定小数点数 ([T_FIX16 \[▶ 302\]](#))。

n: 新しい少数点以下の桁数。

例:

```
PROGRAM FIXALIGN
VAR
  q8, q4 : T_FIX16;
  result : LREAL;
END_VAR

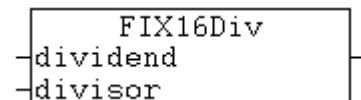
q8 := LREAL_TO_FIX16( 0.6, 8 );
result := FIX16_TO_LREAL( q8 );(* The result is: 0.6015625 *)

q4 := FIX16Align( q8, 4 );
result := FIX16_TO_LREAL( q4 );(* The result is: 0.5625 *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.7.5 FIX16Div



このファンクションは、2つの符号付き16ビット固定小数点数を除算します。2つの数字は同じ精度(少数点以下の桁数)である必要はありません。少数点以下の桁数が多い数字の精度は除算の前に削減されます。すなわち、精度の高い方の数字の小数位は切り捨てられます。除算の結果は、符号付き16ビット固定小数点数です。

ファンクションFIX16Div: [T_FIX16 \[▶ 302\]](#)

VAR_INPUT

```
VAR_INPUT
  dividend : T_FIX16;
  divisor  : T_FIX16;
END_VAR
```

dividend: 除算される数(被除数) ([T_FIX16 \[▶ 302\]](#))。

divisor: 除算する数(除数) ([T_FIX16 \[▶ 302\]](#))。

例:

```
PROGRAM FIXDIV
VAR
  a, b : T_FIX16;
  result : LREAL;
END_VAR
```



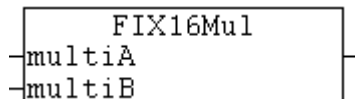
```
a := LREAL_TO_FIX16( -22.5, 8 );
b := LREAL_TO_FIX16( 10.0, 8 );

result := FIX16_TO_LREAL( FIX16Div( a, b ) ); (* The result is: -2.25 *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.7.6 FIX16Mul



このファンクションは、2つの符号付き16ビット固定小数点数を乗算します。2つの数字は同じ精度(少数点以下の桁数)である必要はありません。少数点以下の桁数が多い数字の精度は乗算の前に削減されます。すなわち、精度の高い方の数字の小数位は切り捨てられます。乗算の結果は、符号付き16ビット固定小数点数です。

ファンクションFIX16Mul: [T_FIX16 \[▶ 302\]](#)

VAR_INPUT

```
VAR_INPUT
    multiA : T_FIX16;
    multiB : T_FIX16;
END_VAR
```

multiA: 最初の乗数([T_FIX16 \[▶ 302\]](#))。

multiB: 2番目の乗数([T_FIX16 \[▶ 302\]](#))。

例:

```
PROGRAM FIXMUL
VAR
    a, b : T_FIX16;
    result : LREAL;
END_VAR

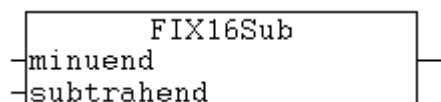
a := LREAL_TO_FIX16( 0.25, 8 );
b := LREAL_TO_FIX16( 10.0, 8 );

result := FIX16_TO_LREAL( FIX16Mul( a, b ) );(* The result is: 2.5 *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.7.7 FIX16Sub



このファンクションは、2つの符号付き16ビット固定小数点数を減算します。2つの数字は同じ精度（少数点以下の桁数）である必要はありません。少数点以下の桁数が多い数字の精度は減算の前に削減されます。すなわち、精度の高い方の数字の小数位は切り捨てられます。減算の結果は、符号付き16ビット固定小数点数です。

ファンクションFIX16Sub: [T_FIX16 \[▶ 302\]](#)

VAR_INPUT

```
VAR_INPUT
    minuend      : T_FIX16;
    subtrahend   : T_FIX16;
END_VAR
```

minuend: 値が減算される方の数(被減数) ([T_FIX16 \[▶ 302\]](#))。

subtrahend: 減算する方の数(減数) ([T_FIX16 \[▶ 302\]](#))。

例:

```
PROGRAM FIXSUB
VAR
    a, b      : T_FIX16;
    result    : LREAL;
END_VAR

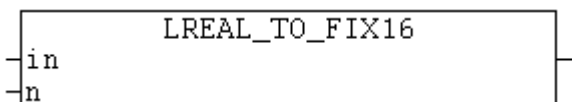
a := LREAL_TO_FIX16( 0.5, 8 );
b := LREAL_TO_FIX16( 0.75, 8 );

result := FIX16_TO_LREAL( FIX16Sub( a, b ) ); (* The result is: -0.25 *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86, x64, ARM)	Tc2_Uutilities (システム)

4.7.8 LREAL_TO_FIX16



LREAL型の浮動小数点数を必要な少数点以下の桁数をもつ符号付き16ビット固定小数点数に変換します。

ファンクションLREAL_TO_FIX16: [T_FIX16 \[▶ 302\]](#)

VAR_INPUT

```
VAR_INPUT
    in : LREAL;
    n  : WORD(0..15) := 15;
END_VAR
```

in: 変換するLREAL数。

n: 必要な小数位の数。

例:

以下の例では、複数の定数が固定小数点数に変換されます。変換の際に、少数点以下の桁数を指定できません。丸め誤差(浮動小数点数の変換同様)が発生することがあることに注意してください(本例ではq2および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 );

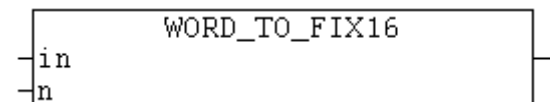
r2 := FIX16_TO_LREAL( q2 ); (* 0.5 *)
r4 := FIX16_TO_LREAL( q4 ); (* -0.25 *)
r8 := FIX16_TO_LREAL( q8 ); (* -0.75 *)
r12 := FIX16_TO_LREAL( q12 ); (* 2.30078125 *)
r15 := FIX16_TO_LREAL( q15 ); (* 0.600006103515625 *)

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.7.9 WORD_TO_FIX16



このファンクションは、WORD型変数を16ビット固定小数点数に変換します (WORD型変数は、固定小数点数のためにコード化数と小数位を格納します)。

ファンクションWORD_TO_FIX16: [T_FIX16](#) [[▶ 302](#)]

VAR_INPUT

```

VAR_INPUT
  in : WORD; (* 16 bit fixed point number *)
  n : WORD(0..15); (* number of fractional bits *)
END_VAR

```

例:

```

PROGRAM WORD_TO_FIX
VAR
  double : LREAL;
END_VAR

```

<code />

```
double := FIX16_TO_LREAL(WORD_TO_FIX16(2#0000110010000000, 8));
```

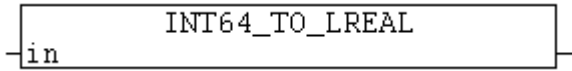
*double*変数の値は、12.5です。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.8 64ビットファンクション（符号付き）

4.8.1 INT64_TO_LREAL



ファンクションは、TwinCAT 2の符号付き64ビット数（「レガシー」 T_LARGE_INTEGER [▶ 305]）をLREAL型の浮動小数点数に変換します。

ファンクションINT64_TO_LREAL: LREAL

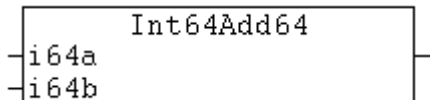
VAR_INPUT

```
VAR_INPUT
  in : T_LARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.8.2 Int64Add64



ファンクションは、2つのTwinCAT 2の符号付き64ビット数（「レガシー」 T_LARGE_INTEGER [▶ 305]）を加算します。演算結果は符号付き64ビット数になります。

FUNCTION Int64Add64: T_LARGE_INTEGER

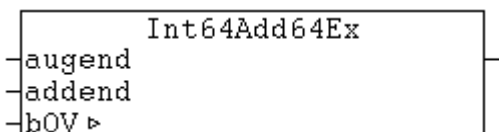
VAR_INPUT

```
VAR_INPUT
  i64a : T_LARGE_INTEGER;
  i64b : T_LARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.8.3 Int64Add64Ex



ファンクションは、2つのTwinCAT 2の符号付き64ビット数（「レガシー」 T_LARGE_INTEGER [▶ 305]）を加算します。演算結果は符号付き64ビット数になります。

ファンクションInt64Add64Ex: T_LARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
  augend : T_LARGE_INTEGER;
  addend : T_LARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

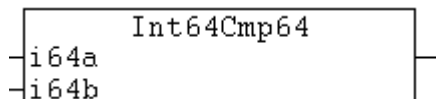
```
VAR_IN_OUT
  bOV : BOOL; (* TRUE => arithmetic overflow, FALSE => no overflow *)
END_VAR
```

bOV: 算術桁あふれ。TRUE → オーバーフロー、FALSE → オーバーフローなし。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.8.4 Int64Cmp64



ファンクションは、2つのTwinCAT 2の符号付き64ビット数を比較します(「レガシー」[T_LARGE_INTEGER](#) [[▶ 305](#)])。

ファンクションInt64Cmp64: DINT

VAR_INPUT

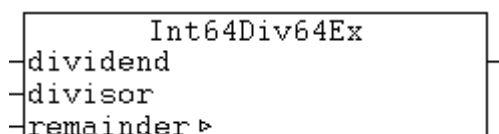
```
VAR_INPUT
  i64a : T_LARGE_INTEGER;
  i64b : T_LARGE_INTEGER;
END_VAR
```

戻りパラメータ	説明
-1	i64aはi64b未満
0	i64aはi64bと同一
1	i64aはi64bより大きい

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.8.5 Int64Div64Ex



ファンクションは、2つのTwinCAT 2の符号付き64ビット数を除算します(「レガシー」[T_LARGE_INTEGER](#) [[▶ 305](#)])。演算結果は符号付き64ビット数になります。

ファンクションInt64Div64Ex: T_LARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
  dividend : T_LARGE_INTEGER;
  divisor  : T_LARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

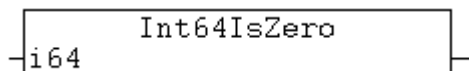
```
VAR_IN_OUT
  remainder : T_LARGE_INTEGER;
END_VAR
```

remainder: 剰余

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.8.6 Int64IsZero



TwinCAT 2の符号付き64ビット数の値がゼロの場合、ファンクションはTRUEを返します(「レガシー」[T_LARGE_INTEGER \[▶_305\]](#))。

ファンクションInt64isZero: BOOL

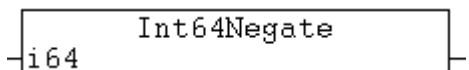
VAR_INPUT

```
VAR_INPUT
  i64 : T_LARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.8.7 Int64Negate



ファンクションは、TwinCAT 2の符号付き64ビット数を反転します(「レガシー」[T_LARGE_INTEGER \[▶_305\]](#))。

ファンクションInt64Negate: T_LARGE_INTEGER

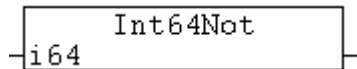
VAR_INPUT

```
VAR_INPUT
  i64 : T_LARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.8.8 Int64Not



TwinCAT 2の符号付き64ビット数のビット単位のNOT(「レガシー」[T_LARGE_INTEGER \[▶_305\]](#))。演算結果は符号付き64ビット数になります。

ファンクションInt64Not: [T_LARGE_INTEGER](#)

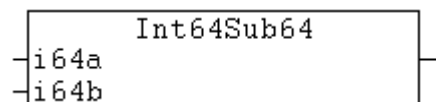
VAR_INPUT

```
VAR_INPUT
    i64 : T_LARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.8.9 Int64Sub64



ファンクションは、2つのTwinCAT 2の符号付き64ビット数を減算します(「レガシー」[T_LARGE_INTEGER \[▶_305\]](#))。演算結果は符号付き64ビット数になります。

ファンクションInt64Sub64: [T_LARGE_INTEGER](#)

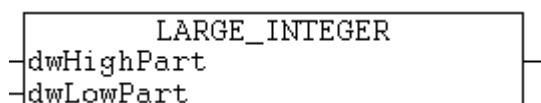
VAR_INPUT

```
VAR_INPUT
    i64a : T_LARGE_INTEGER;
    i64b : T_LARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.8.10 LARGE_INTEGER



ファンクションは、TwinCAT 2の符号付き64ビット数を初期化します(「レガシー」[T_LARGE_INTEGER \[▶_305\]](#))。

ファンクションLARGE_INTEGER: T_LARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
  dwHighPart : DWORD;
  dwLowPart  : DWORD;
END_VAR
```

dwHighPart: 上位の32ビット。

dwLowPart: 下位の32ビット。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.8.11 LARGE_TO_LINT



ファンクションは、TwinCAT 2の符号付き64ビット数(「レガシー」 T_LARGE_INTEGER [▶_305])をTwinCAT 3の符号付き64ビット数(「ネイティブ」タイプ)に変換します。

ファンクションLARGE_TO_LINT: LINT

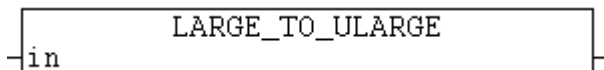
VAR_INPUT

```
VAR_INPUT
  in : T_LARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.8.12 LARGE_TO_ULARGE



ファンクションは、TwinCAT 2の符号付き64ビット数(「レガシー」 T_LARGE_INTEGER [▶_305])をTwinCAT 2の符号なし64ビット数(「ネイティブ」 T_ULARGE_INTEGER [▶_306])に変換します。

ファンクションLARGE_TO_ULARGE: T_ULARGE_INTEGER

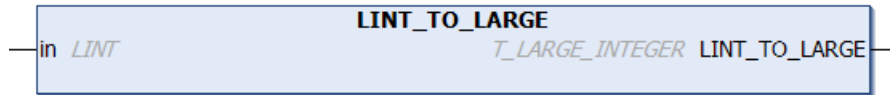
VAR_INPUT

```
VAR_INPUT
  in : T_LARGE_INTEGER;
END_VAR
```


要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.8.13 LINT_TO_LARGE



ファンクションは、TwinCAT 3の符号付き64ビット数(「ネイティブ」タイプ)をTwinCAT 2の符号付き64ビット数(「legacy」 [T_LARGE_INTEGER \[▶ 305\]](#))に変換します。

ファンクションLINT_TO_LARGE: T_LARGE_INTEGER

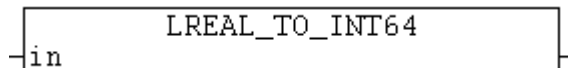
VAR_INPUT

```
VAR_INPUT
    in : LINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.8.14 LREAL_TO_INT64



ファンクションは、LREAL数をTwinCAT 2の符号付き64ビット数(「レガシー」 [T_LARGE_INTEGER \[▶ 305\]](#))に変換します。

ファンクションLREAL_TO_INT64: T_LARGE_INTEGER

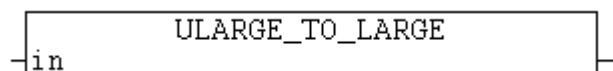
VAR_INPUT

```
VAR_INPUT
    in : LREAL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.8.15 ULARGE_TO_LARGE



ファンクションは、TwinCAT 2の符号なし64ビット数(「レガシー」 [T_ULARGE_INTEGER \[▶ 306\]](#))をTwinCAT 2の符号付き64ビット数(「ネイティブ」 [T_LARGE_INTEGER \[▶ 305\]](#))に変換します。

ファンクションULARGE_TO_LARGE: T_LARGE_INTEGER

VAR_INPUT

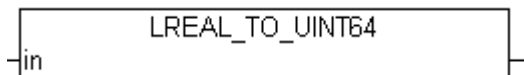
```
VAR_INPUT
  in : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9 64ビット整数ファンクション(符号なし)

4.9.1 LREAL_TO_UINT64



ファンクションは、LREAL数をTwinCAT 2の符号なし64ビット数(「レガシー」 T_ULARGE_INTEGER [▶ 306])に変換します。

ファンクションLREAL_TO_UINT64: T_ULARGE_INTEGER

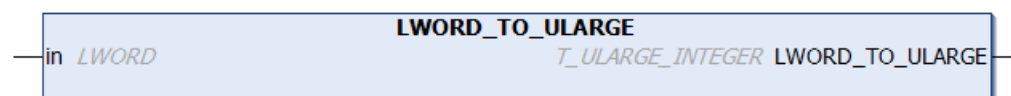
VAR_INPUT

```
VAR_INPUT
  in : LREAL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.2 LWORD_TO_ULARGE



ファンクションは、TwinCAT 3の符号なし64ビット数(「ネイティブ」タイプ)をTwinCAT 2の符号なし64ビット数(「レガシー」 T_ULARGE_INTEGER [▶ 306])に変換します。

ファンクションLWORD_TO_ULARGE: T_ULARGE_INTEGER

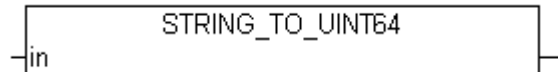
VAR_INPUT

```
VAR_INPUT
  in : LWORD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.3 STRING_TO_UINT64



ファンクションは、文字列をTwinCAT 2の符号なし64ビット数(「レガシー」[T_ULARGE_INTEGER \[▶ 306\]](#))に変換します。

ファンクションSTRING_TO_UINT64: T_ULARGE_INTEGER

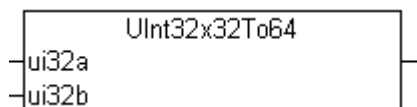
VAR_INPUT

```
VAR_INPUT
  in : STRING(21);
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.4 UInt32x32To64



ファンクションは、2つの符号なし32ビット数を乗算します。演算結果は、TwinCAT 2の符号なし64ビット数(「レガシー」[T_ULARGE_INTEGER \[▶ 306\]](#))です。

ファンクションUInt32x32To64: T_ULARGE_INTEGER

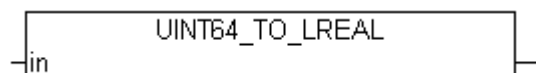
VAR_INPUT

```
VAR_INPUT
  ui32a : DWORD;
  ui32b : DWORD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.5 UINT64_TO_LREAL



ファンクションは、TwinCAT 2の符号なし64ビット数(「legacy」[T_ULARGE_INTEGER \[▶ 306\]](#))をLREAL型の浮動小数点数に変換します。

ファンクションUINT64_TO_LREAL: LREAL

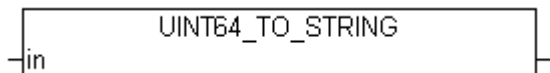
VAR_INPUT

```
VAR_INPUT
  in : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.6 UINT64_TO_STRING



ファンクションは、TwinCAT 2の符号なし64ビット数(「レガシー」T_ULARGE_INTEGER [▶_306])を文字列に変換します。

ファンクションUINT64_TO_STRING: STRING(21)

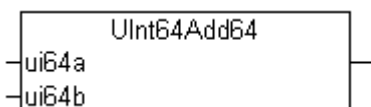
VAR_INPUT

```
VAR_INPUT
  in : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.7 UInt64Add64



ファンクションは、2つのTwinCAT 2の符号なし64ビット数(「レガシー」T_ULARGE_INTEGER [▶_306])を加算します。演算結果は、符号なし64ビット数です。

ファンクションUInt64Add64: T_ULARGE_INTEGER

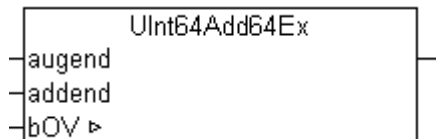
VAR_INPUT

```
VAR_INPUT
  ui64a : T_ULARGE_INTEGER;
  ui64b : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.8 UInt64Add64Ex



ファンクションは、2つのTwinCAT 2の符号なし64ビット数(「レガシー」 [T_ULARGE_INTEGER \[▶ 306\]](#))を加算します。演算結果は、符号なし64ビット数です。

ファンクションUInt64Add64Ex: [T_ULARGE_INTEGER](#)

VAR_INPUT

```
VAR_INPUT
    augend : T_ULARGE_INTEGER;
    addend : T_ULARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

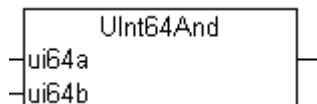
```
VAR_IN_OUT
    bOV    : BOOL; (* TRUE => arithmetic overflow, FALSE => no overflow *)
END_VAR
```

bOV: 算術桁あふれ。TRUE → オーバーフロー、FALSE → オーバーフローなし。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.9.9 UInt64And



2つのTwinCAT 2の符号なし64ビット数のビット単位のAND(「レガシー」 [T_ULARGE_INTEGER \[▶ 306\]](#))。演算結果は、符号なし64ビット数です。

ファンクションUInt64And: [T_ULARGE_INTEGER](#)

VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.9.10 UInt64Cmp64



ファンクションは、2つのTwinCAT 2の符号なし64ビット数を比較します(「レガシー」 [T_ULARGE_INTEGER](#) [[▶ 306](#)])。

ファンクションUInt64Cmp64: DINT

VAR_INPUT

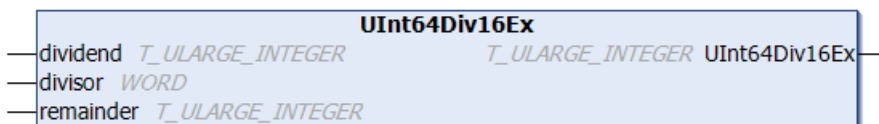
```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

戻りパラメータ	説明
-1	ui64aはui64b未満
0	ui64aはui64bと同一
1	ui64aはui64bより大きい

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.9.11 UInt64Div16Ex



ファンクションは、TwinCAT 2の符号なし64ビット数(「レガシー」 [T_ULARGE_INTEGER](#) [[▶ 306](#)])を16ビット符号なし数で除算します。演算結果は、符号なし64ビット数です。

ファンクションUInt64Div16Ex: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    dividend : T_ULARGE_INTEGER;
    divisor : WORD;
END_VAR
```

VAR_IN_OUT

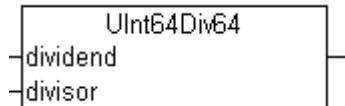
```
VAR_IN_OUT
    remainder : T_ULARGE_INTEGER;
END_VAR
```

remainder: 剰余

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.9.12 UInt64Div64



ファンクションは、2つのTwinCAT 2の符号なし64ビット数(「レガシー」 T_ULARGE_INTEGER [▶_306])を除算します。演算結果は、符号なし64ビット数です。

ファンクションUInt64Div64: T_ULARGE_INTEGER

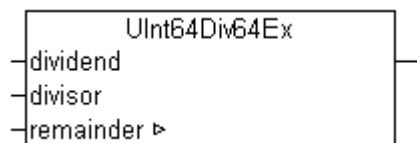
VAR_INPUT

```
VAR_INPUT
    dividend : T_ULARGE_INTEGER;
    divisor  : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.9.13 UInt64Div64Ex



ファンクションは、2つのTwinCAT 2の符号なし64ビット数(「レガシー」 T_ULARGE_INTEGER [▶_306])を除算します。演算結果は、符号なし64ビット数です。

ファンクションUInt64Div64Ex: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    dividend : T_ULARGE_INTEGER;
    divisor  : T_ULARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

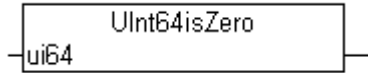
```
VAR_IN_OUT
    remainder : T_ULARGE_INTEGER;
END_VAR
```

remainder: 剰余

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.9.14 UInt64isZero



TwinCAT 2の符号なし64ビット数(「レガシー」[T_ULARGE_INTEGER](#) [[▶ 306](#)])の値がゼロの場合、ファンクションはTRUEを返します。

ファンクションUInt64isZero: BOOL

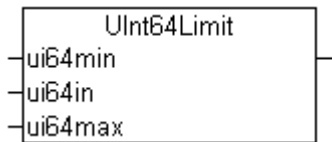
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.15 UInt64Limit



制限。演算結果は、TwinCAT 2の符号なし64ビット数(「レガシー」[T_ULARGE_INTEGER](#) [[▶ 306](#)])です。

ファンクションUInt64Limit: T_ULARGE_INTEGER

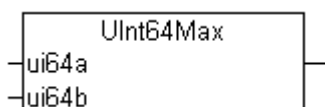
VAR_INPUT

```
VAR_INPUT
    ui64min : T_ULARGE_INTEGER;
    ui64in  : T_ULARGE_INTEGER;
    ui64max : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.16 UInt64Max



最大ファンクション。2つの値の中で大きい方の値を返します(「レガシー」[T_ULARGE_INTEGER](#) [[▶ 306](#)])。

ファンクションUInt64Max: T_ULARGE_INTEGER

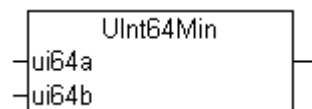
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.17 UInt64Min



最小ファンクション。2つの値の中で小さい方の値を返します(「レガシー」T_ULARGE_INTEGER [▶ 306])。

ファンクションUInt64Min: T_ULARGE_INTEGER

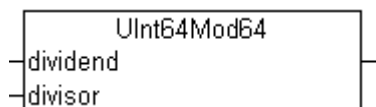
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.18 UInt64Mod64



TwinCAT 2の符号なし64ビット数の他の数でのモジュロ除算(「レガシー」T_ULARGE_INTEGER [▶ 306])。演算結果は、符号なし64ビット数です。

ファンクションUInt64Mod64: T_ULARGE_INTEGER

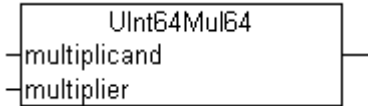
VAR_INPUT

```
VAR_INPUT
    dividend : T_ULARGE_INTEGER;
    divisor : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.19 UInt64Mul64



ファンクションは、2つのTwinCAT 2の符号なし64ビット数を乗算します(「レガシー」 T_ULONG_INTEGER [▶_306])。演算結果は、符号なし64ビット数です。

ファンクションUInt64Mul64: T_ULONG_INTEGER

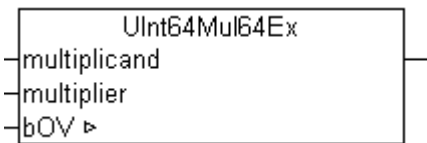
VAR_INPUT

```
VAR_INPUT
    multiplicand : T_ULONG_INTEGER;
    multiplier   : T_ULONG_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.9.20 UInt64Mul64Ex



ファンクションは、2つのTwinCAT 2の符号なし64ビット数を乗算します(「レガシー」 T_ULONG_INTEGER [▶_306])。演算結果は、符号なし64ビット数です。

ファンクションUInt64Mul64Ex: T_ULONG_INTEGER

VAR_INPUT

```
VAR_INPUT
    multiplicand : T_ULONG_INTEGER;
    multiplier   : T_ULONG_INTEGER;
END_VAR
```

VAR_IN_OUT

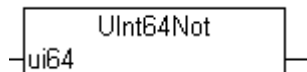
```
VAR_IN_OUT
    bOV : BOOL; (* TRUE => Arithmetic overflow, FALSE => no overflow *)
END_VAR
```

bOV: 算術桁あふれ。TRUE → オーバーフロー、FALSE → オーバーフローなし。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.9.21 UInt64Not



TwinCAT 2の符号なし64ビット数のビット単位のNOT(「レガシー」[T_ULARGE_INTEGER](#) [[▶_306](#)])。演算結果は、符号なし64ビット数です。

ファンクションUInt64Not: [T_ULARGE_INTEGER](#)

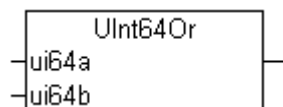
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.22 UInt64Or



2つのTwinCAT 2の符号なし64ビット数のビット単位のOR(「レガシー」[T_ULARGE_INTEGER](#) [[▶_306](#)])。演算結果は、符号なし64ビット数です。

ファンクションUInt64Or: [T_ULARGE_INTEGER](#)

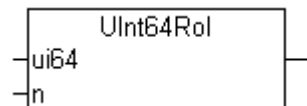
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.23 UInt64Rol



TwinCAT 2の符号なし64ビット数のビット単位の左回転(「レガシー」[T_ULARGE_INTEGER](#) [[▶_306](#)])。演算結果は、符号なし64ビット数です。

ファンクションUInt64Rol: T_ULARGE_INTEGER

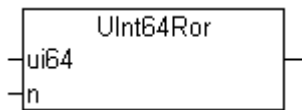
VAR_INPUT

```
VAR_INPUT
  ui64 : T_ULARGE_INTEGER;
  n    : DWORD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.24 UInt64Ror



TwinCAT 2の符号なし64ビット数のビット単位の右回転(「レガシー」[T_ULARGE_INTEGER \[▶_306\]](#))。演算結果は、符号なし64ビット数です。

ファンクションUInt64Ror: T_ULARGE_INTEGER

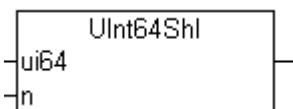
VAR_INPUT

```
VAR_INPUT
  ui64 : T_ULARGE_INTEGER;
  n    : DWORD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.25 UInt64Shl



TwinCAT 2の符号なし64ビット数のビット単位の左シフト(「レガシー」[T_ULARGE_INTEGER \[▶_306\]](#))。演算結果は、符号なし64ビット数です。

ファンクションUInt64Shl: T_ULARGE_INTEGER

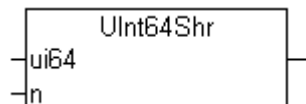
VAR_INPUT

```
VAR_INPUT
  ui64 : T_ULARGE_INTEGER;
  n    : DWORD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.26 UInt64Shr



TwinCAT 2の符号なし64ビット数のビット単位の右シフト(「レガシー」[T_ULARGE_INTEGER \[▶_306\]](#))。演算結果は、符号なし64ビット数です。

ファンクションUInt64Shr: T_ULARGE_INTEGER

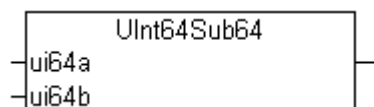
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
    n    : DWORD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.27 UInt64Sub64



ファンクションは、2つのTwinCAT 2の符号なし64ビット数を減算します(「レガシー」[T_ULARGE_INTEGER \[▶_306\]](#))。演算結果は、符号なし64ビット数です。

ファンクションUInt64Sub64: T_ULARGE_INTEGER

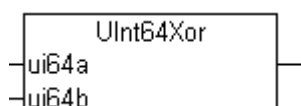
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.28 UInt64Xor



2つのTwinCAT 2の符号なし64ビット数のビット単位のXOR(「レガシー」 T_ULARGE_INTEGER [▶_306])。演算結果は、符号なし64ビット数です。

ファンクションUInt64Xor: T_ULARGE_INTEGER

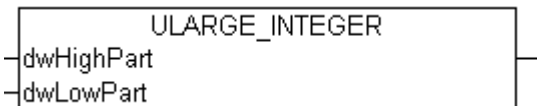
VAR_INPUT

```
VAR_INPUT
  ui64a : T_ULARGE_INTEGER;
  ui64b : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.29 ULARGE_INTEGER



ファンクションは、TwinCAT 2の符号なし64ビット数を初期化します(「レガシー」 T_ULARGE_INTEGER [▶_306])。

ファンクションULARGE_INTEGER: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
  dwHighPart : DWORD;
  dwLowPart : DWORD;
END_VAR
```

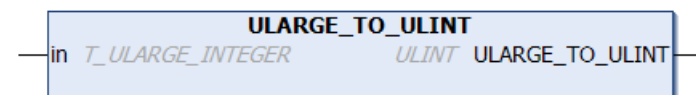
dwHighPart: 上位の32ビット。

dwLowPart: 下位の32ビット。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.9.30 ULARGE_TO_ULINT



ファンクションは、TwinCAT 2の符号なし64ビット数(「レガシー」 T_ULARGE_INTEGER [▶_306])をTwinCAT 3の符号なし64ビット数(「ネイティブ」タイプ)に変換します。

ファンクションULARGE_TO_ULINT: ULINT

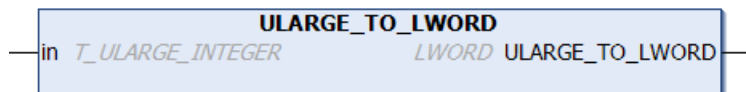
VAR_INPUT

```
VAR_INPUT
  in : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.9.31 ULARGE_TO_LWORD



ファンクションは、TwinCAT 2の符号なし64ビット数(「レガシー」T_ULARGE_INTEGER [▶ 306])をTwinCAT 3の符号なし64ビット数(「ネイティブ」タイプ)に変換します。

ファンクションULARGE_TO_LWORD: LWORD

VAR_INPUT

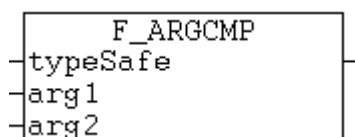
```
VAR_INPUT
  in : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.10 T_Argヘルプファンクション

4.10.1 F_ARGCMP



このファンクションはT_Arg型の2つの変数を比較し、比較結果を戻りパラメータとして返します。

ファンクションF_ARGCMP: DINT

VAR_INPUT

```
VAR_INPUT
  typeSafe : BOOL;
  arg1     : T_Arg;
  arg2     : T_Arg;
END_VAR
```

typeSafe: TRUEの場合、同一タイプを比較できません(型安全比較)。FALSEの場合、異なる型を比較できます(型から独立した比較)。

arg1: 比較される最初の変数 (T_Arg [▶ 302])。

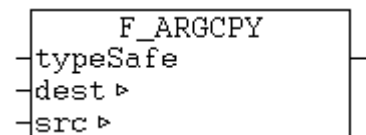
arg2: 比較される2番目の変数 (T_Arg [▶ 302])。

戻りパラメータ	最初の変数と2番目の変数の最初に異なるバイト (型、長さ、値)の関係
-3	arg1の長さはarg2未満
-2	arg1の型はarg2未満
-1	arg1の値はarg2未満
0	arg1はarg2と同一
1	arg1の値はarg2より大きい
2	arg1の型はarg2より大きい
3	arg1の長さはarg2よりも長い
0xFF	arg1またはarg2= 0の誤ったパラメータ値、型、長さ、値

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.10.2 F_ARGCPY



このファンクションは、T_Arg型の変数の値を他の変数にコピーし、コピーに成功したデータバイト数を戻りパラメータとして返します。

ファンクションF_ARGCPY: UDINT

VAR_INPUT

```

VAR_INPUT
    typeSafe : BOOL;
END_VAR
  
```

typeSafe: TRUEの場合、同一タイプを比較できます (型安全比較)。FALSEの場合、異なる型を比較できます (型から独立した比較)。

VAR_IN_OUT

```

VAR_IN_OUT
    dest : T_Arg;
    src : T_Arg;
END_VAR
  
```

dest: コピー操作のターゲット変数 (T_Arg [▶ 302])。

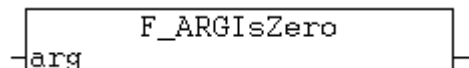
src: コピー操作のソース変数 (T_Arg [▶ 302])。

戻りパラメータ	意味
0	不正なパラメータ値。dest またはsrc== 0の型、長さ、値
> 0	成功した場合、コピーされるバイト数。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.10.3 F_ARGISZERO



T_Argメンバ変数の1つが値ゼロ、または初期化されていない場合、このファンクションはTRUEを返します。

ファンクションF_ARGISZERO: BOOL

VAR_INPUT

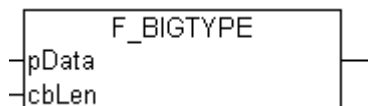
```
VAR_INPUT
  arg : T_Arg;
END_VAR
```

arg: チェックされる変数(タイプ: [T_Arg \[▶ 302\]](#))。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.10.4 F_BIGTYPE



構造変数または配列変数の情報をもつ構造体を返すヘルプファンクション([T_Arg \[▶ 302\]](#))。

ファンクションF_BIGTYPE: T_Arg

VAR_INPUT

```
VAR_INPUT
  pData : POINTER TO BYTE;
  cbLen : DWORD;
END_VAR
```

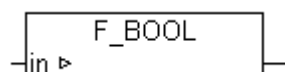
pData: アドレスポインタ(ADR演算子で判定可能)。

cbLen: メモリを占有するバイト数(SIZEOF演算子で判定可能)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.10.5 F_BOOL



BOOL型変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [▶ 302])。

ファンクションF_BOOL: T_Arg

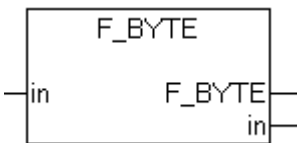
VAR_IN_OUT

```
VAR_IN_OUT
  in : BOOL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.10.6 F_BYTE



BYTE型変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [▶ 302])。

ファンクションF_BYTE: T_Arg

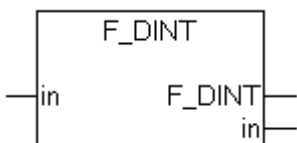
VAR_IN_OUT

```
VAR_IN_OUT
  in : BYTE;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.10.7 F_DINT



DINT変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [▶ 302])。

ファンクションF_DINT: T_Arg

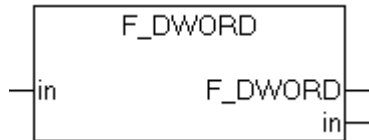
VAR_IN_OUT

```
VAR_IN_OUT
  in : DINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.10.8 F_DWORD



DWORD変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [▶ 302])。

ファンクションF_DWORD: T_Arg

VAR_IN_OUT

```
VAR_IN_OUT
  in : DWORD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.10.9 F_HUGE

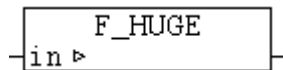


図 1: F_HUGE

T_HUGE_INTEGER [▶ 304]変数(符号付き128ビット整数、TwinCAT 2「レガシー」タイプ)の情報をもつ構造体を返すヘルプファンクション(T_Arg [▶ 302])。

ファンクションF_HUGE: T_Arg

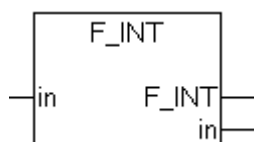
VAR_IN_OUT

```
VAR_IN_OUT
  in : T_HUGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.10.10 F_INT



INT変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [▶ 302])。

ファンクションF_INT: T_Arg

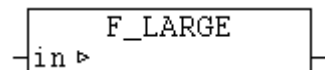
VAR_IN_OUT

```
VAR_IN_OUT
  in : INT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.10.11 F_LARGE



T_LARGE_INTEGER [▶ 305]変数(符号付き64ビット整数、TwinCAT 2「レガシー」タイプ)の情報をもつ構造体を返すヘルプファンクション(T_Arg [▶ 302])。

ファンクションF_LARGE: T_Arg

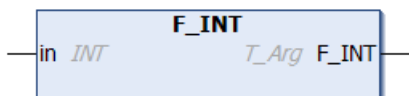
VAR_IN_OUT

```
VAR_IN_OUT
  in : T_LARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.10.12 F_LINT



LINT変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [▶ 302])。

ファンクションF_LINT: T_Arg

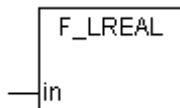
VAR_IN_OUT

```
VAR_IN_OUT
  in : LINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4. 10. 13 F_LREAL



LREAL型変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [▶ 302])。

ファンクションF_LREAL: T_Arg

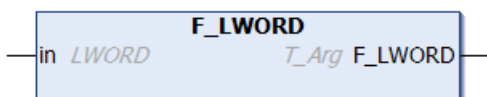
VAR_IN_OUT

```
VAR_IN_OUT
  in : LREAL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4. 10. 14 F_LWORD



LWORD変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [▶ 302])。

ファンクションF_LWORD: T_Arg

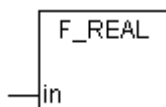
VAR_IN_OUT

```
VAR_IN_OUT
  in : LWORD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4. 10. 15 F_REAL



REAL型変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [▶ 302])。

ファンクションF_REAL: T_Arg

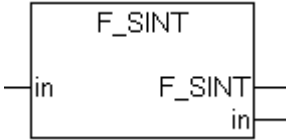
VAR_IN_OUT

```
VAR_IN_OUT
  in : REAL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.10.16 F_SINT



SINT変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [[▶ 302](#)])。

ファンクションF_SINT: T_Arg

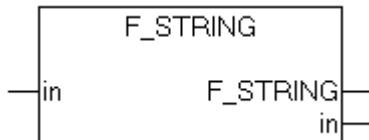
VAR_IN_OUT

```
VAR_IN_OUT
    in : SINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.10.17 F_STRING



T_MaxString変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [[▶ 302](#)])。

ファンクションF_STRING: T_Arg

VAR_IN_OUT

```
VAR_IN_OUT
    in : T_MaxString;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.10.18 F_STRINGEx

STRING変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [[▶ 302](#)])。F_STRING [[▶ 222](#)]ファンクションとは異なり、送信されたSTRING変数の長さは任意です。

ファンクションF_STRINGEx : T_Arg

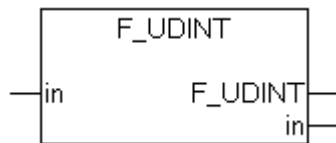
VAR_IN_OUT

```
VAR_IN_OUT CONSTANT
  in : STRING;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.4022	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)、 v3.3.34.0以上

4.10.19 F_UDINT



UDINT変数の情報をもつ構造体を返すヘルプファンクション(T_Arg [[▶ 302](#)])。

ファンクションF_UDINT: T_Arg

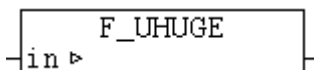
VAR_IN_OUT

```
VAR_IN_OUT
  in : UDINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.10.20 F_UHUGE



T_UHUGE_INTEGER [[▶ 305](#)]変数(符号なし128ビット整数、TwinCAT 2「レガシー」タイプ)の情報をもつ構造体を返すヘルプファンクション(T_Arg [[▶ 302](#)])。

ファンクションF_UHUGE: T_Arg

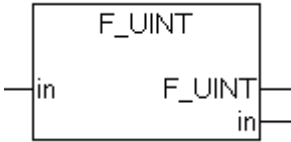
VAR_IN_OUT

```
VAR_IN_OUT
  in : T_UHUGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4. 10. 21 F_UINT



UINT変数の情報をもつ構造体を返すヘルプファンクション (T_Arg [▶ 302])。

ファンクションF_UINT: T_Arg

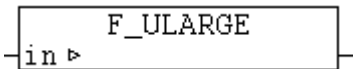
VAR_IN_OUT

```
VAR_IN_OUT
  in : UINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3. 1. 0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4. 10. 22 F_ULARGE



T_LARGE_INTEGER [▶ 306]変数 (符号付き64ビット整数、TwinCAT 2「レガシー」タイプ)の情報をもつ構造体を返すヘルプファンクション (T_Arg [▶ 302])。

ファンクションF_ULARGE: T_Arg

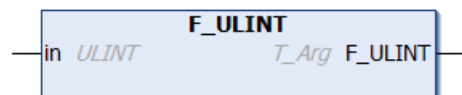
VAR_IN_OUT

```
VAR_IN_OUT
  in : T_ULARGE_INTEGER;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3. 1. 0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4. 10. 23 F_ULINT



ULINT変数の情報をもつ構造体を返すヘルプファンクション (T_Arg [▶ 302])。

ファンクションF_ULINT: T_Arg

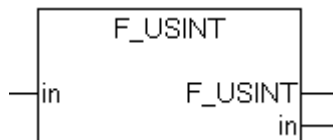
VAR_IN_OUT

```
VAR_IN_OUT
  in : ULINT;
END_VAR
```


要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4. 10. 24 F_USINT



USINT変数の情報をもつ構造体を返すヘルプファンクション([T_Arg \[▶ 302\]](#))。

ファンクションF_USINT: T_Arg

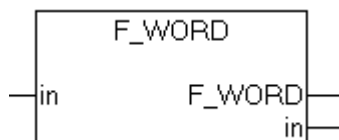
VAR_IN_OUT

```
VAR_IN_OUT
  in : USINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4. 10. 25 F_WORD



WORD型変数の情報をもつ構造体を返すヘルプファンクション([T_Arg \[▶ 302\]](#))。

ファンクションF_WORD: T_Arg

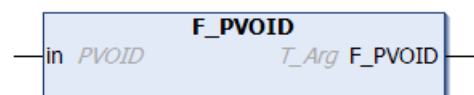
VAR_IN_OUT

```
VAR_IN_OUT
  in : WORD;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4. 10. 26 F_PVOID



BYTE型変数の情報をもつ構造体を返すヘルプファンクション([T_Arg \[▶ 302\]](#))。

ファンクションF_PVOID: T_Arg

VAR_IN_OUT

```
VAR_IN_OUT
  in : PVOID;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.10.27 IsFinite



引数が有限値をもつ場合、ファンクションIsFinite()はTRUEを返します($-\text{INF} < x < +\text{INF}$)。引数が無限値またはNaN (NaN = 非数)の場合、ファンクションはFALSEを返します。IsFinite()は、LREAL型変数またはREAL型変数の形式がIEEEに準拠しているかどうかをチェックします。

数学演算の結果が表すことのできる範囲外になる場合、INF数がランタイムシステムで発生する可能性があります。例:

```
PROGRAM MAIN
VAR
  fSingle : REAL := 12.34;
END_VAR

(*Cyclic called program code*)
fSingle := fSingle*2;
```

実際の形式(メモリの内容)が不正なアクセス(例: MEMSETファンクションのMEMCOPYの使用)によって上書きされた場合に、NaNはランタイムシステムで発生します。例:

```
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 *)
```

NaNまたはINF数をパラメータとして変換ファンクションをコールすると、PC システム(x86、x64)でFPUの例外処理を引き起こします。この例外処理は、結果としてPLCを停止します。ファンクションIsFinite()は変数の値をチェックし、そのためFPUの例外処理が回避され、プログラムの実行が継続します。

ファンクションIsFinite: BOOL

VAR_INPUT

```
VAR_INPUT
  x : T_Arg;
END_VAR
```

x: チェックされるREAL型変数またはLREAL型変数についての情報をもつ補助構造体(T_Arg [▶ 302])。IsFinite()が補助ファンクションF_REAL [▶ 221]またはF_LREAL [▶ 221]からコールされる場合は、構造体パラメータが生成される必要があり、パラメータとして送信される必要があります。

例1:

以下の例では、REAL型変数およびLREAL型変数の形式がチェックされ、FPUの例外処理は回避されます。

```

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

```

例2:

以下の場合では、FPUの例外処理はIsFinite()によるチェックによっては回避できません。

```

PROGRAM MAIN
VAR
  bigFloat    : LREAL := 3.0E100;
  smallDigit  : INT;
END_VAR

IF IsFinite( F_LREAL( bigFloat ) ) THEN
  smallDigit := LREAL_TO_INT( bigFloat );
END_IF

```

bigFloat変数は正しくフォーマットされていますが、変数値をINT型に変換するには大きすぎます。例外処理はPCシステム(x86、x64)でトリガされ、ランタイムシステムは停止します。

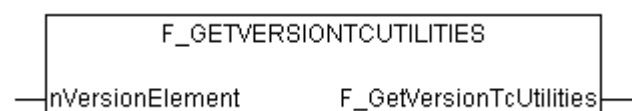
要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.11 [廃止]

4.11.1 ファンクションF_GetVersionTcUtilities

このファンクションは廃止されているため、使用しないでください。グローバル定数、stLibVersion_Tc2_Uutilitiesを使用してPLCライブラリバージョン [\[▶ 308\]](#) 情報を読み込んでください。



このファンクションを使用して、PLCライブラリのバージョン情報を読み込みできます。

ファンクションF_GetVersionTcUtilities : UINT

VAR_INPUT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement: 読み込まれるバージョン要素です。使用可能なパラメータ:

- ・ 1 : メジャー番号
- ・ 2 : マイナー番号
- ・ 3 : リビジョン番号

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.11.2 FLOATIsFinite

● 廃止ファンクション



かわりに、[LrealIsFinite \[▶_178\]](#) () ファンクションを使用してください。

ファンクションFLOATIsFinite: BOOL

VAR_INPUT

```
VAR_INPUT
    x : LREAL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.11.3 FLOATIsNaN

● 廃止ファンクション



かわりに、[LrealIsNaN \[▶_178\]](#) () ファンクションを使用してください。

ファンクションFLOATIsNaN: BOOL

VAR_INPUT

```
VAR_INPUT
    x : LREAL;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.12 ARG_TO_CSVFIELD

ARG_TO_CSVFIELD	
in T_Arg	UDINT ARG_TO_CSVFIELD
bQM BOOL	
pOutput POINTER TO BYTE	
cbOutput UDINT	

ファンクションは、PLC変数の値をCSV形式のデータフィールドに変換します。ソースファイルの単一引用符は、二重引用符と置換されます。bQMパラメータ(QM = 引用符)がセットされている場合、外側の引用符(CSVデータフィールドの周囲)も追加されます。成功した場合、ファンクションは変換されたデータ長を結果として返します。変換エラーが発生した場合、またはデータが欠落している場合は、ファンクションはゼロを返します。結果は、提供されているバイトバッファに書き込まれます。アプリケーションは、バッファが結果を格納するのに十分な大きさがあることを確認する必要があります。

ファンクションは、通常、ファンクションブロックFB_CSVMemBufferWriter [▶ 41]と共に使用し、PLCメモリにCSV形式でデータセットを生成します。次のステップで、メモリの内容はファイルに書き込みできます。STRING_TO_CSVFIELD [▶ 270]ファンクションとは対照的に、このファンクションを使用して、バイナリデータをもつPLC変数をCSVデータフィールドに変換できます。

ファンクションARG_TO_CSVFIELD: UDINT

VAR_INPUT

```
VAR_INPUT
  in      : T_Arg;
  bQM     : BOOL;
  pOutput : POINTER TO BYTE;
  cbOutput : UDINT;
END_VAR
```

in: 値をCSV形式のデータフィールドに変換するPLCソース変数(T_Arg [▶ 302])。

bQM: この入力がTRUEの場合、変換されたフィールドデータは引用符で囲まれます。

pOutput: 出力バッファの開始アドレス(ポインタ)。バッファアドレスは、ADR演算子で判定できます。演算結果のデータは、このバッファに書き込まれます。

cbOutput: バイト単位の出力バッファの最大使用可能サイズ。出力バッファ長はSIZEOF演算子で判定できません。

例:

以下の例で、異なる型のPLC変数はCSV形式に変換でき、その逆もまた可能であることを説明します。ARG_TO_CSVFIELD変換による結果は、バイトバッファにコピーされます(field1..field6)。CSVFIELD_TO_ARG [▶ 234]変換で、バイトバッファ(field1~field6)に格納されているソースデータの結果はTwinCAT PLC変数にコピーされます。

```
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 ), SI
ZEOF( 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 ) )
);

```

結果は(16進数文字列のバイトバッファ)は以下のようになります。

cbField1 = 3, field1 = '22 01 22'

cbField2 = 10, field2 = '22 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

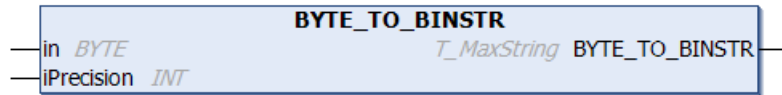
cbVar6 = 3

関連情報は、[例: CSVファイルの書き込み/読み込み \[▶ 333\]](#)に記載されています。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.13 BYTE_TO_BINSTR



このファンクションは、10進数をバイナリ文字列(2進法)に変換します。

ファンクションBYTE_TO_BINSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : BYTE;
    iPrecision  : INT;
END_VAR
```

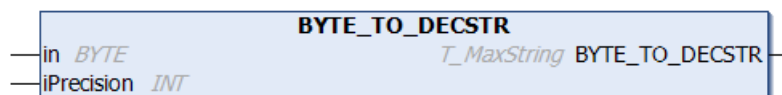
in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.14 BYTE_TO_DECSTR



このファンクションは、10進数を10進数文字列(10進法)に変換します。

ファンクションBYTE_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : BYTE;
    iPrecision  : INT;
END_VAR
```

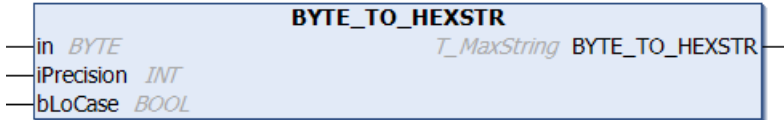
in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.15 BYTE_TO_HEXSTR



このファンクションは、10進数を16進数文字列(16進法)に変換します。

ファンクションBYTE_TO_HEXSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : BYTE;
    iPrecision  : INT;
    bLoCase    : BOOL := FALSE;
END_VAR
```

in: 変換を必要とする10進数。

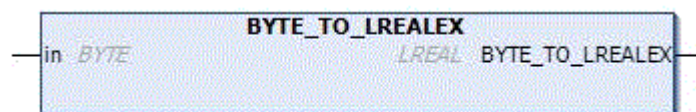
iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

bLoCase: このパラメータは、変換時に小文字または大文字が使われているかどうかを判定します。FALSE → 「ABCDEF」、TRUE → 「abcdef」。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.16 BYTE_TO_LREALX



符号なし数のLREAL型浮動小数点数への変換は、ARMプラットフォームのTwinCAT 2ではサポートしていません。最上位の有効ビットセットをもつ符号なし数は、暗黙的に負の浮動小数点数に変換される可能性があります。ここで説明したファンクションにより、明示的にBYTE型をTwinCAT 2のLREAL型の正の浮動小数点数に変換できます(最上位の有効ビットがセットされ、コンパイラ警告がない場合でも)。変換されたTwinCAT 2プロジェクトを変更せずに、TwinCAT 3でコンパイルできるようにするためにのみ、このファンクションが必要です。

BYTE型の符号なし数は、必ず(暗黙的および明示的に)TwinCAT3で正の浮動小数点数に変換されます。このため、このファンクションは無視できます。

ファンクションBYTE_TO_LREAL: LREAL

VAR_INPUT

```
VAR_INPUT
  in : BYTE;
END_VAR
```

サンプル:

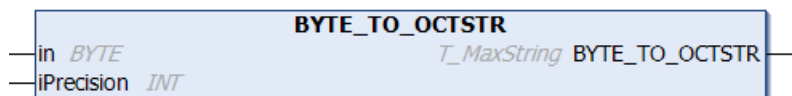
```
PROGRAM MAIN
VAR
  nByte : BYTE := 16#FF;
  fLreal : LREAL := 0.0;
END_VAR

(* 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 *)
(* -: not supported *)
(* TwinCAT 2.x Warning 1105*:warning: Conversion of unsigned integer to LREAL is not supported. The
value is used as signed instead. *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.17 BYTE_TO_OCTSTR



このファンクションは、10進数を8進数文字列(8進数)に変換します。

ファンクションBYTE_TO_OCTSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
  in : BYTE;
  iPrecision : INT;
END_VAR
```

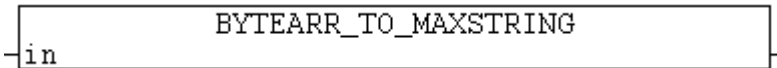
in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.18 BYTEARR_TO_MAXSTRING



バイト列の個々のASCIIコードを文字列に変換します。

ファンクションBYTEARR_TO_MAXSTRING: T_MaxString

VAR_INPUT

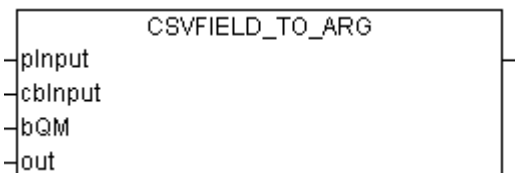
```
VAR_INPUT
    in : ARRAY[0..MAX_STRING_LENGTH] OF BYTE;
END_VAR
```

in: バイト配列変数(MAX_STRING_LENGTHデフォルト値: 255)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.19 CSVFIELD_TO_ARG



ファンクションは、バイトバッファとして存在するCSV形式 のデータフィールドからの値をPLC変数に変換します。データフィールドの二重引用符は、単一引用符に置換されます。bQMパラメータ(QM = 引用符)がセットされている場合、外側の引用符(データフィールドの周囲)は入力データから削除されます。成功した場合、ファンクションは変換されたデータ長を返します。エラーの場合、または入力データ長がゼロの場合、ファンクションは値ゼロを返します。アプリケーションは、PLCターゲット変数が値を格納するのに十分な大きさがあることを確認する必要があります。

ファンクションは、通常、PLCメモリにCSV形式で保存されているデータセットを読み込む(パースする)ために、ファンクションブロックFB_CSVMemBufferReader [▶ 40]と一緒に使用されます。この操作の前に、CSVデータセットは、通常、ファイルからPLCメモリに読み込まれます。CSVFIELD_TO_STRING [▶ 235]ファンクションとは対照的に、このファンクションを使用して、バイナリデータをもつCSVデータフィールドをPLC変数に変換できます。

ファンクションCSVFIELD_TO_ARG: UDINT

VAR_INPUT

```

VAR_INPUT
  pInput    : POINTER TO BYTE;
  cbInput   : UDINT;
  bQM       : BOOL;
  out       : T_Arg;
END_VAR

```

pInput: CSV形式に変換するデータフィールドを格納するバイトバッファへの開始アドレス(ポインタ)。アドレスは、ADR演算子で判定できます。

cbInput: バイト単位で変換するデータフィールド長。長さは、SIZEOF演算子で判定できます。

bQM: この入力がTRUEの場合、囲み引用符はフィールドデータから削除されます。

out: データフィールドの値が書き込まれるPLCされるPLCターゲット変数(T_Arg [[▶ 302](#)])。

例:

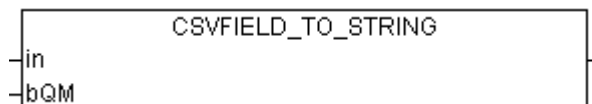
[ARG_TO_CSVFIELD](#) [[▶ 229](#)] ファンクションブロックの説明の例を参照してください。

関連情報は、[例: CSVファイルの書き込み/読み込み](#) [[▶ 333](#)]に記載されています。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.20 CSVFIELD_TO_STRING



ファンクションは、ソース文字列として存在しているCSVデータフィールド形式のデータフィールドをPLC文字列形式の値に変換します。データフィールドの二重引用符は、単一引用符に置換されます。bQMパラメータ(QM = 引用符)がセットされている場合、外側の引用符(データフィールドの周囲)はソース文字列から削除されます。成功した場合、ファンクションは変換文字列を結果として返します。変換中にエラーが発生した場合、ファンクションは空の文字列を返しますが、ソース文字列が空の文字列でなかった場合のみです。

ファンクションは、通常、PLCメモリにCSV形式で保存されているデータセットを読み込む(解釈する)ために、ファンクションブロックFB_CSMemBufferReader [[▶ 40](#)]と一緒に使用されます。この操作の前に、CSVデータセットは、通常、ファイルからPLCメモリに読み込まれます。ソース文字列はバイナリデータを格納してはいけません。値ゼロをもつバイナリデータは、間違った場所で文字列を終了し、切り捨てます。バイナリデータをもつデータフィールドを変換するには、ファンクション: [CSVFIELD_TO_ARG](#) [[▶ 234](#)]を使用してください。

ファンクションCSVFIELD_TO_STRING: T_MaxString

VAR_INPUT

```

VAR_INPUT
  in    : T_MaxString;
  bQM   : BOOL;
END_VAR

```

in: PLC文字列形式の値に変換するCSV形式 のデータフィールドをもつソース文字列(T_MaxString)。

bQM: この入力がTRUEの場合、囲み引用符はソース文字列から削除されます。

bQM	説明	ソース文字列	結果文字列	CSV準拠
FALSE	囲み引用符がないソース文字列には、文字と数字のみが含まれている必要があります。この場合、ソース文字列には、印刷できないすべての制御文字、引用符、セミコロン(;)、カンマ(,) (US CSV形式)またはバイナリデータを含めることはできません。	'Module_XA5'	'Module_XA5'	あり
		'123456'	'123456'	あり
		','	','	あり
		'A""""B'	'A""B'	なし
		'A""B'	'A""B'	なし
		',';	',';	なし
		'\$R\$N'	'\$R\$N'	なし
'AB\$00CD'	'AB' (文字列は切り捨て)	なし		
TRUE	引用符で囲まれていないソース文字列には、印刷できないすべての制御文字、引用符、セミコロン(;)、またはカンマ(,) (US CSV形式)を含めることはできません。バイナリデータが許可されていません。	""Module_XA5""	'Module_XA5'	あり
		""123456""	'123456'	あり
		""""	','	あり
		""A""""B""	'A""B'	あり
		""A""B""	'A""B'	あり
		"";""	',';	あり
		""\$R\$N""	'\$R\$N'	あり
		""AB\$00CD""	'AB' (文字列は切り捨て)	なし

例:

```
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 );
```

結果:

s1 = 'ab_\$04_\$05_cd-"ALFA"_5'

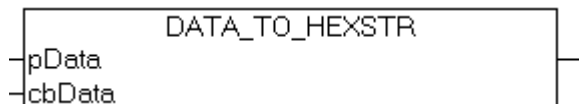
s2 = 'Module_50'

関連情報は、[例: CSVファイルの書き込み/読み込み \[▶ 333\]](#)に記載されています。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.21 DATA_TO_HEXSTR



ファンクションは、バイナリデータを16進数文字列に変換します。このファンクションを使用して、単純データ型を構造体変数に変換できます。バイナリデータの最大長は、85バイトを超過してはいけません。最大長を超えた場合、ドット('.')が結果文字列に追加され、変換は中止されます。残りのデータバイトは変換されません。欠陥のあるファンクションパラメータ (*pData* = ゼロまたは*cbData* = ゼロ) の場合、ファンクションは空の文字列を返します。

ファンクションDATA_TO_HEXSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    pData    : POINTER TO BYTE;
    cbData   : UDINT(0..85);
    bLoCase  : BOOL := FALSE;
END_VAR
```

pData: 変換するバイナリデータの開始アドレス(ポインタ)。アドレスは、ADR演算子で判定できます。

cbData: 変換するバイナリデータの最大長。長さは85バイトを超えることはできません。長さは、SIZEOF演算子で判定できます。

bLoCase: このパラメータは、変換時に小文字または大文字が使われているかどうかを判定します。TRUE = 小文字、FALSE = 大文字。

例:

*overflow*変数のデータサイズが85バイトを超えていないことを確認してください。このため、ドットが結果文字列*sH5*に追加されます。

*number*変数のバイトオーダーは、カウンタ変数のメモリ編成がリトルエンディアン形式(Intelフォーマットとも呼ばれる)に基づいているのでオーダーを入れ替えられます。

```
PROGRAM MAIN
VAR
    str      : T_MaxString := 'abcdefghijklmnopqrstuvwxyABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
    number   : DWORD := 16#BECF1234;
    char     : BYTE := 16#07;
    null     : UDINT := 0;

    overflow : ARRAY[0..86] OF BYTE; (* data overflow *)
    cbOverflow : UDINT;

    sH1, sH2, sH3, sH4, sH5 : T_MaxString;
END_VAR

sH1 := DATA_TO_HEXSTR( pData := ADR(str), cbData := LEN(str), FALSE );
sH2 := DATA_TO_HEXSTR( pData := ADR(number), cbData := SIZEOF(number), FALSE );
sH3 := DATA_TO_HEXSTR( pData := ADR(char), cbData := SIZEOF(char), FALSE );
sH4 := DATA_TO_HEXSTR( pData := ADR(null), cbData := SIZEOF(null), FALSE );
cbOverflow:= SIZEOF(overflow);
sH5 := DATA_TO_HEXSTR( pData := ADR(overflow), cbData := cbOverflow, FALSE );
```

結果:

sH1 = '61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53 54 55 56 57 58 59 5A 30 31 32 33 34 35 36 37 38 39'

sH2 = '34 12 CF BE'

sH3 = '07'

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。負の数の場合、負の符号が結果の文字列に現れます。

例:

```
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 );
```

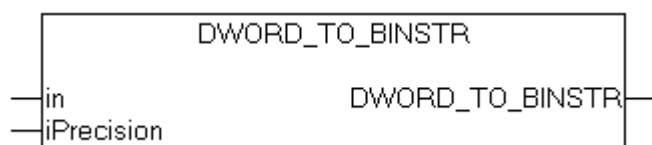
結果:

```
s1 = '-1234'
s2 = '-0000001234'
s3 = ''
s4 = '0000001234'
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.24 DWORD_TO_BINSTR



このファンクションは、10進数をバイナリ文字列(2進法)に変換します。

ファンクションDWORD_TO_BINSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
  in           : DWORD;
  iPrecision  : INT;
END_VAR
```

in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

例:

```
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 );
```

結果:

s1 = '0000000010000001'

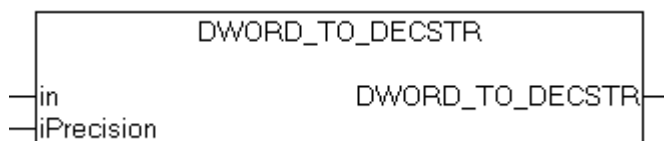
s2 = '1111'

s3 = ''

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.25 DWORD_TO_DECSTR



このファンクションは、10進数を10進数文字列(10進法)に変換します。

ファンクションDWORD_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in           : DWORD;
    iPrecision   : INT;
END_VAR
```

in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

例:

```
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 );
```

結果:

s1 = '43981'

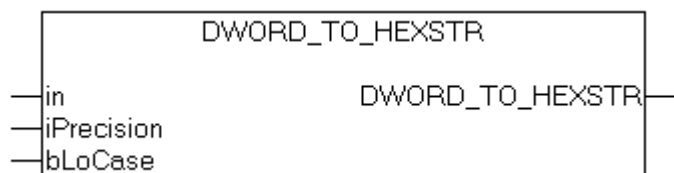
s2 = '0000043981'

s3 = ''

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.26 DWORD_TO_HEXSTR



このファンクションは、10進数を16進数文字列(16進法)に変換します。

ファンクション **DWORD_TO_HEXSTR**: T_MaxString

VAR_INPUT

```
VAR_INPUT
  in       : DWORD;
  iPrecision : INT;
  bLoCase  : BOOL;
END_VAR
```

in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が *iPrecision* パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が *iPrecision* パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision* パラメータと *in* パラメータがゼロの場合、生成される文字列は空です。

bLoCase: このパラメータは、変換時に小文字または大文字が使われているかどうかを判定します。FALSE → 「ABCDEF」、TRUE → 「abcdef」。

例:

```
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 );
```

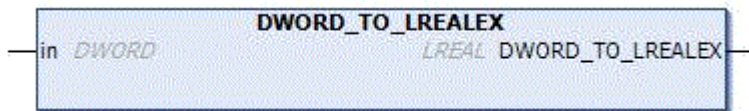
結果:

```
s1 = 'ABCD'
s2 = 'abcd'
s3 = '000F'
s4 = ''
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.27 DWORD_TO_LREALX



符号なし数のLREAL型浮動小数点数への変換は、ARMプラットフォームのTwinCAT 2ではサポートしていません。最上位の有効ビットセットをもつ符号なし数は、暗黙的に負の浮動小数点数に変換される可能性があります。ここで説明したファンクションにより、明示的にDWORD型をTwinCAT 2のLREAL型の正の浮動小数点数に変換できます(最上位の有効ビットがセットされ、コンパイラ警告がない場合でも)。変換されたTwinCAT 2プロジェクトを変更せずに、TwinCAT 3でコンパイルできるようにするためにのみ、このファンクションが必要です。

DWORD型の符号なし数は、必ず(暗黙的および明示的に)TwinCAT3で正の浮動小数点数に変換されます。このため、このファンクションは無視できます。

ファンクションDWORD_TO_LREALX: LREAL

VAR_INPUT

```
VAR_INPUT
    in : DWORD;
END_VAR
```

サンプル:

```
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#FFFFFFFF;		(*		+4294967295	
+4294967295	+4294967295	*)			
fLreal := DWORD_TO_LREALX(nDword);		(*		+4294967295	

```

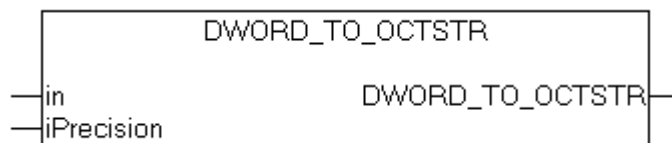
+4294967295 | +4294967295 *)
fLreal := DWORD_TO_LREALEX( DWORD#16#FFFFFFF ); (* | +4294967295 |
+4294967295 | +4294967295 *)
fLreal := DWORD_TO_LREALEX( 16#FFFFFFF ); (* | +4294967295 |
+4294967295 | +4294967295 *)
(* -: not supported *)
(* TwinCAT 2.x Warning 1105*:warning: Conversion of unsigned integer to LREAL is not supported. The
value is used as signed instead. *)

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.28 DWORD_TO_OCTSTR



このファンクションは、10進数を8進数文字列(8進数)に変換します。

ファンクションDWORD_TO_OCTSTR: T_MaxString

VAR_INPUT

```

VAR_INPUT
    in      : DWORD;
    iPrecision : INT;
END_VAR

```

in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

例:

```

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 );

```

結果:

s1 = '125715'

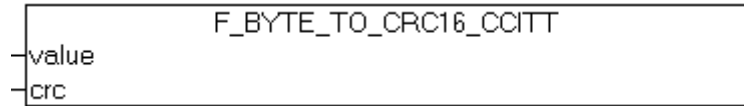
s2 = '0000125715'

s3 = ''

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.29 F_BYTE_TO_CRC16_CCITT



ファンクション「F_BYTE_TO_CRC16_CCITT」を使用して、個々のデータバイトのために16ビットCRC CCITT (巡回冗長検査)を判定できます。

使用する生成多項式: 名前: CRC-16 CCITT

- ・ 規格: CRC-CCITT
- ・ 参照: ITU X.25/T.30、ADCCP、SDLC/HDLC、...
- ・ 多項式値: 0x1021
- ・ 多項式: $x^{16} + x^{12} + x^5 + 1$

ファンクション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
  
```

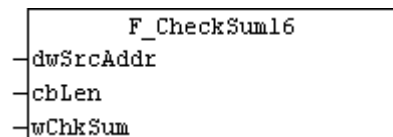
value: 変換するデータバイト。

crc: 初期値 = 16#FFFFまたは16#0000あるいは最後のCRC。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.30 F_CheckSum16



ファンクション「F_CheckSum16」を使用して、任意のデータの16ビットチェックサムを判定できます

ファンクションF_CheckSum16: WORD

VAR_INPUT

```
VAR_INPUT
  dwSrcAddr  : POINTER TO BYTE;
  cbLen      : UDINT;
  wChkSum    : WORD;
END_VAR
```

dwSrcAddr: データバッファのアドレス。

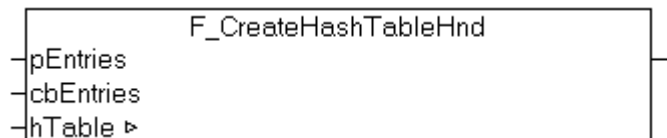
cbLen: データバッファ長。

wChkSum: 初期値 = 0または最新のチェックサム。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.31 F_CreateHashTableHnd



ファンクションはハッシュテーブルのハンドルを初期化します。ハッシュテーブルのテーブルのハンドルは、F_CreateHashTableHndファンクションをコールして一度、初期化する必要があります。

ファンクションF_CreateHashTableHnd: BOOL

VAR_INPUT

```
VAR_INPUT
  pEntries  : POINTER TO T_HashTableEntry := 0;
  cbEntries : UDINT := 0;
END_VAR
```

pEntries: 最初のT_HashTableEntry配列エレメントのアドレス。アドレスは、ADR演算子(T_HashTableEntry [▶ 303])で判定できます。

cbEntries: T_HashTableEntryバイトサイズ。バイトサイズは、sizeof演算子で判定できます。

VAR_IN_OUT

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

hTable: 初期化されるハッシュテーブルのハンドル(T_HHASHTABLE [▶ 304])。ハンドルは、ファンクションブロックFB_HashTableCtrl [▶ 71]からハッシュテーブルにアクセスするのに必要です。

戻りパラメータ	説明
TRUE	エラーはありません
FALSE	エラー

例:

参照: 例: ハッシュテーブル(FB_HashTableCtrl)。 [▶ 322]

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.32 F_CreateLinkedListHnd



ファンクションは、リンクされたリストハンドルを初期化します。リストハンドルは、F_CreateLinkedListHndファンクションをコールして一度、初期化する必要があります。

ファンクションF_CreateLinkedListHnd: BOOL

VAR_INPUT

```
VAR_INPUT
    pEntries : POINTER TO T_LinkedListEntry := 0;
    cbEntries : UDINT := 0;
END_VAR
```

pEntries: 最初の *T_LinkedListEntry* 配列エレメントのアドレス。アドレスは、ADR演算子 (*T_LinkedListEntry* [▶ 305]) で判定できます。

cbEntries: バイト単位の *T_LinkedListEntry* 配列のサイズ。バイトサイズは、SIZEOF演算子で判定できません。

VAR_IN_OUT

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

hList: 初期化されるハッシュテーブルのハンドル (*T_HLINKEDLIST* [▶ 304])。ハンドルは、ファンクションブロック *FB_LinkedListCtrl* [▶ 79] からリストにアクセスするのに必要です。

戻りパラメータ	説明
TRUE	エラーはありません
FALSE	エラー

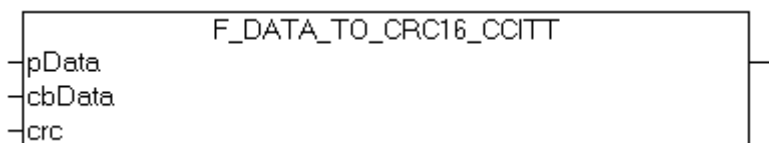
例:

参照: 例: リンクされたリスト (FB_LinkedListCtrl) [▶ 327]。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.33 F_DATA_TO_CRC16_CCITT



ファンクション「F_DATA_TO_CRC16_CCITT」を使用して、任意のデータの16ビットCRC CCITT（巡回冗長検査）を判定できます。ファンクション [F_BYTE_TO_CRC16_CCITT \[▶ 244\]](#) は、内部的に使用されます。

使用するアルゴリズムについては、[F_BYTE_TO_CRC16_CCITT \[▶ 244\]](#) ファンクションの説明に記載されています。

ファンクション [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: データバッファのアドレス。

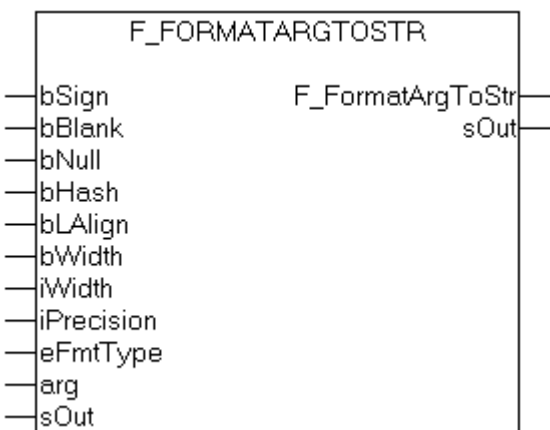
cbData: データバッファ長。

crc: 初期値 = 16#FFFFまたは16#0000あるいは最新のCRC。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.34 F_FormatArgToStr



予備フォーマットファンクション。このファンクションは、内部的に [FB_FormatString \[▶ 55\]](#) ファンクションブロックによって使用されます。ファンクションを使用して、タイプ [T_Arg \[▶ 302\]](#) の変数を [フォーマット仕様 \[▶ 338\]](#) に従ったフォーマット文字列に変換できます。

ファンクション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: 符号フラグ。

bBlank: ブランクフラグ。

bNull: ナルフラグ。

bHash: ハッシュプレフィックスフラグ。

bLAlign: 調整フラグ (TRUE=左整列)。

bWidth: TRUEの場合、iWidthパラメータは解釈されます。そうでない場合、解釈されません。

iWidth: 幅パラメータ。

iPrecision: 精度パラメータ。

eFmtType: タイプパラメータ (E_TypeFieldParam [▶ 289]型)。

arg: フォーマットされる引数。以下の補助ファンクションを使用して、異なるタイプのPLC変数を必要なデータ型T_Arg [▶ 302]に変換できます。 [F_BYTE \[▶ 218\]](#)、[F_WORD \[▶ 225\]](#)、[F_DWORD \[▶ 219\]](#)、[F_LWORD \[▶ 221\]](#)、[F_SINT \[▶ 222\]](#)、[F_INT \[▶ 219\]](#)、[F_DINT \[▶ 218\]](#)、[F_LINT \[▶ 220\]](#)、[F_USINT \[▶ 225\]](#)、[F_UINT \[▶ 224\]](#)、[F_UDINT \[▶ 223\]](#)、[F_ULINT \[▶ 224\]](#)、[F_STRING \[▶ 222\]](#)、[F_REAL \[▶ 221\]](#)、[F_LREAL \[▶ 221\]](#)。

VAR_IN_OUT

```
VAR_IN_OUT
  sOut      : T_MaxString;
END_VAR
```

sOut: 成功した場合、この変数はフォーマットされた出力文字列 (T_MaxString型) を返します。

戻りパラメータ	意味
0	エラーはありません
<> 0	エラーです。エラー説明については、 フォーマットエラーコード [▶ 341] を参照してください。

例:

BYTE型変数をバイナリ文字列としてフォーマット。

```
PROGRAM MAIN
VAR
  s1      : T_MaxString;
  s2      : T_MaxString;
  s3      : T_MaxString;
  s4      : T_MaxString;
  s5      : T_MaxString;
  errID   : UDINT;
  varByte : BYTE;
  double  : LREAL;
END_VAR
```



```

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 );

```

結果:

```

s1 = '10000000'
s2 = '          10000000'
s3 = '10000000'
s4 = '2#10000000'
L1 = 8
L2 = 20
L3 = 20
L4 = 20

```

LREAL型変数のフォーマット。

```

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 );

```

結果:

```

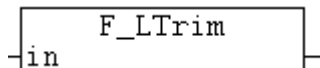
s1 = '12345.67890000'
s2 = '          12345.67890000'
s3 = '12345.67890000'
s4 = '00000012345.67890000'
s5 = '+12345.67890000'
L1 = 14
L2 = 20
L3 = 20
L4 = 20
L5 = 20

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.35 F_LTrim



文字列から先頭の空白を削除し、短縮された文字列を返します。

ファンクションF_LTrim: T_MaxString

VAR_INPUT

```
VAR_INPUT
  in : T_MaxString;
END_VAR
```

in: 変換する文字列(タイプ:T_MaxString)。

例:

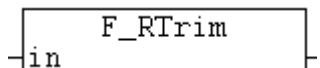
```
PROGRAM MAIN
VAR
  sLTrim : STRING;
END_VAR

sLTrim := F_LTrim(' <trim ');(* result: '<trim ' *)
sLTrim := F_LTrim(' <trim');(* result: '<trim' *)
sLTrim := F_LTrim('<trim');(* result: '<trim' *)
sLTrim := F_LTrim('');(* result: '' *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.36 F_RTrim



指定した値から後に続くすべての空白を切り捨て、結果を返します。

ファンクションF_RTrim: T_MaxString

VAR_INPUT

```
VAR_INPUT
  in : T_MaxString;
END_VAR
```

in: 変換する文字列(T_MaxString型)。

例:

```

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: '' *)

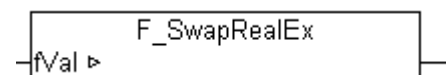
sLTrim := F_LTrim( F_RTrim( ' <trim> '));(* result: '<trim>' *)

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.37 F_SwapRealEx



バスターミナルコントローラのREAL数のメモリ表現(例: BC2000、BC3100、BC9000)は、x86/x64/ARMシステム(IPCまたは組込み型PC)のREAL数のメモリ表現とは異なっています。

バスターミナルコントローラのREAL数がIPCで適切に表現できるためには、REAL数のHiおよびLoワードを入れ替えられる必要があります。プログラミング環境の下で、これは既にオンラインまたはシミュレーションモードで実行されています。ネットワーク(ADSプロトコル、ADSDLL、Ads0cxなど)経由でバスコントローラのREALデータをリクエストし、x86/x64/ARM IPCで正しく表現できるためには、REALデータは正しいフォーマットに変換されなければいけません。これは、バスターミナルコントローラ側またはIPC側で実行できません。

ファンクションF_SwapRealExを使用して、REAL型変数(例: VBアプリケーションで読み込まれ、またはTwinCAT Scope Viewで記録される変数)を適切なPC側のフォーマットに変換できます。ファンクションは、送信された *fVal*/パラメータ (VAR_IN_OUT)のメモリ表現を変更します。

ファンクションF_SwapRealEx: BOOL

VAR_IN_OUT

```

VAR_IN_OUT
    fVal : REAL;
END_VAR

```

fVal: 変換するREAL値。

戻りパラメータ	意味
TRUE	エラーはありません
FALSE	ファンクション実行中のエラー

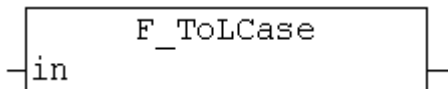
例:

例: [通信BC/BX<->PC/CX \(F_SwapRealEx\) \[▶_310\]](#)を参照してください。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.38 F_ToLCase



F_ToLCaseファンクションは、文字列のすべての大文字を小文字に変換します。

● 文字セット

i デフォルトでは、変換ファンクションはWindows code pager 1252 Latin 1、SBCS (シングルバイト文字セット)の文字セットを使用します。グローバル変数GLOBAL_SBCS_TABLEによってランタイムで異なる文字セットを選択できます(現在はWindows code page 1250 Central Europeanのみ)(例を参照)。

ファンクションF_ToLCase: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in : T_MaxString;
END_VAR
```

in: 変換する文字列(T_MaxString型)。

例:

```
PROGRAM MAIN
VAR
    sLCase : STRING;
END_VAR

sLCase := F_ToLCase( 'TO LOWER CASE 1234567890 ÄÖÜß' );
```

変換の結果: 'to lower case 1234567890 äöüß'

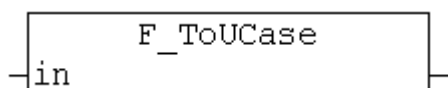
```
GLOBAL_SBCS_TABLE := eSBCS_CentralEuropean;
sLCase := F_ToLCase( 'TO LOWER CASE 1234567890 ĄĘŚĆŹŻŁÓ' );
```

変換の結果: 'to lower case 1234567890 ąęśćźżłó'

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.39 F_ToUCase



F_ToUCaseファンクションは、文字列のすべての小文字を大文字に変換します。

● 文字セット

i デフォルトでは、変換ファンクションはWindows code pager 1252 Latin 1、SBCS (シングルバイト文字セット)の文字セットを使用します。グローバル変数GLOBAL_SBCS_TABLEによってランタイムで異なる文字セットを選択できます(現在はWindows code page 1250 Central Europeanのみ)(例を参照)。

ファンクションF_ToUCase: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in : T_MaxString;
END_VAR
```

in: 変換する文字列 (T_MaxString型)。

例:

```
PROGRAM MAIN
VAR
    sUCase : STRING;
END_VAR
sUCase := F_ToUCase( 'to upper case 1234567890 äöüß' );
```

変換の結果: ' TO UPPER CASE 1234567890 ÄÖÜß'

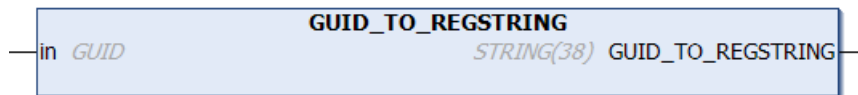
```
GLOBAL_SBCS_TABLE := eSBCS_CentralEuropean;
sUCase := F_ToUCase( 'to upper case 1234567890 aęśćźżłó' );
```

変換の結果: ' TO UPPER CASE 1234567890 AĘŚĆŹŻŁÓ'

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.40 GUID_TO_REGSTRING



このファンクションは、構造化したGUID [▶_290]変数をレジストリGUID文字列変数に変換します(波括弧で囲まれています)。

ファンクションGUID_TO_REGSTRING: STRING(38)

VAR_INPUT

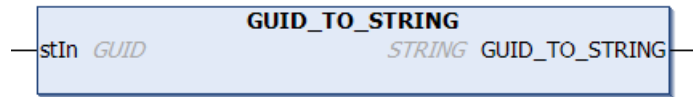
```
VAR_INPUT
    in : GUID;
END_VAR
```

戻り値	意味
' {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}'	エラーはありません(「x」は16進数のハーフバイト)
' {00000000-0000-0000-0000-000000000000}'	エラーはありません。GUIDに初期値があります(すべてのバイトはゼロ)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.41 GUID_TO_STRING



このファンクションは、構造化したGUID [▶ 290]変数をGUID文字列変数(波括弧なし)に変換します。

ファンクションGUID_TO_STRING: STRING

VAR_INPUT

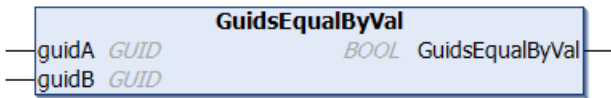
```
VAR_INPUT
    stIn : GUID;
END_VAR
```

戻り値	意味
'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX'	エラーはありません(「x」は16進数のハーフバイト)
00000000-0000-0000-0000-000000000000'	エラーはありません。GUIDに初期値があります(すべてのバイトはゼロ)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.42 GuidEqualByVal



このファンクションは2つのGUID値を比較します。

ファンクションGuidEqualByVal: BOOL

VAR_INPUT

```
VAR_INPUT
    guidA : GUID;
    guidB : GUID;
END_VAR
```

戻り値	意味
FALSE	guidA <> guidB
TRUE	guidA = guidB

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.43 HEXASCNIBBLE_TO_BYTE

```

      HEXASCNIBBLE_TO_BYTE
asc BYTE                               BYTE HEXASCNIBBLE_TO_BYTE

```

このファンクションは、16進ハーフバイト文字のASCIIコードを10進数に変換します。

ファンクションHEXASCNIBBLE_TO_BYTE: BYTE

VAR_INPUT

```

VAR_INPUT
  asc : BYTE;
END_VAR

```

asc: 16進ハーフバイト文字のAsciiコード(ASCIIコード: 「0」から「9」または「a」から「f」または「A」から「F」)。

戻り値	意味
0~15	成功、エラーはありません。
255	エラー、誤った入力パラメータ値。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.44 HEXCHRNIBBLE_TO_BYTE

```

      HEXCHRNIBBLE_TO_BYTE
chr STRING(1)                           BYTE HEXCHRNIBBLE_TO_BYTE

```

このファンクションは、16進ハーフバイト文字を10進数に変換します。

ファンクションHEXCHRNIBBLE_TO_BYTE: BYTE

VAR_INPUT

```

VAR_INPUT
  chr : STRING(1);
END_VAR

```

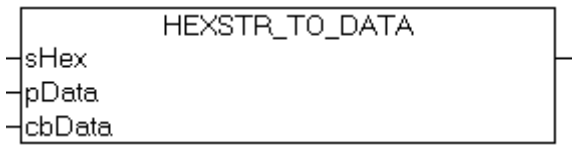
chr: 16進ハーフバイト文字(「0」～「9」または「a」～「f」、または「A」～「F」)。

戻り値	意味
0~15	成功、エラーはありません。
255	エラー、誤った入力パラメータ値。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.45 HEXSTR_TO_DATA



ファンクションは16進数文字列をバイナリデータに変換し、変換に成功したデータバイト数を結果として返します。変換する16進数文字列では、空白のみが区切り文字として使用できます。小文字と大文字が16進文字として許可されています。エラーまたは不正な文字の場合、変換は中止され、ゼロ長が結果として返されます。

ファンクションHEXSTR_TO_DATA: UDINT

VAR_INPUT

```
VAR_INPUT
    sHex    : T_MaxString;
    pData   : POINTER TO BYTE;
    cbData  : UDINT;
END_VAR
```

sHex: 変換する16進数文字列 (T_MaxString型、例: 'AB CD 01 23')。

pData: 変換されたデータバイトが書き込まれるターゲットバッファへの開始アドレス (ポインタ)。アドレスは、ADR演算子で判定できます。

cbData: ターゲットバッファの最大使用可能長。長さは、sizeof演算子で判定できます。

例:

```
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 ) );
```

結果(オンライン):

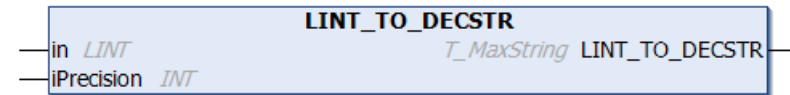
Expression	Type	Value	Prepared value	Address	Comment
sH	STRING	'AB CD EF 01 23 45 67 89'			
data	ARRAY [0..10] OF BYTE				
data[0]	BYTE	16#AB			
data[1]	BYTE	16#CD			
data[2]	BYTE	16#EF			
data[3]	BYTE	16#01			
data[4]	BYTE	16#23			
data[5]	BYTE	16#45			
data[6]	BYTE	16#67			
data[7]	BYTE	16#89			
data[8]	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: 'ABCD EF 0', ADR( data ), sizeof( data ) );RETURN
```


要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.46 LINT_TO_DECSTR



このファンクションは、符号付き10進数を10進数文字列(10進法)に変換します。

ファンクションLINT_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in      : LINT;
    iPrecision : INT;
END_VAR
```

in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。負の数の場合、負の符号が結果の文字列に現れます。

例:

```
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 );
```

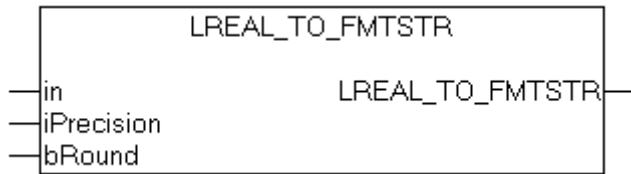
結果:

```
s1 = '-1234'
s2 = '-0000001234'
s3 = ''
s4 = '0000001234'
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.47 LREAL_TO_FMTSTR



ファンクションは、浮動小数点数をフォーマット[-]dddd.dddd (ddddは10進数)の文字列変数に変換し、フォーマットします。小数点の前の桁数は、浮動小数点数の値によります。小数点の後の桁数は、必要な精度によります。符号は、負の値に対してのみ表示されます。'#INF'が無限の正の値のために返され、'-#INF'が無限の負の値のために返されます。送信された変数は、不正な値(NaN、非数)を保持している場合、'#QNAN'または'-#QNAN'が返されます。フォーマット文字列の長さが生成される文字列の最大許容長を超過する場合、'#OVF'または'-#OVF'が返されます。

ファンクションLREAL_TO_FMTSTR: STRING(510)

VAR_INPUT

```
VAR_INPUT
  in      : LREAL;
  iPrecision : INT;
  bRound  : BOOL;
END_VAR
```

in: 変換され、フォーマットされる浮動小数点数。

iPrecision: 精度。値は小数点の後の桁数を判定します。最小値(ゼロ)の場合、小数位は表示されません。*iPrecision*の最大値は、小数点の前の桁数と結果文字列の最大許容長によって制限されます。*in* = 0と*iPrecision* = 0の場合、文字列「0」が返されます。

bRound: *bRound*パラメータがセットされている場合、フォーマット文字列がそれぞれの少数点以下の桁数に丸められます(*iPrecision*)。次のルールが丸めに適用されます。最後に必要な小数位が5以上の後の小数は、値が丸められ、そうでない場合は丸められません。

例1:

0.46523は2つの小数位をもつ文字列に変換され、丸められます。

```
sOut := LREAL_TO_FMTSTR( 0.46523, 2, TRUE );
```

結果は、「0.47」です。

例:2

● LREAL型変数の有効桁の最大数は15に制限されます。

I 浮動小数点数の内部表現と変換中の丸め誤差のために、生成される文字列は正確に*in*変数の値に一致しないことがあります。

```
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 );
```


s3 = ''

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.49 LWORD_TO_DECSTR

LWORD_TO_DECSTR	
in LWORD	T_MaxString LWORD_TO_DECSTR
iPrecision INT	

このファンクションは、10進数を10進数文字列(10進法)に変換します。

ファンクションLWORD_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in      : LWORD;
    iPrecision : INT;
END_VAR
```

in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数がiPrecisionパラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数がiPrecisionパラメータより大きい場合、生成される文字列は切り捨てられません iPrecisionパラメータとinパラメータがゼロの場合、生成される文字列は空です。

例:

```
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 );
```

結果:

s1 = '43981'
s2 = '0000043981'
s3 = ''

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.50 LWORD_TO_HEXSTR

LWORD_TO_HEXSTR	
in <i>LWORD</i>	<i>T_MaxString</i> LWORD_TO_HEXSTR
iPrecision <i>INT</i>	
bLoCase <i>BOOL</i>	

このファンクションは、10進数を16進数文字列(16進法)に変換します。

ファンクションLWORD_TO_HEXSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
  in          : LWORD;
  iPrecision  : INT;
  bLoCase    : BOOL;
END_VAR
```

in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

bLoCase: このパラメータは、変換時に小文字または大文字が使われているかどうかを判定します。FALSE → 「ABCDEF」、TRUE → 「abcdef」。

例:

```
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 );
```

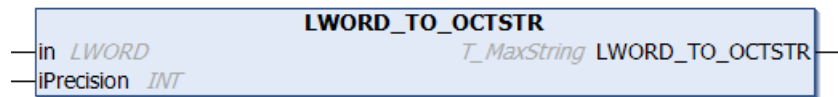
結果:

```
s1 = 'ABCD'
s2 = 'abcd'
s3 = '000F'
s4 = ''
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.51 LWORD_TO_OCTSTR



このファンクションは、10進数を8進数文字列(8進数)に変換します。

ファンクションLWORD_TO_OCTSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in      : LWORD;
    iPrecision : INT;
END_VAR
```

in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

例:

```
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 );
```

結果:

s1 = '125715'

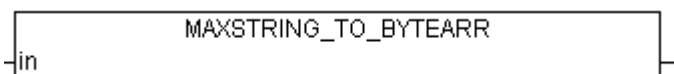
s2 = '0000125715'

s3 = ''

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.52 MAXSTRING_TO_BYTEARR



文字列をバイト列の個々のASCIIコードに変換します。

ファンクションMAXSTRING_TO_BYTEARR: ARRAY[0..MAX_STRING_LENGTH] OF BYTE

VAR_INPUT

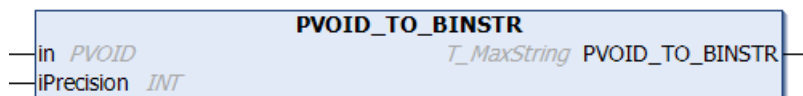
```
VAR_INPUT
  in : T_MaxString;
END_VAR
```

in: 変換する文字列 (T_MaxString型)。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.53 PVOID_TO_BINSTR



ファンクションは、タイプPVOIDのポインタ変数の値をバイナリ文字列 (2進数) に変換します。

ファンクションPVOID_TO_BINSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
  in          : PVOID;
  iPrecision  : INT;
END_VAR
```

in: 変換するポインタ変数。

iPrecision: 表示される数字の最小桁 (桁数)。実際の有効桁数が *iPrecision* パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が *iPrecision* パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision* パラメータと *in* パラメータがゼロの場合、生成される文字列は空です。

例:

```
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 );
```

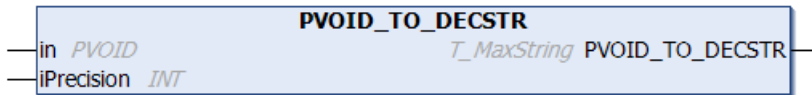
結果:

```
s1 = ''
s2 = '0'
s3 = '00000000000000000000000000000000'
s4 = '10000111110111100000001001010101' (変わることがあります)
s5 = '10000111110111100000001001010101' (変わることがあります)
s6 = '10000111110111100000001001010101' (変わることがあります)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.54 PVOID_TO_DECSTR



ファンクションは、タイプPVOIDのポインタ変数の値を10進数文字列(10進数)に変換します。

ファンクションPVOID_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in      : PVOID;
    iPrecision : INT;
END_VAR
```

in: 変換するポインタ変数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

例:

```
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 );
```


結果:

s1 = ''

s2 = '0'

s3 = '0000000000000000'

s4 = '2279473749' (変わることがあります)

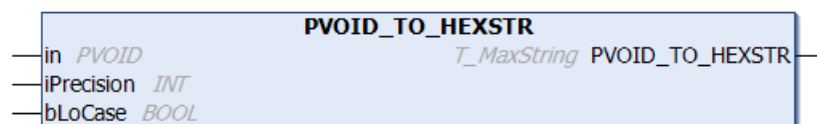
s5 = '2279473749' (変わることがあります)

s6 = '0000002279473749' (変わることがあります)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.55 PVOID_TO_HEXSTR



ファンクションは、タイプPVOIDのポインタ変数の値を16進数文字列(16進数)に変換します。

ファンクションPVOID_TO_HEXSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
  in          : PVOID;
  iPrecision  : INT;
  bLoCase    : BOOL;
END_VAR
```

in: 変換するポインタ変数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

bLoCase: このパラメータは、変換時に小文字または大文字が使われているかどうかを判定します。FALSE → 「ABCDEF」、TRUE → 「abcdef」。

例:

```
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 );

```

結果:

```

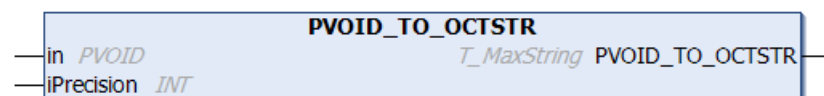
s1 = ''
s2 = ''
s3 = '0'
s4 = '0'
s5 = '0000000000000000'
s6 = '0000000000000000'
s7 = '87CBC255' (変わることがあります)
s8 = '87cbc255' (変わることがあります)
s9 = '87CBC255' (変わることがあります)
s10 = '87cbc255' (変わることがあります)
s11 = '000000087CBC255' (変わることがあります)
s12 = '000000087cbc255' (変わることがあります)

```

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.56 PVOID_TO_OCTSTR



ファンクションは、タイプPVOIDのポインタ変数の値を8進数文字列(8進数)に変換します。

ファンクションPVOID_TO_OCTSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
  in      : PVOID;
  iPrecision : INT;
END_VAR
```

in: 変換するポインタ変数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

例:

```
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 );
```

結果:

s1 = ''

s2 = '0'

s3 = '0000000000000000'

s4 = '20767501125' (変わることがあります)

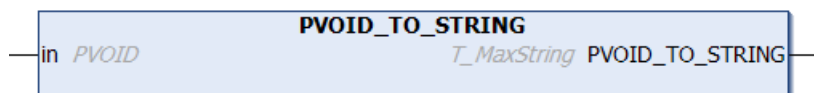
s5 = '20767501125' (変わることがあります)

s6 = '0000020767501125' (変わることがあります)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.57 PVOID_TO_STRING



ファンクションは、タイプPVOIDのポインタ変数の値を16進数文字列(16進数)に変換します。16進数文字列はPLCプレフィックス、「16#」を含みます。精度は32ビットシステムでは8桁に固定され、64ビットシステムでは16桁に固定されています。

ファンクションPVOID_TO_STRING: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in : PVOID;
END_VAR
```

in: 変換するポインタ変数。

例:

```
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 );
```

32ビットシステムの結果:

```
s1 = '16#00000000'
s2 = ''16#87DE0255' (変わることがあります)
```

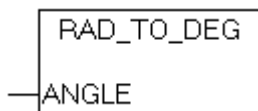
64ビットシステムの結果:

```
s1 = '16#0000000000000000'
s2 = ''16#8734651087DE0255' (変わることがあります)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4. 58 RAD_TO_DEG



ファンクションはラジアンを度単位の角度に変換します。

ファンクションRAD_TO_DEG: LREAL

VAR_INPUT

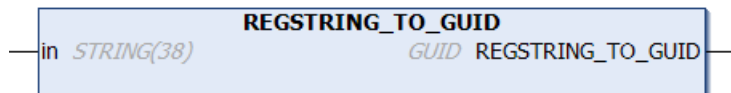
```
VAR_INPUT
    ANGLE : LREAL;
END_VAR
```

ANGLE: 変換するラジアン。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.59 REGSTRING_TO_GUID



このファンクションは、レジストリGUID文字列変数(波括弧で囲まれている)を構造化したGUID [▶ 290]変数に変換します。

ファンクションREGSTRING_TO_GUID: GUID

VAR_INPUT

```
VAR_INPUT
    in : STRING(38);
END_VAR
```

戻り値	意味
' {xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}'	エラーはありません(「x」は16進数のハーフバイト)
' {00000000-0000-0000-0000-000000000000}'	エラーはありません。GUIDに初期値があります(すべてのバイトはゼロ)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.60 ROUTETRANSPORT_TO_STRING



ファンクションは、AMSメッセージルータトランスポート層IDを文字列に変換します。

ファンクションROUTETRANSPORT_TO_STRING: STRING

VAR_INPUT

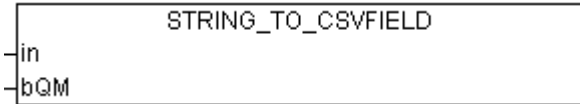
```
VAR_INPUT
    eType : E_RouteTransportType;
END_VAR
```

eType: 変換するトランスポート層ID (E_RouteTransportType [▶ 287]型)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.61 STRING_TO_CSVFIELD



ファンクションは、PLC文字列変数の値をCSV形式の文字列データフィールドに変換します。ソース文字列の単一引用符は、二重引用符に置換されます。bQMパラメータ(QM = 引用符)がセットされている場合、外側の引用符(CSVデータフィールドの周囲)も追加されます。成功した場合、ファンクションは変換文字列を結果として返します。変換中にエラーが発生した場合、ファンクションは空の文字列を返しますが、ソース文字列が空の文字列でなかった場合のみです。

ファンクションは、通常、ファンクションブロックFB_CSVMemBufferWriter [▶ 41]と共に使用し、PLCメモリにCSV形式でデータセットを生成します。次のステップで、メモリの内容はファイルに書き込みできません。

ソース文字列はバイナリデータを格納してはいけません。値ゼロをもつバイナリデータは、間違った場所で文字列を終了し、切り捨てます。バイナリデータに変換するためには、ファンクションARG_TO_CSVFIELD [▶ 229]を使用してください。

ファンクションSTRING_TO_CSVFIELD: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in : T_MaxString;
    bQM : BOOL;
END_VAR
```

in: 値をCSV形式のデータフィールドに変換するソース文字列(T_MaxString型)。

bQM: この入力がTRUEの場合、囲み引用符が結果文字列から追加されます。

bQM	説明	ソース文字列	結果文字列	CSV準拠
FALSE	囲み引用符がないソース文字列には、文字と数字のみが含まれている必要があります。この場合、ソース文字列には、印刷できないすべての制御文字、引用符、セミコロン (;)、カンマ (,) (US CSV形式) またはバイナリデータを含めてはいけません。	'Module_XA5'	'Module_XA5'	あり
		'123456'	'123456'	あり
		' '	' '	あり
		'A"B'	'A""B'	なし
		'A"B'	'A""B'	なし
		' ;'	' ;'	なし
		' \$R\$N'	' \$R\$N'	なし
		'AB\$00CD'	'AB' (文字列は切り捨て)	なし
TRUE	引用符で囲まれていないソース文字列には、印刷できないすべての制御文字、引用符、セミコロン (;)、またはカンマ (,) (US CSV形式) を含めることはできません。バイナリデータが許可されていません。	'Module_XA5'	'"Module_XA5"'	あり
		'123456'	'"123456"'	あり
		' '	'" "'	あり
		'A"B'	'"A""B"'	あり
		'A"B'	'"A""B"'	あり
		' ;'	'" ;"'	あり

bQM	説明	ソース文字列	結果文字列	CSV準拠
	'\$R\$N'	'"\$R\$N"'	あり	
	'AB\$00CD'	'"AB"' (文字列は切り捨て)	なし	

例:

```
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 );
```

結果:

s1 = '"Module_"ALFA_\$05"_6"'

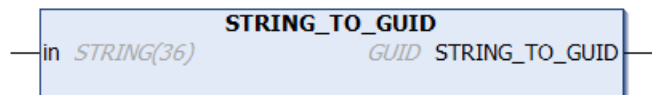
s2 = 'Module_50'

関連情報は、例: CSVファイルの書き込み/読み込み [▶ 333]に記載されています。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4.62 STRING_TO_GUID



このファンクションは、レジストリGUID文字列変数(波括弧なし)を構造化したGUID [▶ 290]変数に変換します。

ファンクションSTRING_TO_GUID: GUID

VAR_INPUT

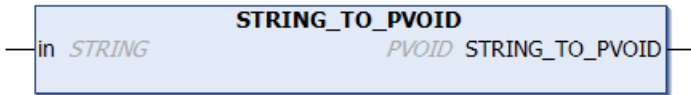
```
VAR_INPUT
  in : STRING(36);
END_VAR
```

戻り値	意味
'XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX'	エラーはありません(「x」は16進数のハーフバイト)
'00000000-0000-0000-0000-000000000000'	エラーはありません。GUIDに初期値がありません(すべてのバイトはゼロ)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

4. 63 STRING_TO_PVOID



ファンクションは、文字列変数をタイプPVOIDのポインタ変数に変換します。入力文字列が不正な文字を格納し、アドレスとして解釈できない場合、ファンクションはゼロを返します。

ファンクションSTRING_TO_PVOID: PVOID

VAR_INPUT

```
VAR_INPUT
  in : STRING;
END_VAR
```

in: 変換する文字列変数。

例:

```
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;
EEND_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 );
```

結果:

pP1 = 2301908600

pP2 = 2301908600

pP3 = 2301908600

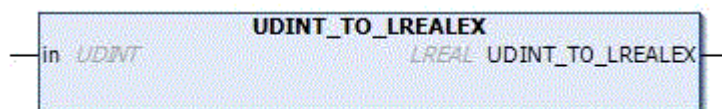
pP4 = 2301908600

pP5 = 0

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3. 1. 0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4. 64 UDINT_TO_LREAL



符号なし数のLREAL型浮動小数点数への変換は、ARMプラットフォームのTwinCAT 2ではサポートしていません。最上位の有効ビットセットをもつ符号なし数は、暗黙的に負の浮動小数点数に変換される可能性があります。ここで説明したファンクションにより、明示的にUDINT型をTwinCAT 2のLREAL型の正の浮動小数点数に変換できます(最上位の有効ビットがセットされ、コンパイラ警告がない場合でも)。変換されたTwinCAT 2プロジェクトを変更せずに、TwinCAT 3でコンパイルできるようにするためにのみ、このファンクションが必要です。

UDINT型の符号なし数は、必ず(暗黙的および明示的に)TwinCAT3で正の浮動小数点数に変換されます。このため、このファンクションは無視できます。

ファンクションUDINT_TO_LREALEX: LREAL

VAR_INPUT

```
VAR_INPUT
    in : UDINT;
END_VAR
```

サンプル:

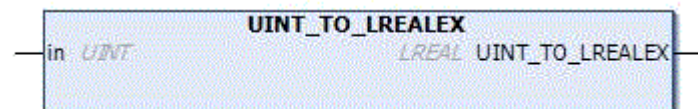
```
PROGRAM MAIN
VAR
    nUdint : UDINT := 16#FFFFFFFF;
    fLreal : LREAL := 0.0;
END_VAR

(* fLreal value | Tc2.x ARM |
Tc2.x X86 | Tc3.x ARM, X86, X64 *)
(* _____ *)
fLreal := nUdint; (* | -1, Warning 1105* |
+4294967295 | +4294967295 *)
fLreal := UDINT_TO_LREAL( nUdint ); (* | -1, Warning 1105* |
+4294967295 | +4294967295 *)
fLreal := UDINT#16#FFFFFFFF; (* | -1, Warning 1105* |
+4294967295 | +4294967295 *)
fLreal := 16#FFFFFFFF; (* | +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#FFFFFFFF ); (* | +4294967295 |
+4294967295 | +4294967295 *)
(* -: not supported *)
(* TwinCAT 2.x Warning 1105*:warning: Conversion of unsigned integer to LREAL is not supported. The
value is used as signed instead. *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.65 UINT_TO_LREALEX



符号なし数のLREAL型浮動小数点数への変換は、ARMプラットフォームのTwinCAT 2ではサポートしていません。最上位の有効ビットセットをもつ符号なし数は、暗黙的に負の浮動小数点数に変換される可能性があります。ここで説明したファンクションにより、明示的にUINT型をTwinCAT 2のLREAL型の正の浮動小数点数に

変換できます(最上位の有効ビットがセットされ、コンパイラ警告がない場合でも)。変換されたTwinCAT 2プロジェクトを変更せずに、TwinCAT 3でコンパイルできるようにするためにのみ、このファンクションが必要です。

UINT型の符号なし数は、必ず(暗黙的および明示的に)TwinCAT3で正の浮動小数点数に変換されます。このため、このファンクションは無視できます。

ファンクションUINT_TO_LREAL: LREAL

VAR_INPUT

```
VAR_INPUT
    in : UINT;
END_VAR
```

サンプル:

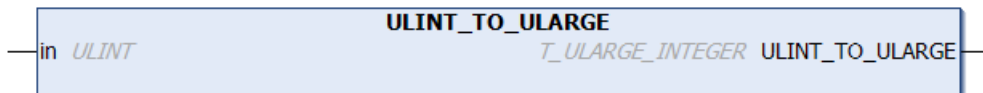
```
PROGRAM MAIN
VAR
    nUInt : UINT := 16#FFFF;
    fLreal : LREAL := 0.0;
END_VAR
```

```
(* 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_LREAL( nUInt ); (* | +65535 |
+65535 | +65535 *)
fLreal := UINT_TO_LREAL( UINT#16#FFFF ); (* | +65535 |
+65535 | +65535 *)
fLreal := UINT_TO_LREAL( 16#FFFF ); (* | +65535 |
+65535 | +65535 *)
(* -: not supported *)
(* TwinCAT 2.x Warning 1105*:warning: Conversion of unsigned integer to LREAL is not supported. The
value is used as signed instead. *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

4.66 ULINT_TO_ULONG



ファンクションは、TwinCAT 3の符号なし64ビット数(「ネイティブ」タイプ)をTwinCAT 2の符号なし64ビット数(「レガシー」T_ULONG_INTEGER [▶ 306]型)に変換します。

ファンクションULINT_TO_ULARGE: T_ULARGE_INTEGER

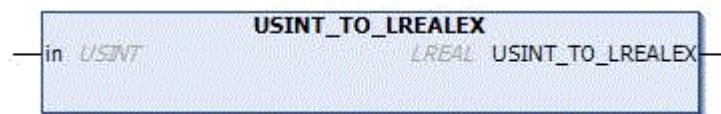
VAR_INPUT

```
VAR_INPUT
    in : ULINT;
END_VAR
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.67 USINT_TO_LREALX



符号なし数のLREAL型浮動小数点数への変換は、ARMプラットフォームのTwinCAT 2ではサポートしていません。最上位の有効ビットセットをもつ符号なし数は、暗黙的に負の浮動小数点数に変換される可能性があります。ここで説明したファンクションにより、明示的にUSINT型をTwinCAT 2のLREAL型の正の浮動小数点数に変換できます(最上位の有効ビットがセットされ、コンパイラ警告がない場合でも)。変換されたTwinCAT 2プロジェクトを変更せずに、TwinCAT 3でコンパイルできるようにするためにのみ、このファンクションが必要です。

USINT型の符号なし数は、必ず(暗黙的および明示的に)TwinCAT3で正の浮動小数点数に変換されます。このため、このファンクションは無視できます。

ファンクションUSINT_TO_LREALX: LREAL

VAR_INPUT

```
VAR_INPUT
    in : USINT;
END_VAR
```

サンプル:

```
PROGRAM MAIN
VAR
    nUsint : USINT := 16#FF;
    fLreal : LREAL := 0.0;
END_VAR

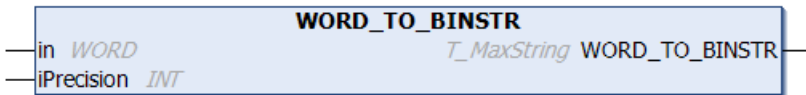
(* fLreal value | Tc2.x X86 | Tc3.x ARM, X86, X64 *)
(* Tc2.x X86 | Tc3.x ARM, X86, X64 *)
(* *)
fLreal := nUsint; (* | +255, Warning 1105* |
+255 | +255 *)
fLreal := USINT_TO_LREAL( nUsint ); (* | +255, Warning 1105* |
+255 | +255 *)
fLreal := USINT#16#FF; (* | +255, Warning 1105* |
+255 | +255 *)
fLreal := 16#FF; (* | +255 |
+255 | +255 *)
fLreal := USINT_TO_LREALX( nUsint ); (* | +255 |
+255 | +255 *)
fLreal := USINT_TO_LREALX( USINT#16#FF ); (* | +255 |
+255 | +255 *)
fLreal := USINT_TO_LREALX( 16#FF ); (* | +255 |
+255 | +255 *)
```

```
(* -: not supported *)
(* TwinCAT 2.x Warning 1105*:warning: Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead. *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.68 WORD_TO_BINSTR



このファンクションは、10進数をバイナリ文字列(2進法)に変換します。

ファンクションWORD_TO_BINSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : WORD;
    iPrecision  : INT;
END_VAR
```

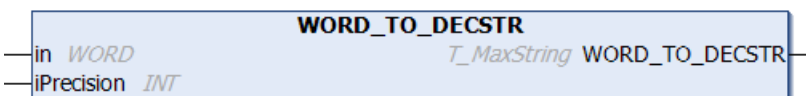
in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.69 WORD_TO_DECSTR



このファンクションは、10進数を10進数文字列(10進法)に変換します。

ファンクションWORD_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : WORD;
    iPrecision  : INT;
END_VAR
```

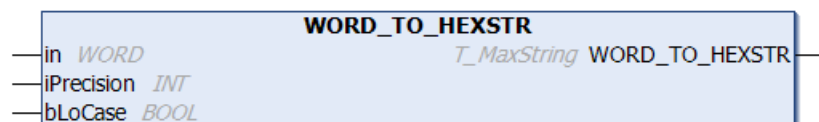
in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.70 WORD_TO_HEXSTR



このファンクションは、10進数を16進数文字列(16進法)に変換します。

ファンクションWORD_TO_HEXSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
  in      : WORD;
  iPrecision : INT;
  bLoCase  : BOOL := FALSE;
END_VAR
```

in: 変換を必要とする10進数。

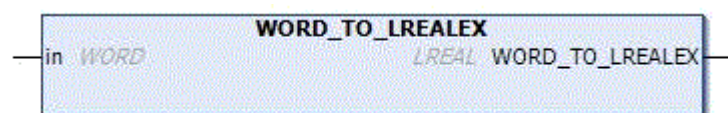
iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

bLoCase: このパラメータは、変換時に小文字または大文字が使われているかどうかを判定します。FALSE → 「ABCDEF」、TRUE → 「abcdef」。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.71 WORD_TO_LREALX



符号なし数のLREAL型浮動小数点数への変換は、ARMプラットフォームのTwinCAT 2ではサポートしていません。最上位の有効ビットセットをもつ符号なし数は、暗黙的に負の浮動小数点数に変換される可能性があります。ここで説明したファンクションにより、明示的にWORD型をTwinCAT 2のLREAL型の正の浮動小数点数に変換できます(最上位の有効ビットがセットされ、コンパイラ警告がない場合でも)。変換されたTwinCAT 2プロジェクトを変更せずに、TwinCAT 3でコンパイルできるようにするためにのみ、このファンクションが必要です。

WORD型の符号なし数は、必ず(暗黙的および明示的に)TwinCAT3で正の浮動小数点数に変換されます。このため、このファンクションは無視できます。

ファンクションWORD_TO_LREAL: LREAL

VAR_INPUT

```
VAR_INPUT
    in : WORD;
END_VAR
```

サンプル:

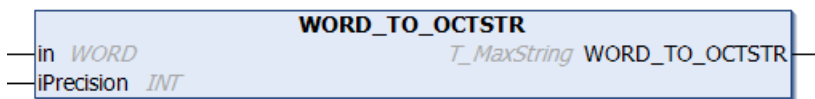
```
PROGRAM MAIN
VAR
    nWord : WORD := 16#FFFF;
    fLreal : LREAL := 0.0;
END_VAR

(* fLreal value | Tc2.x ARM |
   Tc2.x X86 | Tc3.x ARM, X86, X64 *)
(*
_____*)
fLreal := nWord; (* | +65535, Warning 1105* |
+65535 | +65535 *)
fLreal := WORD_TO_LREAL( nWord ); (* | +65535, Warning 1105* |
+65535 | +65535 *)
fLreal := WORD#16#FFFF; (* | +65535, Warning 1105* |
+65535 | +65535 *)
fLreal := 16#FFFF; (* | +65535 |
+65535 | +65535 *)
fLreal := WORD_TO_LREAL( nWord ); (* | +65535 |
+65535 | +65535 *)
fLreal := WORD_TO_LREAL( WORD#16#FFFF ); (* | +65535 |
+65535 | +65535 *)
fLreal := WORD_TO_LREAL( 16#FFFF ); (* | +65535 |
+65535 | +65535 *)
(* -: not supported *)
(* TwinCAT 2.x Warning 1105*:warning: Conversion of unsigned integer to LREAL is not supported. The
value is used as signed instead. *)
```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

4.72 WORD_TO_OCTSTR



このファンクションは、10進数を8進数文字列(8進数)に変換します。

ファンクションWORD_TO_OCTSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in : WORD;
    iPrecision : INT;
END_VAR
```

in: 変換を必要とする10進数。

iPrecision: 表示される数字の最小桁(桁数)。実際の有効桁数が*iPrecision*パラメータ未満の場合、生成される文字列は左からゼロで埋められます。有効桁数が*iPrecision*パラメータより大きい場合、生成される文字列は切り捨てられません *iPrecision*パラメータと*in*パラメータがゼロの場合、生成される文字列は空です。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

5 データ型

5.1 ADSDATATYPEID

ADSデータ型ID。このデータ型は、たとえば、ファンクションブロック `FB_ReadSymInfoByNameEx` [▶ 123] によって使用されます。

```

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

```

値	意味
ADST_VOID	予約
ADST_INT8	符号付き8ビット整数
ADST_UINT8	符号なし8ビット整数
ADST_INT16	符号付き16ビット整数
ADST_UINT16	符号なし16ビット整数
ADST_INT32	符号付き32ビット整数
ADST_UINT32	符号なし32ビット整数
ADST_INT64	符号付き64ビット整数
ADST_UINT64	符号なし64ビット整数
ADST_REAL32	32ビット浮動小数点数
ADST_REAL64	64ビット浮動小数点数
ADST_BIGTYPE	構造化タイプ
ADST_STRING	文字列タイプ
ADST_WSTRING	ワイド文字タイプ
ADST_REAL80	予約
ADST_BIT	ビットタイプ
ADST_MAXTYPES	最大使用可能タイプ

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.2 E_AmsLoggerMode

AMSロガー制御モード。このデータ型は、ファンクションブロックFB_AmsLogger_ [▶_35]によって使用されます。

```
TYPE E_AmsLoggerMode :
(
  AMSLOGGER_RUN      := 1,
  AMSLOGGER_STOP     := 2
);
END_TYPE
```

値	意味
AMSLOGGER_RUN	AMSロガーの起動
AMSLOGGER_STOP	AMSロガーの停止

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.3 E_ArgType

内部引数タイプID。このタイプは、文字列フォーマットファンクション/ファンクションブロックによって使用されます。

```
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_ULARGE,
  ARGTYPE_UHUGE,
  ARGTYPE_LARGE,
  ARGTYPE_HUGE,
  ARGTYPE_LWORD
);
END_TYPE
```

値	対応するPLCデータ型
ARGTYPE_UNKNOWN	タイプは未知、または初期化されていません
ARGTYPE_BYTE	BYTE (8ビット)
ARGTYPE_WORD	WORD (16ビット)
ARGTYPE_DWORD	DWORD (32ビット)
ARGTYPE_REAL	REAL
ARGTYPE_LREAL	LREAL
ARGTYPE_SINT	SINT

値	対応するPLCデータ型
ARGTYPE_INT	INT
ARGTYPE_DINT	DINT
ARGTYPE_USINT	USINT
ARGTYPE_UINT	UINT
ARGTYPE_UDINT	UDINT
ARGTYPE_STRING	文字列タイプ: T_MaxString
ARGTYPE_BOOL	BOOL
ARGTYPE_BIGTYPE	任意のデータ構造体またはバイトバッファ
ARGTYPE_ULARGE	T_ULARGE_INTEGERまたはULINT (符号なし64ビット整数)
ARGTYPE_UHUGE	T_UHUGE_INTEGER (符号なし128ビット整数)
ARGTYPE_LARGE	T_LARGE_INTEGERまたはLINT (符号付き64ビット整数)
ARGTYPE_HUGE	T_HUGE_INTEGER (符号付き128ビット整数)
ARGTYPE_LWORD	LWORD (64ビット)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.4 E_DbgContext

この変数タイプは、プロトコルブロックによって使用できます。デバッグ出力の内容を判定します。

```

TYPE E_DbgContext :
(
  eDbgContext_NONE := 0, (* Not used *)
  eDbgContext_USER := 1, (* Service user *)
  eDbgContext_PROV := 2 (* Service provider *)
);
END_TYPE

```

値	意味
eDbgContext_NONE	パラメータは使用されません
eDbgContext_USER	デバッグ出力はサービスユーザによってトリガされます。
eDbgContext_PROV	デバッグ出力はサービスプロバイダによってトリガされます。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.5 E_DbgDirection

デバッグ出力を設定するためにバッファブロックまたはプロトコルブロックが、この変数タイプを使用できます。

```

TYPE E_DbgDirection :
(
  eDbgDirection_OFF := 0, (* Disabled (no debug ooutput) *)

```

```
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
```

値	意味
eDbgDirection_OFF	デバッグ出力の無効化
eDbgDirection_IN	着信テレグラムの出力の有効化
eDbgDirection_OUT	発信テレグラムの出力の有効化
eDbgDirection_ALL	着信および発信テレグラムの出力の有効化

例:

たとえば、デバッグ出力はそれ自体、ADSLOGSTRファンクションで実現できます。

たとえば、リングバッファでは、デバッグ出力は次のようにして変数によって制御できます。

- ・ 値がeDbgDirection_INまたはeDbgDirection_ALLで、新しい値がバッファに追加される場合、デバッグ出力がトリガされます。
- ・ 値がeDbgDirection_outまたはeDbgDirection_ALLで、値がバッファから削除される場合、デバッグ出力がトリガされます。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

5.6 E_EnumCmdType

列挙型ブロックの制御パラメータ。すべてのパラメータが各列挙型ブロックによって使用できるわけではありません

```
TYPE E_EnumCmdType :
(
  eEnumCmd_First := 0,
  eEnumCmd_Next,
  eEnumCmd_Abort
);
END_TYPE
```

値	意味
eEnumCmd_First	最初のエレメントをリスト
eEnumCmd_Next	次のエレメントをリスト
eEnumCmd_Abort	リスト作成をキャンセル(開いているハンドルを閉じる)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

5.7 E_LDevType

```
TYPE E_LDevType :
(
  eLDT_Unknown : UDINT,
  eLDT_Beckhoff : UDINT,
);
```

```

eLDT_GenericPC      : UDINT,
eLDT_TerminalDongle : UDINT,
eLDT_UsbDongle     : UDINT
);
END_TYPE

```

値	意味
eLDT_Unknown	デバイスタイプが未知。
eLDT_Beckhoff	ベッコフのデバイス (例: IPC、CX、 …)
eLDT_GenericPC	サードパーティ製PC
eLDT_TerminalDongle	ライセンスキーターミナル、例: EL6070
eLDT_UsbDongle	ライセンスキーUSBスティック、例: C9900-L100

5.8 E_LDongleStatus

```

TYPE E_LDongleStatus :
(
  eLDT_Unknown      : UDINT,
  eLDT_OK           : UDINT,
  eLDT_Pending      : UDINT,
  eLDT_Invalid      : UDINT,
  eLDT_NoConnection : UDINT
);
END_TYPE

```

値	意味
eLDT_Unknown	ライセンス dongle のステータスが未知
eLDT_OK	ライセンス dongle の検証に成功しました
eLDT_Pending	ライセンス dongle の検証を実行中
eLDT_Invalid	ライセンス dongle が無効
eLDT_NoConnection	ライセンス dongle の接続がありません

5.9 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

```

値	意味
E_LHR_LicenseOK	ライセンスは有効
E_LHR_LicenseOK_Pending	ライセンス認証デバイスの検証が必要(例: ライセンスキーターミナル)
E_LHR_LicenseOK_Demo	評価用ライセンスが有効
E_LHR_LicenseOK_OEM	OEMライセンスが有効
E_LHR_LicenseNoFound	ライセンスが見つかりません
E_LHR_LicenseExpired	ライセンスの期限が切れています
E_LHR_LicenseExceeded	ライセンスのインスタンスが少なすぎます
E_LHR_LicenseInvalid	ライセンスが無効
E_LHR_LicenseSystemIdInvalid	ライセンス用の不正なシステムID
E_LHR_LicenseNoTimeLimit	ライセンスは時間で制限されません
E_LHR_LicenseTimeInFuture	ライセンスの問題: 将来の時間の問題
E_LHR_LicenseTimePeriodToLong	ライセンス期間が長すぎます
E_LHR_DeviceException	システムスタートアップの例外処理
E_LHR_LicenseDuplicated	ライセンスデータが複数回読み込まれます
E_LHR_SignatureInvalid	無効なシグネチャ
E_LHR_CertificateInvalid	無効な証明書
E_LHR_LicenseOemNotFound	未知のOEM用のOEMライセンス
E_LHR_LicenseRestricted	システムの無効なライセンス
E_LHR_LicenseDemoDenied	評価用ライセンスは許可されません
E_LHR_LicensePlatformLevelInv	ライセンス用の無効なプラットフォームレベル

5.10 E_MIB_IF_Type

管理情報ベースインターフェイスタイプ。

```

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

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.11 E_NumGroupTypes

数値グループ。たとえば、このデータ型はファンクションブロックFB_EnumStringNumbers [▶_48]によって使用されています。

```

TYPE E_NumGroupTypes :
(
  eNumGroup_Float,
  eNumGroup_Unsigned,
  eNumGroup_Signed
);
END_TYPE

```

値	意味
eNumGroup_Float	浮動小数点数
eNumGroup_Unsigned	符号なし数
eNumGroup_Signed	符号付き数

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.12 E_PersistentMode

永続データが書き込まれるモード。このデータ型は、ファンクションブロックFB [WritePersistentData](#) [[▶ 109](#)]によって使用されます。

```

TYPE E_PersistentMode :
(
  SPDM_2PASS      := 0,
  SPDM_VAR_BOOST := 1
);
END_TYPE

```

値	意味
SPDM_2PASS	すべてのデータは、同じサイクルからです。
SPDM_VAR_BOOST	個々の永続変数のデータは、同じサイクルからです。

例:

永続データの書き込み: システム動作 [[▶ 338](#)]も参照してください。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.13 E_RegValueType

レジストリ値のタイプID

```

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
```

値	意味
REG_NONE	値TYPEがありません
REG_SZ	ユニコードのヌルで終端されたSTRING
REG_EXPAND_SZ	ユニコードのヌルで終端されたSTRING (環境変数の参照を使用)
REG_BINARY	フリーフォームのバイナリ
REG_DWORD	32ビット数とREG_DWORD_LITTLE_ENDIAN (REG_DWORDと同じ)
REG_DWORD_BIG_ENDIAN	32ビット数
REG_LINK	シンボリックリンク (ユニコード)
REG_MULTI_SZ	複数のUnicode文字列
REG_RESOURCE_LIST	リソースマップのリソースリスト
REG_FULL_RESOURCE_DESCRIPTOR	ハードウェア記述のリソースリスト
REG_RESOURCE_REQUIREMENTS_LIST	-
REG_QWORD	64ビット数とREG_QWORD_LITTLE_ENDIAN (REG_QWORDと同じ)

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

5.14 E_RouteTransportType

AMSメッセージを送信するのに使用するトランスポート層。現在のところ、TCP/IPのトランスポート層のみがサポートしています。

```
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
```

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

5.15 E_SBCSType

Windows SBCS (シングルバイト文字セット)コードページタイプ。

```
TYPE E_SBCSType :
(
  eSBCS_WesternEuropean := 1,
  eSBCS_CentralEuropean := 2
);
END_TYPE
```

値	意味
eSBCS_WesternEuropean	Windows 1252 (デフォルト)コードページ
eSBCS_CentralEuropean	Windows 1251コードページ

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

5.16 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
);
```

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

5.17 E_TimeZoneID

OSの設定されたタイムゾーンについての追加情報

```
TYPE E_TimeZoneID :
(
  eTimeZoneID_Invalid := -1,
  eTimeZoneID_Unknown := 0,
  eTimeZoneID_Standard := 1,
  eTimeZoneID_Daylight := 2
);
END_TYPE
```

値	意味
eTimeZoneID_Invalid	読み込むことのできなかったタイムゾーン設定
eTimeZoneID_Unknown	タイムゾーン設定を読み込むことができたが、標準/夏時間情報が未知です。
eTimeZoneID_Standard	使用されている現在の標準時間
eTimeZoneID_Daylight	使用されている現在の夏時間

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

5.18 E_TypeFieldParam

文字列フォーマットタイプのフィールド。

```

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

```

値	意味
TYPEFIELD_UNKNOWN	未知、または初期化されていません
TYPEFIELD_B	bまたはB: バイナリ数
TYPEFIELD_O	oまたはO: 8進数
TYPEFIELD_U	uまたはU: 符号なし10進数
TYPEFIELD_C	cまたはC: ASCII文字
TYPEFIELD_F	fまたはF: 浮動小数点数(正規化表現)
TYPEFIELD_D	dまたはD: 符号付き10進数
TYPEFIELD_S	sまたはS: 文字列
TYPEFIELD_XU	X: 16進数(大文字)
TYPEFIELD_XL	x: 16進数(小文字)
TYPEFIELD_EU	E: 浮動小数点数(科学的表記)
TYPEFIELD_EL	e: 浮動小数点数(科学的表記)

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

5.19 FLOAT

LREALエイリアス型

```

TYPE FLOAT :LREAL;
END_TYPE

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

5.20 GUID

システムID.

```

TYPE GUID :
STRUCT
    Data1 : DWORD;
    Data2 : WORD;
    Data3 : WORD;
    Data4 : ARRAY[0..7] OF BYTE;
END_STRUCT
END_TYPE

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.21 OTSTRUCT

稼働時間カウンタの時間の形式

```

TYPE OTSTRUCT :
STRUCT
    wWeek      : WORD;
    wDay       : WORD;
    wHour      : WORD;
    wMinute    : WORD;
    wSecond    : WORD;
    wMilliseconds : WORD;
END_STRUCT
END_TYPE

```

wWeek: 週数: 0 ~ 65535;

wDay: 日数: 0 ~ 7;

wHour: 時間: 0 ~ 23;

wMinute: 分: 0 ~ 59;

wSecond: 秒: 0 ~ 59;

wMilliseconds: ミリ秒: 0~999;

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.22 PROFILERSTRUCT

プロファイラファッシュンブロックのステータス情報。

```

TYPE PROFILERSTRUCT :
STRUCT
    LastExecTime      : DWORD;
    MinExecTime       : DWORD;
    MaxExecTime       : DWORD;
    AverageExecTime   : DWORD;

```

```

    MeasureCycle      : DWORD;
END_STRUCT
END_TYPE

```

LastExecTime: 実行時間の最新の計測値[マイクロ秒]。

MinExecTime: 最小の実行時間[マイクロ秒単位]。

MaxExecTime: 最大の実行時間[マイクロ秒単位]。

AverageExecTime: 最後の10個の計測値の平均実行時間[マイクロ秒単位]。

MeasureCycle: 既に実行された計測数。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

5.23 REMOTEPC

リモートPCコンフィグレーションエン트리。

```

TYPE REMOTEPC :
STRUCT
    NetId : T_AmsNetId;
    Name  : STRING(31);
END_STRUCT
END_TYPE

```

NetId: リモートPCのネットワークアドレス(T_AmsNetID型)。

Name: リモートPCのID。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

5.24 REMOTEPCINFOSTRUCT

複数のリモートPCのコンフィグレーションエントリをもつリスト(REMOTEPC [▶ 291]型)。

```

TYPE REMOTEPCINFOSTRUCT : ARRAY[0..99] OF REMOTEPC;
END_TYPE

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

5.25 ST_AmsRouteEntry

このデータ型は、リモートTwinCAT接続のコンフィグレーションについての情報を格納します。

```

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: リモートTwinCATシステムのシンボル名この名前は自由に選ぶことができます。最大文字列長は定数で制限されます(デフォルト: 31文字)。

sNetID: リモートTwinCATシステムネットワークアドレス(T_AmsNetID型)。

sAddress: それぞれのトランスポート層に関連するシステムアドレス。TCP/IPでトランスポート層を使用する場合、IPアドレスをここで指定します。最大文字列長は定数で制限されます(デフォルト: 79文字)。

eTransport: AMSメッセージを送信するのに使用するトランスポート層(E_RouteTransportType [▶ 287]型)。現在のところ、TCP/IPのトランスポート層のみがサポートしています。

tTimeout: タイムアウト時間(現在、予約されていますが、使用されていません)。

dwFlags: 追加オプション(現在、予約されていますが、使用されていません)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.26 ST_CheckLicense

ライセンス情報をもつ構造体

```

TYPE ST_CheckLicense :
STRUCT
    stLicenseId      : GUID;
    tExpirationTime  : TIMESTRUCT;
    sExpirationTime  : STRING(80);
    eResult          : E_LicenseHResult;
    nCount           : UDINT;
END_STRUCT
END_TYPE

```

名前	説明
stLicenseId	ライセンスID
tExpirationTime	有効期限
sExpirationTime	有効期限
eResult	ライセンスステータス(E_LicenseHResult [▶ 284]を参照)
nCount	このライセンスのインスタンス数(0=無制限)

5.27 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: ターゲットシステムタイプ、例: 'CX1000 CE'、...

strHardwareModel: ハードウェア型式、例: '1001'。

strHardwareSerialNo: ハードウェアシリアル番号、例: '123'。

strHardwareVersion: ハードウェアバージョン、例: '1.7'。

strHardwareDate: ハードウェア製造日、例: '18.8.06'。

strHardwareCPU: ハードウェアCPUアーキテクチャ、例: 'INTELx86'、'ARM'、'UNKNOWN' または '' (空の文字列)。

strImageDevice: ソフトウェアプラットフォーム、例: 'CX1000'、...

strImageVersion: ソフトウェアプラットフォームバージョン、例: '2.15'。

strImageLevel: ソフトウェアプラットフォームレベル、例: 'HMI'。

strImageOsName: OS名、例: 'Windows CE'。

strImageOsVersion: OSバージョン、例: '5.0'。

strTwinCATVersion: TwinCATバージョン、例: TwinCAT 2.10.1307: '2'。

strTwinCATRevision: TwinCATリビジョン、例: TwinCAT 2.10.1307: '10'。

strTwinCATBuild: TwinCATビルド、例: TwinCAT 2.10.1307: '1307'。

strTwinCATLevel: 登録されたTwinCATレベル、例: 'PLC'、'NC-PTP'、'NC-I'、...

strAmsNetId: TwinCAT AMS-NetID、例: '5.0.252.31.1.1'。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

5.28 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: ターゲットシステムタイプ、例: 'CX1000 CE'、...

strHardwareModel: ハードウェア型式、例: '1001'。

strHardwareSerialNo: ハードウェアシリアル番号、例: '123'。

strHardwareVersion: ハードウェアバージョン、例: '1.7'。

strHardwareDate: ハードウェア製造日、例: '18.8.06'。

strHardwareCPU: ハードウェアCPUアーキテクチャ、例: 'INTELx86'、'ARM'、'UNKNOWN' または '' (空の文字列)。

strImageDevice: ソフトウェアプラットフォーム、例: 'CX1000'、...

strImageVersion: ソフトウェアプラットフォームバージョン、例: '2.15'。

strImageLevel: ソフトウェアプラットフォームレベル、例: 'HMI'。

strImageOsName: OS名、例: 'Windows CE'。

strImageOsVersion: OSバージョン、例: '5.0'。

strTwinCATVersion: TwinCATバージョン、例: TwinCAT 2.10.1307: '2'。

strTwinCATRevision: TwinCATリビジョン、例: TwinCAT 2.10.1307: '10'。

strTwinCATBuild: TwinCATビルド、例: TwinCAT 2.10.1307: '1307'。

strTwinCATLevel: 登録されたTwinCATレベル、例: 'PLC'、'NC-PTP'、'NC-I'、...

strAmsNetId: TwinCAT AMS-NetID、例: '5.0.252.31.1.1'。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.29 ST_FileAttributes

ファイルまたはディレクトリアトリビュート

```

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: ファイルまたはディレクトリには、読み込みアクセスアクセスだけが許可されています。ファイルはアプリケーションで読み込みできますが、書き込みをしたり、削除したりできません。ディレクトリの場合は、アプリケーションで削除できません。

bHidden: ファイルまたはディレクトリは非表示で、標準のリスト作成では表示されません。

bSystem: ファイルまたはディレクトリはOSの一部であり、OSによって排他的に使用されます。

bDirectory: このアトリビュートを使用して、ディレクトリを特定できます。

bArchive: ファイルまたはディレクトリはアーカイブに属します。アプリケーションは、バックアップや削除のためにファイルにタグ付けするのにこのアトリビュートを使用します。

bDevice: 予約。

bNormal: ファイルまたはディレクトリは他のアトリビュートセットを持ちません。排他的に使用される場合のみ、このアトリビュートは有効です。

bTemporary: データを保存するためにのみ、ファイルが一時的に使用されます。

bSparseFile: ファイルはスリムダウンファイルです。

bReparsePoint: 「リパースポイント」がファイルまたはディレクトリと関連付けられました。

bCompressed: ファイルまたはディレクトリは圧縮されています。ファイルは圧縮データを格納していません。ディレクトリの場合、圧縮はデフォルトで新しいファイルやサブディレクトリに対してアクティブです。

bOffline: ファイルはいつも利用できるとは限りません。

bNotContentIndexed: ファイルは、インデックスサービスによりインデックス化されていません。

bEncrypted: ファイルまたはディレクトリは暗号化されています。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.30 ST_FileRBufferHead

リングバッファファイルのヘッダステータス。この構造体はファンクションブロックFB `FileRingBuffer` [▶ 50]によって使用されます。この構造体はリングバッファファイルが開くときに読み込まれ、閉じるときにリングバッファファイルに保存されます。この構造体は、データセットが読み込まれる/書き込まれるときに必ず更新されます。

```

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: ステータスフラグ。ビット0 = 1 → ファイルが開いていて、ビット0 = 0 → ファイルは閉じています。Bit 1 = 1 → ファイルは破損しています (正常に閉じられていなかった、または最大バッファサイズが合致していない)。

aces: カウンタにアクセス。このカウンタは、ファイルが開くたびにインクリメントします。

nID: ユーザ定義の32ビット値。

cbBuffer: 最大リングバッファファイルのサイズ。

nCount: 保存されたデータセットの現在数。

cbSize: 保存されたデータバイトの現在数。

ptrFirst: 最も古いデータセットのファイルポインタの位置。

ptrLast: 最新のデータセットのファイルポインタの位置。

rsrv0..rsrv3: 予約。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3. 1. 0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

5. 31 ST_FindFileEntry

このデータ型は、ファイル検索ファンクションブロック [FB_EnumFindFileEntry](#) [▶ 43] と [FB_EnumFindFileList](#) [▶ 44] によって使用されます。

```

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: ファイル名またはディレクトリ名をもつゼロ終端の文字列 (T_MaxString型)。

sAlternateFileName: 従来の8.3フォーマットの代替ファイル名またはディレクトリ名をもつゼロ終端の文字列 (filename.ext)。

fileAttributes: ファイル/ディレクトリアトリビュートをもつ構造体 ([ST_FileAttributes](#) [▶ 294]型)。

fileSize: ファイルのバイトサイズ (64ビット数、[T_ULARGE_INTEGER](#) [▶ 306]型)。

creationTime: ファイルまたはディレクトリが作成された時間を示す構造体変数 ([T_FILETIME](#) [▶ 302]型)。

lastAccessTime: ファイルの場合、構造体は最後にアクセスされた時間を示します (読み込みまたは書き込みまたは書き込み) ([T_FILETIME](#) [▶ 302]型)。ディレクトリの場合、構造体は作成された時間を示します。

lastWriteTime: ファイルの場合、最後に書き込みアクセスした時間を示します ([T_FILETIME](#) [▶ 302]型)。ディレクトリの場合、構造体は作成された時間を示します。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3. 1. 0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

5.32 ST_IPAdapterHwAddr

物理アドレス (MAC)。

```
TYPE ST_IPAdapterHwAddr :
STRUCT
    length : UDINT := 0;
    b      : ARRAY[0..MAX_ADAPTER_ADDRESS_LENGTH] OF BYTE;
END_STRUCT
END_TYPE
```

length: 物理ハードウェアアドレスのバイト長。

b: MACアドレスバイト。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.33 ST_IPAdapterInfo

ネットワークアダプタ情報。

```
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: この変数は、現在のところ、Windows CEでのみ使用されています TRUEの場合、TwinCATはデフォルトアダプタとしてネットワークアダプタを使用します。

sAdapterName: 文字列のアダプタ名。

sDescription: 文字列のアダプタ記述。

physAddr: 物理ハードウェアアドレス (ST_IPAdapterHwAddr [[▶ 297](#)]型)。

dwIndex: 内部アダプタシステムインデックス。

eType: アダプタタイプ (E_MIB_IF_Type [[▶ 285](#)]型)。

sIpAddr: IPアドレス (T_Ipv4Addr型)。

sSubNet: IPネットワークマスク (T_Ipv4Addr型)。

sDefGateway: デフォルトゲートウェイのIPアドレス (T_Ipv4Addr型)。

bDhcpEnabled: Indicates whether このアダプタのためにDHCPが有効になったかどうかを示します。

sDhcpSrv: DHCPサーバのIPアドレス (T_Ipv4Addr型)。

bHaveWins: Windowsインターネットネームサービス (WINS) が使用されているかどうかを示します。

sPrimWinsSrv: プライマリWINSサーバのIPアドレス (T_Ipv4Addr型)。

sSecWinsSrv: セカンダリWINSサーバのIPアドレス (T_Ipv4Addr型)。

tLeaseObt: IPアドレスをDHCPサーバ(UTC)から「借りる」時間を示します。

tLeaseExp: 「延長」がDHCPサーバ(UTC)によってリクエストされる前にDHCPサーバが IPアドレスを「貸し出し」できる期間を示します。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.34 ST_LicenseDongle

接続されたライセンス dongle すべての識別データをもつ構造体

```

TYPE ST_LicenseDongle :
STRUCT
  stAmsAddr      : AMSADDR;
  eDevType       : E_LDevType;
  nFlags         : UDINT;
  eDongleStatus  : E_LDongleStatus;
  nSerialNo      : UDINT;
  nReserved1     : UDINT;
  nReserved2     : UDINT;
END_STRUCT
END_TYPE

```

名前	説明
stAmsAddr	ライセンス dongle のネットワーク ID (AmsNetId とポート)
eDevType	E_LDevType [▶ 283] を参照してください。
nFlags	0: 静的に設定 1: 動的 dongle
eDongleStatus	ライセンス dongle の検証ステータス (E_LDongleStatus [▶ 284] を参照)
nSerialNo	dongle の ID 番号 / ライセンスキーターミナル
nReserved1	将来の用途のために予約
nReserved2	将来の用途のために予約

5.35 ST_TcOnlineLicenseInfoDataEx

```

TYPE ST_TcOnlineLicenseInfoDataEx :
STRUCT
  stLicenseId      : GUID
  stLicenseName    : STRING(80);
  tExpirationTime  : TIMESTRUCT;
  sExpirationTime  : 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

```

名前	説明
stLicenseId	ライセンスID
stLicenseName	ライセンス名
tExpirationTime	有効期限
sExpirationTime	有効期限
nMaxCount	このライセンスのために使用できるインスタンスの最大数(0=無制限)
nUsedCount	使用するライセンスインスタンス数
eResult	ライセンスステータス (E_LicenseHResult [▶_284]を参照)
nVolumeNo	(ボリュームライセンスのみ): ボリュームバンドル番号 0: バンドル番号なし
nOptInfo	(内部)
nRestriction	ライセンスには使用制限があります
bOemLicense	OEMライセンス
bBeckhoffLicense	ベッコフライセンス
bBeckhoffPC	ベッコフのハードウェアのライセンス (IPC、CXなど)
bEtherCATDongle	EtherCATライセンスキーターミナル用のライセンス、例: EL6070
bUSB Dongle	USBライセンス dongle 用のライセンス、例: C9900-L100
bGenDevTypeLic	デバイスタイプ用のライセンス

5.36 ST_TcOnlineLicensesInfoData

ライセンス情報。

```

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: ライセンスIDをGUID [▶_290]値で定義。

sLicenseName: 文字列のライセンス名。

tExpirationTime: ライセンスの使用有効期限を示します (TIMESTRUCT [▶_306]型)。

sExpirationTime: ライセンスの使用有効期限を文字列で示します。

nMaxCount: 該当のライセンスがインスタンスの制限を含むライセンスの場合、許可されたインスタンスの最大長を示します (例: TC3 NC PTP 25軸)。

nUsedCount: 該当のライセンスがインスタンスの制限を含むライセンスの場合、使用するインスタンス数を示します。

eResult: このライセンスのためにエラーコードをHResult列挙値として出力します(ここでは、エラーは負の値で表現されます)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0.4018	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム) v3.3.9.0以降

5.37 ST_TcRouterStatusInfo

TwinCATルータステータス情報。

```

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

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.38 ST_TimeZoneInformation

タイムゾーン情報。標準時間も冬時間として参照されます。また、バイアスパラメータは負の値を含みません。

```

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: 現地時間とUTC時間の現在の差を分単位で定義します。UTC = 現地時間 + バイアス。

standardName: 文字列の標準時間名。

standardDate: この構造体は、夏時間から標準時間への移行についての情報を格納します (TIMESTRUCT [▶ 306]型)。この値を使用しない場合、構造体パラメータ *wMonth* はゼロです。このパラメータを使用する場合、*daylightDate* パラメータも使用する必要があります。*standardDate* を設定できるようにするために、*wYear* パラメータをゼロにセットし、*wDayOfWeek* 用に必要な曜日を選択し、*wDay* (月の週、5が最後の週) 用に1と5の間の値を選択します。

standardBias: 標準時間の間に現地時間を計算するための分単位の時間差。この値は、通常ゼロです。

daylightName: 文字列での夏時間名。

daylightDate: この構造体は、標準時間から夏時間への移行についての情報を格納します (TIMESTRUCT [▶ 306]型)。この値を使用しない場合、構造体パラメータ *wMonth* はゼロです。このパラメータを使用する場合、*standardDate* パラメータも使用する必要があります。*daylightDate* を設定できるようにするために、*wYear* パラメータをゼロにセットし、*wDayOfWeek* 用に必要な曜日を選択し、*wDay* (月の週、5が最後の週) 用に1と5の間の値を選択します。

daylightBias: 夏時間の間に現地時間を計算するための分単位の時間差。

例:

FB_SetTimeZoneInformation [▶ 101]を参照してください。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.39 SYMINFOSTRUCT

TwinCAT PLCシンボル情報。

```

TYPE SYMINFOSTRUCT :
STRUCT
    symEntryLen    : UDINT;
    idxGroup       : UDINT;
    idxOffset      : UDINT;
    byteSize       : UDINT;
    adsDataType    : ADSDATATYPEID;
    symDataType    : T_MaxString;
    symComment     : T_MaxString;
END_STRUCT
END_TYPE

```

symEntryLen: シンボルテーブルのシンボルエントリのバイト単位の実際長。シンボルは、シンボルテーブルに保存されます。個々のエントリの長さは可変で、シンボル名、タイプID、およびコメントの長さによります。

idxGroup: シンボル変数のインデックスグループ。

idxOffset: シンボル変数のインデックスオフセット。

byteSize: シンボル変数の値によって実際に占有されるバイト単位のメモリ容量。たとえば、ブールのPLC変数は1バイトを占有し、一方、20文字の文字列は、実際は21バイトを占有します(20バイトの文字と文字列の終わりをゼロマークするための1バイト)。

adsDataType: ADSデータ型ID (ADSDATATYPEID [▶ 280]型)。このタイプIDはシンボル変数へのADSアクセスで使用されます。すべてのPLC構造体と配列(ユーザ定義のデータ型)はADSデータ型ID、ADST_BIGTYPEを保持していて、このデータ型定数によって識別できません。ユーザ定義データ型を識別できるようにするためには、*symDataType*変数を使用するか、構造体の個々の変数のベースタイプを読み込みします。

symDataType: 文字列でのシンボル変数のデータ型ID。たとえば、これはユーザによって定義されるPLCデータ構造体のタイプ名である可能性があります(T_MaxString型、最大255文字)。

symComment: ユーザがPLC変数定義行に追加したシンボル変数についてのコメント(T_MaxString型、最大255文字)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.40 T_Arg

文字列フォーマットファンクション/ファンクションブロック用の引数タイプ。

```

TYPE T_Arg :
STRUCT
  eType : E_ArgType := ARGTYPE_UNKNOWN;
  cbLen : UDINT     := 0;
  pData : PVOID     := 0;
END_STRUCT
END_TYPE

```

eType: データ型ID (E_ArgType [[▶ 281](#)]型)。

cbLen: メモリで割り当てられるバイト数。

pData: アドレスポインタ。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.41 T_FILETIME

このタイプの変数は、64ビット数です。値は、1601年1月1日 (UTC) からの100ナノ秒間隔での経過時間数に相当します。

```

TYPE T_FILETIME :
STRUCT
  dwLowDateTime : DWORD;
  dwHighDateTime : DWORD;
END_STRUCT
END_TYPE

```

dwLowDateTime: 下位の32ビット。

dwHighDateTime: 上位の32ビット。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.42 T_FIX16

このタイプの変数は、符号付き16ビット固定小数点数を表現します。このデータ型は、しばしば、FPUユニットを持たないシステムによって使用されます(例: 遠隔制御のエリアからのマイクロコントローラまたはデバイス)。たとえば、固定小数点数形式のデータをシリアルインターフェイス経由で送信する必要がある場合、これらのデータは適切なフォーマットに変換しなければいけません。

少数点以下の桁数は、必要な数値範囲と精度ごとに選択されます。小数位が15の場合、 $-1.1-2^{15}$ の範囲で固定小数点数を表示できます。これは、おおよそ浮動小数点数範囲、 $-1.0.999969482421875$ に相当します。

浮動小数点数とは異なり、固定小数点数の精度は数値範囲全体にわたって一定です。残念なことに、表示のための固定小数点数の数値範囲はより小さくなっています。正または負のオーバーフローを生成できる数学的演算では、細心の注意を払ってください。

```

TYPE T_FIX16 :
STRUCT
  value : INT := 0;
  n      : WORD(0..15);
  status : DWORD := 0;
END_STRUCT
END_TYPE

```

value: このメンバ変数は、固定小数点数の現在値を格納します(小数点の前の16ビット、小数点の後の16ビット)。

n: 少数点以下の桁数。許容範囲: 0~15。最上位値のビットは符号ビットのために予約されています。

status: ステータスフラグ(予約済み、現在のところ使用されていません)。

例1:

A/D-Cは、小数位が15である符号付き16ビット固定小数点数として値を返します。これらの計測値はPLCにインポートし、LREALデータ型に変換しなければいけません。

```

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 );

```

例2:

マイクロコントローラのパラメータは、小数点以下8桁の符号付き16ビット固定小数点数です。PLCのLREALパラメータは、このフォーマットに変更される必要があります。

```

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 );

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

5.43 T_HashTableEntry

ハッシュテーブルエントリ/エレメント。

```

TYPE T_HashTableEntry :
STRUCT
  key : DWORD := 0; (* Entry key *)
  value : PVOID := 0; (* Entry value *)
END_STRUCT
END_TYPE

```

key: キー (32ビット符号なし数または32ビットポインタ)。

value: 値 (32/64ビット符号なし数またはポインタの場合があります)。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

5.44 T_HHASHTABLE

ハッシュテーブルハンドル。ハッシュテーブルハンドルは、ファンクションブロック [FB_HashTableCtrl](#) [[▶_71](#)]によって使用されます。

```
TYPE T_HHASHTABLE :
STRUCT
  nCount : UDINT := 0;
  nFree  : UDINT := 0;
END_STRUCT
END_TYPE
```

nCount: 占有されるエレメント数。

nFree: フリーエレメント数。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

5.45 T_HLINKEDLIST

リンクされたリストハンドル。リンクされたリストハンドルは、ファンクションブロック [FB_LinkedListCtrl](#) [[▶_79](#)]によって使用されます。

```
TYPE T_HLINKEDLIST :
STRUCT
  nCount : UDINT := 0;
  nFree  : UDINT := 0;
END_STRUCT
END_TYPE
```

nCount: 占有されるエレメント数。

nFree: フリーエレメント数。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

5.46 T_HUGE_INTEGER

このタイプの変数は、TwinCAT 2符号付き128ビット数を表現します (「legacy」型)。

```
TYPE T_HUGE_INTEGER :
STRUCT
  qwLowPart : T_ULARGE_INTEGER;
```



```

    qwHighPart : T_ULARGE_INTEGER;
END_STRUCT
END_TYPE

```

qwLowPart: 下位の64ビット。

qwHighPart: 上位の64ビット。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.47 T_LARGE_INTEGER

このタイプの変数は、TwinCAT 2符号付き64ビット数を表現します(「legacy」型)。

```

TYPE T_LARGE_INTEGER :
STRUCT
    dwLowPart : DWORD;
    dwHighPart : DWORD;
END_STRUCT
END_TYPE

```

dwLowPart: 下位の32ビット。

dwHighPart: 上位の32ビット。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.48 T_LinkedListEntry

このタイプの変数は、リンクされたリストのノード/エレメントを表します。

```

TYPE T_LinkedListEntry :
STRUCT
    value : PVOID := 0;
END_STRUCT
END_TYPE

```

value: 値(32/64ビット符号なし数またはポインタの場合があります)。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

5.49 T_UHUGE_INTEGER

このタイプの変数は、TwinCAT 2符号なし128ビット数を表現します(「legacy」型)。

```

TYPE T_UHUGE_INTEGER :
STRUCT
    qwLowPart : T_ULARGE_INTEGER;
    qwHighPart : T_ULARGE_INTEGER;
END_STRUCT
END_TYPE

```

qwLowPart: 下位の64ビット。

qwHighPart: 上位の64ビット。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

5.50 T_ULARGE_INTEGER

このタイプの変数は、TwinCAT 2符号なし64ビット数を表現します(「legacy」型)。

```
TYPE T_ULARGE_INTEGER :
STRUCT
    dwLowPart    : DWORD;
    dwHighPart   : DWORD;
END_STRUCT
END_TYPE
```

dwLowPart: 下位の32ビット。

dwHighPart: 上位の32ビット。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

5.51 TIMESTRUCT

システム時間の形式の時間。

```
TYPE TIMESTRUCT
STRUCT
    wYear        : WORD;
    wMonth       : WORD;
    wDayOfWeek   : WORD;
    wDay         : WORD;
    wHour        : WORD;
    wMinute      : WORD;
    wSecond      : WORD;
    wMilliseconds : WORD;
END_STRUCT
END_TYPE
```

wYear: 年: 1970 ~ 2106。

wMonth: 月: 1 ~ 12 (1月 = 1、2月 = 2など)。

wDayOfWeek: 曜日: 0 ~ 6 (日曜日 = 0、月曜日 = 1など)。

wDay: 月の日: 1~31;

wHour: 時: 0 ~ 23;

wMinute: 分: 0 ~ 59;

wSecond: 秒: 0 ~ 59;

wMilliseconds: ミリ秒: 0~999;

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

6 グローバル定数

6.1 ライブラリバージョン

すべてのライブラリに対して、特定のバージョンが付与されています。バージョンは、PLCライブラリリポジトリなどで確認できます。グローバル定数には、ライブラリバージョンに関する情報が含まれています。

Global_Version

```
VAR_GLOBAL CONSTANT
    stLibVersion_Tc2_Uilities : ST_LibVersion;
END_VAR
```

stLibVersion_Tc2_Uilities: Tc2_Uilitiesライブラリのバージョン番号(ST_LibVersion型)。

保有しているバージョンが必要なバージョンかどうかをチェックするためには、ファンクション **F_CmpLibVersion** を使用します (Tc2_Systemライブラリで定義)。



TwinCAT 2で使用されていたライブラリバージョン比較のその他の方法は、すべて使用できなくなっています。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

7 グローバル変数

VAR_GLOBAL

名前	タイプ	値	使用	意味
MAX_AVERAGE_MEASURES	INT	10	プロファイラ [▶_127]	平均計測読み込み数数。可能な値: 2~100
GLOBAL_FORMAT_HASH_PREFIX_TYPE	E_HashPrefixTypes	HASHPREFIX_IEC	FB FormatString [▶_55] 、 F FormatArgToStr [▶_247]	バイナリ、8進数、16進数形式のIECプレフィックス規格
GLOBAL_SBCS_TABLE	E_SBCSType	eSBCS_WesternEuropean	F ToLowerCase [▶_252] 、 F ToUpperCase [▶_252]	Windows SBCS (シングルバイト文字セット)コードページテーブル
GLOBAL_DCF77_PULSE_SPLIT	TIME	T#140ms	DCF77_TIME [▶_26]	パルス長。0 == pulses < 140ms、1 == pulses > 140
GLOBAL_DCF77_SEQUENCE_CHECK	BOOL	FALSE	DCF77_TIME [▶_26]	2つの連続したテレグラムの妥当性チェック: TRUE = 有効、FALSE = 無効。
DEFAULT_CSV_FIELD_SEP	BYTE	16#2C	FB CSVMemBufferWriter [▶_41] 、 FB CSVmemBufferReader [▶_40]	データフィールドの区切り文字。セミコロン(;) = 16#3B → ドイツのフィールド区切り文字、カンマ(,) = 16#2C → 米国のフィールド区切り文字

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

8 サンプル

8.1 例：通信BC/BX<->PC/CX (F_SwapRealEx)

例は、F_SwapRealEx [▶ 251]関クションのアプリケーションを説明します。この例は、2つのコンポーネント、TwinCAT 2.xx BC/BX (バスターミナルコントローラ)アプリケーションとTwinCAT 3.xx PC/CX (x86)アプリケーションを取り上げます。PC/CXアプリケーションはBC/BXのフラグ領域から構造体変数を読み込みし、BC/BXのフラグ領域に構造体変数を書き込みます。構造体変数は、REALエレメントを格納します。これらは、PC/CXで使用され、BC/BXに送信される前に、正しいフォーマットに変換しなければいけません。

下記のリンクから、ソースすべての圧縮ファイル入手できます。

TwinCAT 2.xx - BC/BX (バスターミナルコントローラ)アプリケーション/プロジェクトファイル:
DataExBC.zip (Resources/zip/9007200058074123.zip)

TwinCAT 3.xx - PC/CX (x86, x64, ARM)アプリケーション/アーカイブファイル: DataExPC.zip
(Resources/zip/9007200058077963.zip)

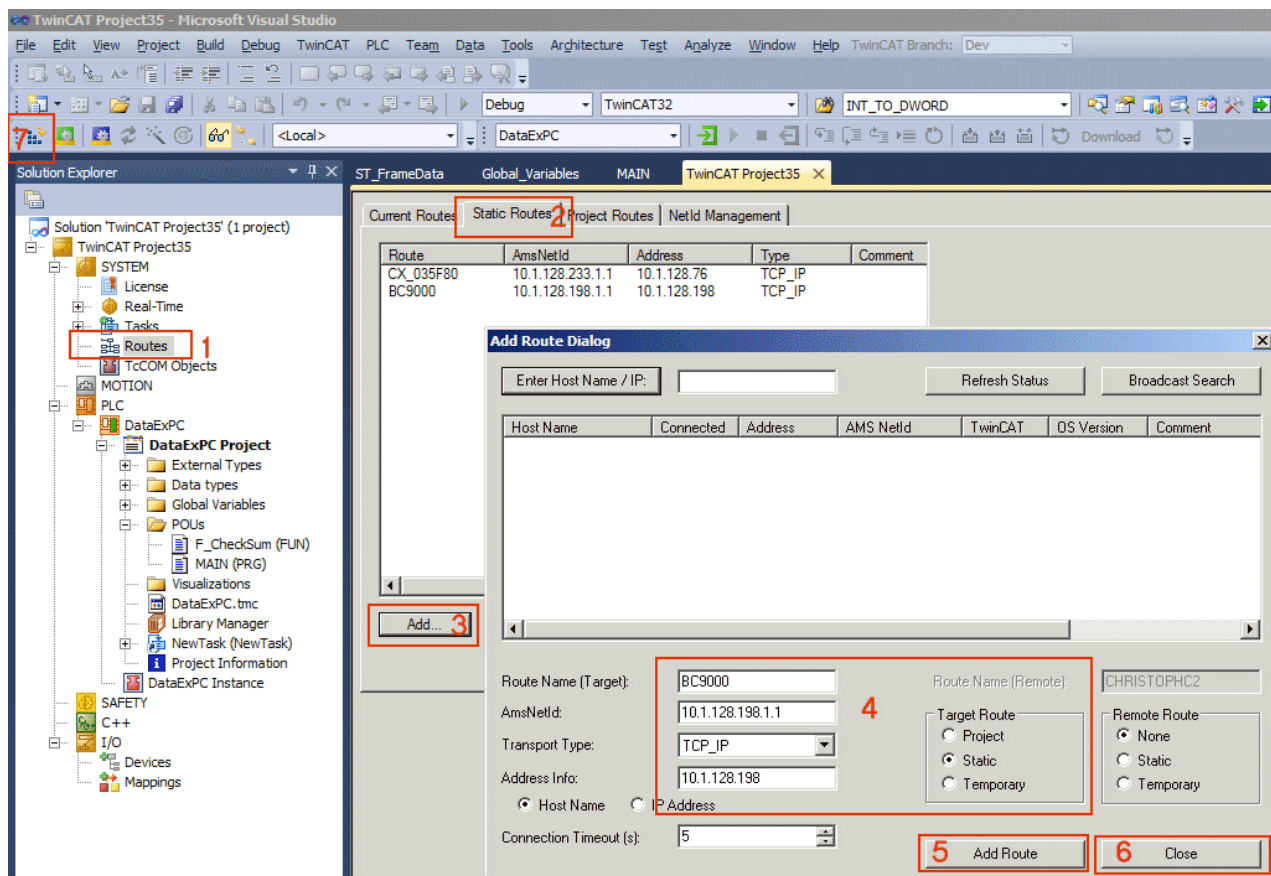
システム要件:

- ・ TwinCAT 2.xx PLC (BC/BXアプリケーションをダウンロードするために必要) + BC/BXハードウェア (例: BC9000)。
- ・ TwinCAT 3.xxエンジニアリングおよびランタイムシステム(PC/CXアプリケーションをダウンロードするために必要)。

プロジェクトのダウンロード

TwinCAT 2.xx PLCを使用して、BC/BXアプリケーションをバスターミナルコントローラ(BC9000など)のランタイムシステムにダウンロードします。ブートプロジェクトを作成し、PLCを起動します。次のステップで、TwinCAT 3.xxを使用して新しいXAEプロジェクトを作成します。PLCノードでマウスを右クリックしてTwinCAT XAEにアーカイブファイルをインポートし、次に既存のアイテムを追加します。

ADSによりBC/BX (バスターミナルコントローラ)にアクセスできるようにするために、TwinCAT AMSルート接続(経路)のリストにデバイスとして入力する必要があります。新しい静的ルートを作成します(図のステップにしたがってください)。BC/BXのAmsNetIDおよびIPアドレスは適切に設定しなければいけません(PLCプログラムコードのAmsNetIDに注意してください)。



重要な注記

BC/BX (バスターミナルコントローラ)およびPC/CX (x86、x64、ARM)のメモリ配置(データ配置)は、異なります。データ交換BC/BX <-> PC/CXの場合、8バイトのメモリ配置の構造体を定義してください。

- ・ TwinCAT 2. xx + PC/CX (x86) プラットフォーム → データ構造体のメモリ配置は1バイトです。
- ・ TwinCAT 2. xx + CX (ARM) プラットフォーム → データ構造体のメモリ配置は4バイト(DWORD)です。
- ・ TwinCAT 2. xx + BC/BX (バスターミナルコントローラ) プラットフォーム → データ構造体のメモリ配置は2バイト(WORD)です。
- ・ TwinCAT 3. xx + PC/CX (x86, x64, ARM) プラットフォーム → データ構造体のメモリ配置は8バイトです。

両方のシステムで使用されている構造体変数の定義。

```
(* 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 (バスターミナルコントローラ)アプリケーション

PC/CXからの各書き込みアクセスアクセスの後で、データ長とチェックサムはチェックされます。読み込みアクセスの新しいランダム値がその後で生成され、単純なチェックサムと関連付けられます。

```
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
END_IF
```

PC/CX (x86, x64, ARM) アプリケーション

bWriteでの立ち上がりで書き込みプロセスプロセスが開始します。REALエレメントは、書き込み操作操作の前にBC/BXフォーマットに変換しなければいけません。データ長とチェックサムを判定し、セットします。bReadの立ち上がりで読み込みプロセスプロセスが開始します。読み込み操作操作が成功した後で、データ長がチェックされ、単純なチェックサムがチェックされます。REALエレメントは、それからPC/CXフォーマットに変換されます。

```
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 Contoroller) *)
    stRxFrame : ST_FrameData; (* Data transported from BC/BX (Bus Terminal Controller) to PC/
CX (x86) *)
    fbWrite : ADSWRITE := ( 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 *)
END_VAR
```



```

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
END_VAR

```

```
(*#####*)
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
```

アプリケーションテスト

PC/CXアプリケーションを開き、TRUEをbWrite変数に書き込みます。次のステップで、TRUEをbRead変数に書き込みます。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

8.2 例：ファイル検索 (FB_EnumFindFileEntry、FB_EnumFindFileList)

次のリンクから、ソースすべての圧縮ファイルを入手できます。EnumFindFile.zip (Resources/zip/9007200058081803.zip)

例：FB_EnumFindFileEntry (ST)

ローカルTwinCAT PCシステムでは、すべてのファイルは次のディレクトリにリストされる必要があります：C:\Windows\System32。ファイル名は、TwinCAT XAEエラーリストにメッセージとして書き込まれます。このプロセスをキャンセルすることは可能です。bEnum変数の立ち上がりで、検出されたファイルのリスト作成を開始します。bAbort変数の立ち上がりでプロセスが中止します。

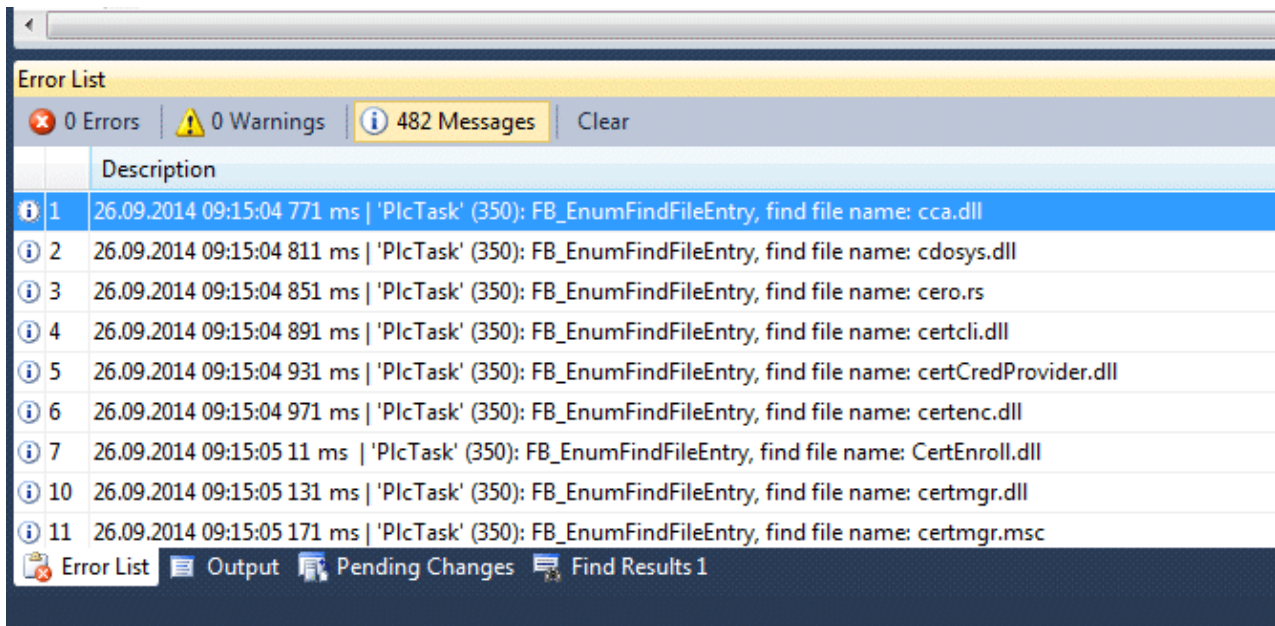
```
PROGRAM P_TestEnumEntry
VAR
    fbEnum: FB_EnumFindFileEntry := ( sNetID := '', tTimeout := T#5s, sPathName := 'C:\Windows\System32\*.*' );
    bEnum : BOOL;
    bAbort: BOOL;
    nState: BYTE;
END_VAR

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 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
END_CASE
```

ログメッセージがTwinCAT XAEエラーリストに書き込まれます。



例: FB_EnumFindFileList (FBD)

*bFirst*変数の立ち上がりでプロセスが有効になります。成功した場合、ファイル名が*fileList*配列変数に入力されます。

```
PROGRAM P_TestEnumList
VAR
    fbEnum      : FB_EnumFindFileList;
    filesList   : ARRAY[1..10] OF ST_FindFileEntry;
    bFirst      : BOOL;
    bNext       : BOOL;
    bAbort      : BOOL;
    bBusy       : BOOL;
    bError      : BOOL;
    nErrID      : UDINT;
    bEOE       : BOOL;
    nFindFiles  : UDINT;
END_VAR
```

オンラインビュー:

P_TestEnumList [Online] × P_TestEnumEntry [Online] MAIN [Online] Source Control Explorer History - Tc2_Uilities.plcproj

TwinCAT_Device.EnumFindFile.P_TestEnumList

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			

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

8.3 例: ファイルリングFIFO (FB_FileRingBuffer)

次のリンクから、ソースすべての圧縮ファイルを入手できます。FileRingBufferExample.zip (Resources/zip/9007200058136843.zip)

以下の例で、ファンクションブロックの簡単なアプリケーションを説明します。*bOpen*の立ち上がりで、既存のリングバッファファイルが開きます。ファイルが存在しない場合、新しいファイルが作成されます。*bClose*の立ち上がりで開いたファイルを閉じます。*bCreate*の立ち上がりで新しいファイルを作成します。*bAdd = TRUE*をセットした場合は新しいデータレコードがリングバッファファイルに書き込まれ、され、*bRemove = TRUE*の場合は最も古いデータレコードが削除されます。

```
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
END_IF
```

```

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

IF bReset THEN
    bReset := FALSE;
    fbFileBuffer.A_Reset();
END_IF

END_IF

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

8.4 例: メモリリングFiFo (FB_MemRingBuffer)

次のリンクから、ソースすべての圧縮ファイルを入手できます。MemRingBufferExample.zip (Resources/zip/9007200058140683.zip)

以下の例で、ファンクションブロックの簡単なアプリケーションを説明します。同じ長さをもつデータセットがバッファリングされます(ただし、これは強制されません)。データセット構造は、次のようになります。

```

TYPE ST_DataSetEntry :
STRUCT
    bFlag : BOOL;
    nValue : BYTE;
    sMsg : STRING(20) := 'Unknown';
END_STRUCT
END_TYPE

```

FB_Data setFifoファンクションブロックのインターフェイス。

サンプルプロジェクトで使用されているアプリケーション固有のファンクションブロックFB_Data setFifoは、内部的にFB_MemRingBufferファンクションブロックを使用します。このブロックは、データセットの追加/削除を単純化します。さらに、新しいファンクションブロックは、書き込みオプションオプションとバッファの現在の占有率を割合で提供します。bOverwrite入力がセットされてバッファがフルの場合、最も古いエントリはバッファから削除され、新しいエントリで上書きされます。

```

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 functi
on block *)
    fbBuffer  : FB_MemRingBuffer;
END_VAR

```

メインプログラム:

*bReset*の立ち上がりで、すべてのバッファエントリを削除します。*bAdd* = TRUEをセットした場合は新しいデータレコードがリングバッファに書き込まれ、*bRemove* =TRUEの場合は最も古いデータレコードが削除されます。*bGet*の立ち上がりでは、最も古いデータセットが読み込まれますが、削除されません。

```

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

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

8.5 例：メモリリングFiFo (FB_MemRingBufferEx)

次のリンクから、ソースすべての圧縮ファイルを入手できます。MemRingBufferExExample.zip (Resources/zip/9007200058145803.zip)

bAddの立ち上がりによって、新しいデータエレメント(pubObj array)がリングバッファに保存されます。bGetの立ち上がりによって、最も古いデータエレメントがgetObj変数にコピーできます。

必要のないデータエレメントは、bReleaseの立ち上がりによってバッファから削除されます。

```
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
```

```

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

```

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

8.6 例: ハッシュテーブル (FB_HashTableCtrl)

次のリンクから、ソースすべての圧縮ファイルを入手できます。HashTableExample.zip (Resources/zip/9007200058162699.zip)

サンプルプロジェクトは、2つのプログラム部分から構成されます。

- ・ P_TABLE_OF_UDINTは、32ビット値をハッシュテーブルに保有するのみの簡単なサンプルプログラムです。
- ・ P_TABLE_OF_STRUCTDATAは、どのようにして他のデータ型 (例: 構造体データ型) がハッシュテーブルで処理できるかを示します。

テーブル要素の最大数はランタイムでは変更できず、MAX_DATA_ELEMENTSによってサンプルプロジェクトでは制限されています。もっと多くの要素が必要な場合、テーブル配列は適切に拡大できます (すなわち、定数の値の増加による)。

```

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

最初のPLCサイクルで、注文番号と注文型式がテーブルに保存されます。注文番号はキーとして機能し、注文型式の配列インデックスは値として機能します。bLookupの立ち上がりで、注文型式は注文番号によって検出できます。

```

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
    END_IF
  END_IF
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

プログラムのこのセクションは、どのようにして構造体データセットが32ビット数の代わりに簡単にテーブルで操作できるかを示します。32ビットエレメント値は、実際のエレメント値への参照ポインタとしてのみ使用されます。参照ポインタは、構造体変数または他のデータ型のインスタンスを指し示すことができます。機能はファンクションブロックでカプセル化されます。ファンクションブロック *FB_SpecialHashTableCtrl* は、*FB_HashTableCtrl* ファンクションブロックの特殊バージョンとみなすことができます。*FB_HashTableCtrl* ブロックも、内部的にFB、専用FBによって使用されます。

DATAELEMENT_TO_STRING ファンクションのみが、ノードの値を視覚的に出力するのを許可するために使用されます。

タイプ *ST_DataElement* の構造体変数を例として使用します。ハイ書き込み：プログラムや *FB_SpecialHashTableCtrl* ファンクションブロックに変更を加えずに、*ST_DataElement* のデータ型宣言にさらにメンバ変数を追加できます。

ST_DataElement のためのタイプ宣言：

```

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

```

32ビットエレメント値が *ST_DataElement* 配列のインスタンスへの参照ポインタになる方法を示します。

テーブルの最大サイズは、定数 *MAX_DATA_ELEMENTS* によって制限されます。したがって、*MAX_DATA_ELEMENTS* 参照ポインタ以外はテーブルに保存できません。内部的に *FB_SpecialHashTableCtrl* ブロックは、*T_HashTableEntry* 配列変数と同じ配列サイズの *ST_DataElement* 配列変数を保持しています。つまり、配列インデックスは両方の配列で同じです

各 *T_HashTableEntry* 配列エレメントは、テーブルで一度のみ使用できます。*FB_HashTableCtrl* ファンクションブロックは、空の/未使用の *T_HashTableEntry* 配列エレメントを検索します。成功した場合、エレメントがテーブルに追加されます。アクション *A_GetIndexAtPosPtr* を使用して、*T_HashTableEntry* 配列のインデックスを判定できます。次のステップで、追加されたばかりの32ビットノード値は、*ST_DataElement* 配列の同じ配列エレメントのアドレスを割り当てられます。サンプルプロジェクトでの第2のアクションのコール *A_Add*。

nodes[index].value := ADR(dataPool[index])

たとえば、割り当ては *FB_SpecialHashTableCtrl* → *A_Add* アクションで実現できます。

```

(* 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, bOk=>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
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

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

8.7 例：リンクされたリスト (FB_LinkedListCtrl)

次のリンクから、ソースすべての圧縮ファイル入手できます。LinkedListExample.zip (Resources/zip/9007200058166539.zip)

サンプルプロジェクトは、2つのプログラム部分から構成されます。

- ・ P_LIST_OF_UDINTは、リンクされたリストで32ビット値を編集するのみの簡単なサンプルプログラムです。
- ・ P_LIST_OF_STRUCTDATAは、どのようにして他のデータ型(例：構造化データ型)がリンクされたリストのコンテキストを管理できるのかを示します。

ノードエレメントの最大数はランタイムでは変更できず、MAX_DATA_ELEMENTSによってサンプルプロジェクトでは制限されています。もっと多くのノードが必要な場合、ノード配列のサイズを適度に大きくしなければいけません(すなわち、定数の値を増加)。

```
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

リンクされたリストのハンドルは、最初のPLCサイクルで初期化されます。リストにアクセスするときに、このハンドルはVAR_IN_OUT変数としてFB_LinkedListCtrlファンクションブロックに渡されます。リンクされたリストは、ファンクションブロックのアクションのコールによって操作されます。これで、ノードエレメントを追加、削除、検索できます。必要なアクションが、関連付けされたブール変数の立ち上がりに対応して実行されます。プログラムを実行する場合、すべてのオペレーションが一度、実行されます。

```
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=>get
```

```
Value );
    putValue := 12345;
    fbList.A_AddTailValue( hList := hList, putValue := putValue, getPosPtr=>getPosPtr, getValue=>get
Value );
    putValue := 67890;
    fbList.A_AddTailValue( hList := hList, putValue := putValue, getPosPtr=>getPosPtr, getValue=>get
Value );
END_IF

IF bAddHeadValue THEN
    bAddHeadValue := FALSE;

    putValue := 33333;
    fbList.A_AddHeadValue( hList := hList, putValue := putValue, getPosPtr=>getPosPtr, getValue=>get
Value );
    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

プログラムのこのセクションは、どのようにして構造体データセットが32ビット数の代わりに簡単にリストで操作できるかを示します。この場合、32ビットノード値は、実際のノード値への参照ポインタとしてのみ使用されます。参照ポインタは、構造体変数または他のデータ型のインスタンスを指し示すことができます。機能はファンクションブロックでカプセル化されます。ファンクションブロック *FB_SpecialLinkedListCtrl* は、*FB_LinkedListCtrl* ファンクションブロックの特殊バージョンとみなすことができます。 *FB_LinkedListCtrl* ブロックも、内部的に専用FBによって使用されます。

DATAELEMENT_TO_STRING ファンクションのみが、ノードの値を視覚的に出力するのを許可するために使用されます。

タイプ *ST_DataElement* の構造体変数を例として使用します。ハイ書き込み：プログラムや *FB_SpecialLinkedListCtrl* ファンクションブロックに変更を加えずに、*ST_DataElement* のデータ型宣言にさらにメンバ変数を追加できます。

ST_DataElement のためのタイプ宣言：

```

(* 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

```

簡単な検索ファンクションが実装されています。特定の**名前**、**番号**または**価格**をもつノードを検索できません。

どのようにして32ビットノード値がST_DataElement配列のインスタンスへの参照ポインタになるのでしょうか?

リストの最大サイズは、定数MAX_DATA_ELEMENTSによって制限されます。したがって、MAX_DATA_ELEMENTS参照ポインタ以外はリストに保存できません。FB_SpecialLinkedListCtrlファンクションブロックは、T_LinkedListEntry配列変数と同じサイズの内部ST_DataElement配列変数を保持しています。つまり、配列インデックスは両方の配列で同じです

各T_LinkedListEntry配列要素は、一度のみリストに挿入のみできます。そのため、FB_LinkedListCtrlファンクションブロックは空の/未使用のT_LinkedListEntry配列要素を検索し、成功した場合にそれをリストに挿入します。使用するT_LinkedListEntryのインデックスは、アクションA_GetIndexAtPosPtrによって判定できます。次のステップで、追加されたばかりの32ビットノード値は、ST_DataElement配列の同じ配列要素のアドレスを割り当てられます。サンプルプロジェクトでのアクションのコールA_SetValueAtPosPtr。

```
nodes[index].value := ADR( dataPool[index] )
```

たとえば、割り当てはFB_SpecialLinkedListCtrl->A_AddHeadValueアクションで実行されます。

```

(* 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 *), getPosPtr=>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 *)

        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
```

```
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

```

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

8.8 例: CSVファイルの書き込み/読み込み

次のリンクから、サンプルプロジェクトのソースすべての圧縮ファイルを入手できます。CSVExample.zip (Resources/zip/9007200058342155.zip)

サンプルプロジェクトで生成されたCSVファイル:

バイナリデータのないデータフィールド: TextModeGen.zip (Resources/zip/9007200058345995.zip)

バイナリデータを格納しているデータフィールド: BinaryModeGen.zip (Resources/zip/18014399312912907.zip) (正しく解釈されるためには、このファイルが特殊なソフトウェアを必要とすることに注意してください)

CSVは、comma-separated values(カンマ区切りの値)の略です。次の記述では、どのようにして補助PLC CSV関クションの助けでCSVファイルの書き込み、読み込みできるかを説明します。基本的にテキストファイルであるCSVファイルは、2つのシステムの間でのデータ交換に使用できる構造化データセットのみを保存できます。このフォーマットにより、テーブルの保存、異なる長さのリストの保存ができます。テーブルの行は、CSVファイルのデータセット(または行)に相当します。テーブルのセルは、CSVファイルのデータフィールドに相当します。

サポートされたCSV形式についての全般情報

- ・ CSV形式のファイルは、拡張子.csvを持っています。
- ・ CRLF文字(CR = キャリッジリターン、LF= ラインフィード)は個々のデータセット(行)を区切るのに使用されます(WindowsOS)。すなわち、各データセットはCRLFが後に続く必要があります。
- ・ CSVファイルは、CRLF文字で終わる必要があります。
- ・ バイナリデータは、単一引用符で囲む必要があります。単一引用符が使用されていない場合、データフィールドは数字と文字のみを格納することができます。
- ・ 特殊文字/制御文字を格納しているデータフィールドは、二重引用符で囲まれます。データフィールドが二重引用符を含む場合、第2の二重引用符が追加されます。

- ・ 特殊文字は、データフィールドを区切るのに使用されます(列)。補助機能によって使用される個々のデータフィールド用の標準区切り文字はセミコロン(;)です。ドイツとヨーロッパではセミコロン(;)がデータフィールドの区切り文字として使用され、米国ではカンマ(,)が使用される傾向にあります。区切り文字は、グローバルPLC変数DEFAULT_CSV_FIELD_SEPを使用してセミコロン(;)からカンマ(,)またはその逆に変更できます。
- ・ 各データセットは、同じ数のデータフィールド(列)を含みます。

n列とn行をもつCSVファイルの基本構成(CRLF文字は通常、不可視ですが、図では文字CRLFで示していません)。

```
"Field1Record1";"Field2Record1"; ... ;"Field(n)Record1"CRLF
"Field1Record2";"Field2Record2"; ... ;"Field(n)Record2"CRLF
...
"Field1Record(n)";"Field2Record(n)"; ... ;"Field(n)Record(n)"CRLF
```

使用可能なファンクションブロックとファンクション

- ・ STRING TO CSVFIELD [▶ 270]、ARG TO CSVFIELD [▶ 229]: データフィールドのPLCデータをCSV形式に変換。
- ・ CSVFIELD TO STRING [▶ 235]、CSVFIELD TO ARG [▶ 234]: CSV形式のデータフィールドをPLCデータに変換。
- ・ FB_CSVMemBufferWriter [▶ 41]: 複数のデータフィールドからバイトバッファにデータセットを生成。
- ・ FB_CSVMemBufferReader [▶ 40]: バイトバッファのデータセットを個々のデータフィールドに分割。

テキストモードまたはバイナリモードでのCSVファイルの書き込み/読み込み

CSVファイルは、ファイルアクセスのためにPLCファンクションブロックの助けでテキストモードまたはバイナリモードで読み込み、書き込みができます。選択したモードに応じて、利点と欠点があります。

99%のケースで、CSVファイルはテキストモードで読み込み//書き込まれます。バイナリモードは、稀なケースでのみ必要とされます。

	テキストモード	バイナリモード
ファイル読み込みアクセスのためのファンクションブロック	FB_FileGets (特殊な機能: 読み込みアクセスの間に、このブロックは、最新のデータセットの最後のCR文字を自動的に削除します。文字は、FB_CSVMemBufferReaderブロックがそのようなデータセットを解釈できるように復元/再挿入しなければいけません)	FB_FileRead
ファイル書き込みアクセス用のファンクションブロック	FB_FilePuts (特殊な機能: 書き込みアクセスの間に、このブロックは、最新のデータセットの最後のCR文字を自動的に追加します。ただし、FB_CSVMemBufferWriterもCR文字を生成します。CSVファイルでの文字の複製を避けるために、書き込みアクセス前にバッファから削除する必要があります)	FB_FileWrite
プログラミングの労力	小さい	大きい

	テキストモード	バイナリモード
データフィールドの特殊文字、印刷不能の制御文字	許可されていません	許可されています
書き込み/読み込み可能な最大データセット長	253文字に制限(データセット + CRLF)、すなわちデータセット長は253文字を超えてはいけません。	最大データセット長は理論的には無制限です。
データセット全体は、書き込みアクセスのためにファンクションブロックで書き込みできます。	あり	あり
データセット全体は、読み込みアクセスのためにファンクションブロックで読み込みできます。	あり。純粋なテキストファイルのデータセットは、CRLFで終わります。そのようなファイルでは、CRLFは行の終わりを示します。FB_FileGetsファンクションブロックは、CRLFまでのデータを読み込みします。	なし
データフィールドのバイナリデータ	許可されていません	許可されています
PLCデータのCSV形式への変換、およびその逆の変換するための補助機能	CSVFIELD TO STRING [▶ 235] STRING TO CSVFIELD [▶ 270]	CSVFIELD TO ARG [▶ 234] ARG TO CSVFIELD [▶ 229]
直接書き込み/読み込みができるサポートされたPLC変数タイプ	T_MaxString (255文字のSTRING)、他のデータ型は最初に文字列に変換し、その後に文字列フォーマットでデータフィールドとして読み込み/書き込みする必要があります。	すべてのデータ型の書き込み/読み込みが可能です。
サンプルコード	P_TextModeRead() P_TextModeWrite()	P_BinaryModeRead() P_BinaryModeWrite()

サンプルプロジェクト

実際は、サンプルプロジェクトには4つのサンプルがあります。テキストモードの書き込み/読み込みアクセス(推奨)の2つのサンプルおよびバイナリモードの書き込み/読み込みアクセスの2つのサンプルです。

P_TextModeRead();

P_TextModeWrite();

P_BinaryModeRead();

P_BinaryModeWrite();

テキストモードでのCSVファイルの読み込みのためのための基本的プログラムシーケンス:

ステップ1: テキストモードでCSVファイルを開きます(FB_FileOpen)。成功した場合、ステップ2に進みます。

ステップ2: ファンクションブロックFB_FileGetsで1行読み込みします。CR文字を追加します(表の注記を参照)。成功した場合はステップ3に進み、そうでない場合はステップ4に進みます(ファイルの終わりに達した、またはエラーが発生した)。

ステップ3: ファンクションブロックFB_CSVMemBufferReaderで読み込んだ行をパーシングします。個々のデータフィールドが読み込まれます。次にステップ2に戻り、次の行を読み込みします。ファイルの終わりに達するか、エラーが発生するまでステップ2と3を繰り返します。

ステップ4: CSVファイルを閉じます(FB_FileClose)。

テキストモードでのCSVファイル書き込みのための基本的プログラムシーケンス。

ステップ1: テキストモードでCSVファイルを開きます (FB_FileOpen)。成功した場合、ステップ2に進みます。

ステップ2: ファンクションブロックFB_CSVMemBufferWriterを使用して、新しいデータセットを生成します。個々のデータフィールドがバッファに書き込まれます。このバッファはより大きい文字列であることがあります。データセットの終わりでCR文字を削除し、ステップ3に進みます。

ステップ3: ファンクションブロックFB_FilePutsで行を書き込みます。すべてのデータセットを書き込みするまでステップ2と3を繰り返します。その後、ステップ4に進みます。

ステップ4: ファイルを閉じます (FB_FileClose)。

要件

開発環境	ターゲットシステム	PLCライブラリ (カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uilities (システム)

8.9 例: ソフトウェアクロック (RTC、RTC_EX、RTC_EX2)

次のリンクから、ソースすべての圧縮ファイルを入手できます。RTCSample.zip (Resources/zip/9007200058349835.zip)

以下の例では、3つのソフトウェアクロックが5秒ごとにローカルWindowsシステム時間と同期します (ローカルWindowsシステム時間はタスクバーに表示されます)。

```
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		

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3. 1. 0	PCまたはCX (x86、x64、ARM)	Tc2_Utilities (システム)

9 付録

9.1 永続データを書き込む場合する場合のシステム動作

書き込みトリガトリガ	永続データアクセスの内部最適化	永続データの一貫性	PLCサイクルタイムの超過
ファンクションブロック <u>WritePersistentData</u> [▶ 137]	なし	すべてのデータは同じPLCサイクルからです。	あり。すべてのデータの書き込みがPLCサイクルタイム以上にかかる場合。
ファンクションブロック <u>FB_WritePersistentData</u> [▶ 109] および <u>SPDM_2PASS</u> [▶ 286]	あり	すべてのデータは同じPLCサイクルからです。	あり。すべてのデータの書き込みがPLCサイクルタイム以上にかかる場合。
ファンクションブロック <u>FB_WritePersistentData</u> [▶ 109]および <u>SPDM_VAR_BOOST</u> [▶ 286]	あり	各変数のデータは同じPLCサイクルからです。	滅多にありません。変数ごとの最大の最大の書き込みがPLCサイクルタイムよりかかる場合。
TwinCATシステムが停止 (TwinCATシステムの停止で、すべての永続データは自動的に書き込まれます)。	あり	すべてのデータは同じPLCサイクルからです。	なし

9.2 フォーマット仕様

このフォーマット仕様は、FB_FormatString [▶ 55]およびFB_FormatString2 [▶ 56]ファンクションブロック、およびF_FormatArgToStr [▶ 247]ファンクションによって使用されます。フォーマット仕様が文字列入力変数によってファンクションブロックに送信される一方で、個々のファンクションパラメータによってファンクションに送信されます。

フォーマット仕様は、以下の各種の必要なオプションのパラメータフィールドを含みます。

- ・ Type [▶ 338]
- ・ Flags [▶ 339]
- ・ Width [▶ 340]
- ・ Precision [▶ 340]

最も簡単なフォーマット仕様は、パーセントの記号とタイプフィールド(例: %s)のみを含みます。パーセントの記号の後に続くすべての文字は、タイプフィールドまでのパラメータフィールドとして評価されます。パーセントの記号の前の文字とタイプフィールドの後の文字は、出力文字列にコピーされます。識別不能のまたは不正な文字の際は、フォーマットはエラーで中止されます。出力文字列でパーセントの記号を出力するためには、2つの連続したパーセントの記号(%%)を使用します。

Type

必要なパラメータフィールド。Type(タイプ)フィールドはASCII文字を格納し、関連する引数が文字列、整数、または浮動小数点数として解釈されるかどうかを指定します。タイプフィールドパラメータによっては大文字、小文字の区別があることに注意してください。

タイプ	引数	出力
b、B	BYTE、WORD、DWORD、REAL ¹ 、SINT ² 、INT ² 、DINT ² 、USINT、UINT、UDINT	バイナリ文字列(例: '101010111000')
o、O	BYTE、WORD、DWORD、REAL ¹ 、SINT ² 、INT ² 、DINT ² 、USINT、UINT、UDINT	8進数文字列
u、U	BYTE、WORD、DWORD、SINT ² 、INT ² 、DINT ² 、USINT、UINT、UDINT	符号なしの10進数文字列
c、C	BYTE、USINT	シングルバイト文字(ASCII)
f、F	REAL ³ 、LREAL	浮動小数点数 文字列の形式は[-]dddd.dddd(dddは10進数)です。 小数点の前の桁数は、浮動小数点数の値によります。小数点の後の桁数は、必要な精度によります。 符号は、負の値に対してのみ表示されます。'#INF'は無限の正の値を返し、'-#INF'は無限の負の値を返します。 送信された変数は、不正な値(NaN、非数)を保持している場合、'#QNAN'または'-#QNAN'が返されます。 フォーマット文字列の長さが生成される文字列の最大許容長を超過する場合、'#OVF'または'-#OVF'が返されます。
d、D	BYTE、WORD、DWORD、SINT、INT、DINT、USINT、UINT、UDINT	10進数文字列 符号は、負の値に対してのみ表示されます。
s、S	STRING	シングルバイト文字列 文字は、最後のゼロまたは精度フィールドパラメータに達するまで出力されます。
X	BYTE、WORD、DWORD、REAL ¹ 、SINT ² 、INT ² 、DINT ² 、USINT、UINT、UDINT	16進数文字列 大文字(「ABCDEF」)はフォーマットに使用されます。
x	BYTE、WORD、DWORD、REAL ¹ 、SINT ² 、INT ² 、DINT ² 、USINT、UINT、UDINT	16進数文字列 小文字(「abcdef」)はフォーマットに使用されます。
E	実装されていません。将来、使用するために予約されています。	指数表現の浮動小数点数
e	実装されていません。将来、使用するために予約されています。	指数表現の浮動小数点数

¹ REAL型変数の内容はバイナリ、8進数値、16進数値または10進数文字列として返されます。

² 符号付きタイプの内容は、バイナリ、8進数値、16進数値または10進数文字列として返されます。

³ REAL型変数はLREAL型に変換され、それからフォーマットされます。

Flags

オプションのパラメータフィールド。1つ以上のフラグをフラグフィールドに、任意の必要な順序で指定できます。フラグフィールドパラメータは、フォーマット値、符号の出力、空白、およびバイナリ/8進/16進プレフィックスのアライメントを指定します。

フラグ	意味	タイプ	標準
-	左揃えフラグ。 フォーマット値は、パラメータ Widthで調整された幅の範囲内での左揃えです。	すべてのタイプと併用可能です。	右揃え。
+	符号フラグ。 符号付き正数のための正符号の強制出力。	e、E、f、F、d、Dとのみ併用されます。そうでない場合、フラグは無視されます。	負符号は負の値にのみ表示されます。
0	ゼロフラグ。 このフラグがWidthパラメータの前に付いていると、生成される文字列は必要な幅に達するまで左からゼロを埋めます。	e、E、f、F、s、Sとのみ併用されます。そうでない場合、フラグは無視されます。 また、左揃えフラグ(-)が追加でセットされる場合、ゼロフラグは無視されます。	ゼロの充填はありません。
Space (' ')	スペースフラグ 正の値は、空白が先行します。	e、E、f、F、d、Dとのみ併用されます。そうでない場合、フラグは無視されます。 また、左に符号フラグ(+)が同時にセットされた場合、空白フラグは無視されます。	空白なし。
#	プレフィックスフラグ。 IECまたはスタンダードCプレフィックスは、フォーマット値の前に配置されます。 IECプレフィックス: 2#、8#、16# (デフォルト) スタンダードCプレフィックス: 0、0x、0X	b、B、o、O、x、Xとのみ併用されます。そうでない場合、フラグは無視されます。 スタンダードCプレフィックスタイプは、プログラム GLOBAL_FORMAT_HASH_PREFIX_TYPE := HASHPREFIX_STDCでグローバル変数 GLOBAL_FORMAT_HASH_PREFIX_TYPE をセットして有効にできます。	プレフィックスはありません。

Width

オプションのパラメータフィールド。Width(幅)フィールドは、出力文字列で最小の文字数の出力を指定する正の10進数を格納します。

アライメントフラグに応じて、必要な幅に達するまでスペースが出力文字列の右、または左に付加されます。幅フィールドパラメータの前にゼロフラグがある場合、生成される文字列は必要な幅に達するまで左からゼロで埋められます。出力文字列は、決して幅フィールドパラメータによって必要な長さに切り捨てられません。

アスタリスク(*)も、幅フィールドパラメータ用に入力できます。その後、必要な値が引数(許容BYTE、WORD、DWORD、USINT、UINT、UDINT)で提供します。幅フィールドパラメータの引数は、値がフォーマットされる引数の後に続きます。

Precision

オプションのパラメータフィールド。Precision(精度)フィールドはドット(.)の後に続き、正の10進数を格納します。ドットの後に値がない場合、デフォルトの精度値が使用されます(表を参照)。

タイプ	意味	標準
b、B、o、O、u、U、x、X、d、D	精度フィールドパラメータは、出力文字列で出力する10進数の文字列(桁数)を指定します。 桁数が文字列に比べて十分でない場合、文字列は左からゼロで埋められます。出力文字列は決して切り捨てられません。	標準: 1
c、C	意味がなく、無視されます。	1文字が出力されます。

タイプ	意味	標準
f、F	精度フィールドパラメータは、少数点以下の桁数を指定します。 引数値は、常にそれぞれの少数点以下の桁数に丸められます。	標準：小数位6
s、S	精度フィールドパラメータは、引数文字列から出力される文字数を指定します。 精度値を超える文字は出力されません。	最後のゼロまでのすべての文字が出力されます。

アスタリスク(*)も、精度フィールドパラメータ用に入力できます。その後、必要な値を引数(許容BYTE、WORD、DWORD、USINT、UINT、UDINT)によって提供されます。精度フィールドパラメータの引数は、値がフォーマットされる引数の後に続きます。

9.3 フォーマットエラーコード

以下のエラーコードは、FB_FormatString [▶ 55]とFB_FormatString2 [▶ 56]ファンクションブロック、およびF_FormatArgToStr [▶ 247]ファンクションによって返されます。複数の引数を使用している場合、引数番号(1~9)がエラーコードに追加されて返されます。引数番号は、フォーマット中に正確にどこでエラーが検出されたかについての情報を提供します。

エラーコード	意味
16#00000000	エラーはありません
16#00000010 + 引数番号(1~9)	無効な位置にあるパーセント記号(%)
16#00000020 + 引数番号(1~9)	無効な位置にあるアスタリスクパラメータ
16#00000040 + 引数番号(1~9)	無効な幅フィールド値
16#00000080 + 引数番号(1~9)	無効な精度フィールド値
16#00000100 + 引数番号(1~9)	フラグの1つが無効な位置にあります
16#00000200 + 引数番号(1~9)	幅フィールド値または精度フィールド値が無効な位置にあります。
16#00000400 + 引数番号(1~9)	精度フィールドのドット「.」符号が無効な位置にあります
16#00000800 + 引数番号(1~9)	無効な(サポートされていない)タイプフィールド値
16#00001000 + 引数番号(1~9)	タイプフィールドと引数パラメータが違います
16#00002000 + 引数番号(1~9)	無効なフォーマット文字列パラメータ
16#00004000 + 引数番号(1~9)	フォーマット文字列の引数が多すぎます
16#00008000 + 引数番号(1~9)	宛先文字列バッファオーバーフロー(フォーマット文字列が長すぎます)
16#00010000 + 引数番号(1~9)	無効なポインタ入力

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

9.4 スコープサーバエラーコード

以下のエラーコードが、FB_ScopeServerControl [▶ 98]ファンクションブロックによって返されます。

```
TYPE E_UTILITIES_ERRORCODES :
(
  eUtilError_NoError           := 0,
  eUtilError_ScopeServerNotAvailable := 16#8001,
```

```
eUtilError_ScopeServerStateChange := 16#8002
);
END_TYPE
```

エラーコード	Enum	意味
0x0000	eUtilError_NoError	エラーはありません
0x8001	eUtilError_ScopeServerNotAvailable	TwinCATスコープサーバは利用できません。インストールされていない可能性があります。
0x8002	eUtilError_ScopeServerStateChange	リクエストされたステータスの変更は許可されていません。有効なステータス変更のリストについては、 B_ScopeServerControl 状態図 [▶ 98]を参照してください。

要件

開発環境	ターゲットシステム	PLCライブラリ(カテゴリグループ)
TwinCAT v3.1.0	PCまたはCX (x86、x64、ARM)	Tc2_Uutilities (システム)

9.5 ADSリターンコード

エラーコードのグループ分け: [0x000 \[▶ 342\]](#)...、[0x500 \[▶ 343\]](#)...、[0x700 \[▶ 343\]](#)...、[0x1000 \[▶ 345\]](#)...

グローバルエラーコード

Hex	Dec	HRESULT	名前	説明
0x0	0	0x9811 0000	ERR_NOERROR	エラーはありません。
0x1	1	0x9811 0001	ERR_INTERNAL	内部エラーです。
0x2	2	0x9811 0002	ERR_NORTIME	リアルタイムではありません。
0x3	3	0x9811 0003	ERR_ALLOCLOCKEDMEM	ロック済みの割り当て - メモリエラーです。
0x4	4	0x9811 0004	ERR_INSERTMAILBOX	メールボックスの空き容量なし - ADSメッセージを送信できませんでした。サイクル当たりのADSメッセージ数を減らすと改善されます。
0x5	5	0x9811 0005	ERR_WRONGRECEIVEHMSG	間違ったHMSG。
0x6	6	0x9811 0006	ERR_TARGETPORTNOTFOUND	ターゲットポートが見つかりません - ADSサーバが開始されていないか、ADSサーバに到達できません。
0x7	7	0x9811 0007	ERR_TARGETMACHINENOTFOUND	ターゲットコンピュータが見つかりません - AMSルートが見つかりませんでした。
0x8	8	0x9811 0008	ERR_UNKNOWNCMDID	不明なコマンドIDです。
0x9	9	0x9811 0009	ERR_BADTASKID	タスクIDが無効です。
0xA	10	0x9811 000A	ERR_NOIO	IOがありません。
0xB	11	0x9811 000B	ERR_UNKNOWNAMSCMD	不明なAMSコマンドです。
0xC	12	0x9811 000C	ERR_WIN32ERROR	Win32エラー。
0xD	13	0x9811 000D	ERR_PORTNOTCONNECTED	ポートが接続されていません。
0xE	14	0x9811 000E	ERR_INVALIDAMSLength	AMS長が無効です。
0xF	15	0x9811 000F	ERR_INVALIDAMSNETID	AMS Net IDが無効です。
0x10	16	0x9811 0010	ERR_LOWINSTLEVEL	インストールレベルが低すぎます - TwinCAT 2ライセンスエラーです。
0x11	17	0x9811 0011	ERR_NODEBUGINTAVAILABLE	デバッグが利用できません。
0x12	18	0x9811 0012	ERR_PORTDISABLED	ポートが無効です - TwinCATシステムサービスが開始されていません。
0x13	19	0x9811 0013	ERR_PORTALREADYCONNECTED	ポートが接続済みです。
0x14	20	0x9811 0014	ERR_AMSSync_W32ERROR	AMS Sync Win32エラー。
0x15	21	0x9811 0015	ERR_AMSSync_TIMEOUT	AMS Syncタイムアウトです。
0x16	22	0x9811 0016	ERR_AMSSync_AMSERROR	AMS同期エラーです。

Hex	Dec	HRESULT	名前	説明
0x17	23	0x9811 0017	ERR_AMSSYNC_NOINDEXINMAP	使用可能なAMS Sync用のインデックスマップがありません。
0x18	24	0x9811 0018	ERR_INVALIDAMSPORT	AMSポートが無効です。
0x19	25	0x9811 0019	ERR_NOMEMORY	メモリがありません。
0x1A	26	0x9811 001A	ERR_TCPSEND	TCP送信エラーです。
0x1B	27	0x9811 001B	ERR_HOSTUNREACHABLE	ホストに到達できません。
0x1C	28	0x9811 001C	ERR_INVALIDAMSFAGMENT	AMSフラグメントが無効です。
0x1D	29	0x9811 001D	ERR_TLSEND	TLS送信エラー - セキュアなADS接続に失敗しました。
0x1E	30	0x9811 001E	ERR_ACCESSDENIED	アクセス拒否 - セキュアなADSアクセスが拒否されました。

ルータのエラーコード

Hex	Dec	HRESULT	名前	説明
0x500	1280	0x9811 0500	ROUTERERR_NOLOCKEDMEMORY	ロックされているメモリを割り当てられません。
0x501	1281	0x9811 0501	ROUTERERR_RESIZEMEMORY	ルータメモリのサイズを変更できませんでした。
0x502	1282	0x9811 0502	ROUTERERR_MAILBOXFULL	メールボックスが可能なメッセージの最大数に達しています。
0x503	1283	0x9811 0503	ROUTERERR_DEBUGBOXFULL	デバッグメールボックスが可能なメッセージの最大数に達しています。
0x504	1284	0x9811 0504	ROUTERERR_UNKNOWNPORTTYPE	ポートタイプが不明です。
0x505	1285	0x9811 0505	ROUTERERR_NOTINITIALIZED	ルータが初期化されていません。
0x506	1286	0x9811 0506	ROUTERERR_PORTALREADYINUSE	ポート番号が既に割り当てられています。
0x507	1287	0x9811 0507	ROUTERERR_NOTREGISTERED	ポートが登録されていません。
0x508	1288	0x9811 0508	ROUTERERR_NOMOREQUEUES	ポートの最大数に達しています。
0x509	1289	0x9811 0509	ROUTERERR_INVALIDPORT	ポートが無効です。
0x50A	1290	0x9811 050A	ROUTERERR_NOTACTIVATED	ルータがアクティブではありません。
0x50B	1291	0x9811 050B	ROUTERERR_FRAGMENTBOXFULL	メールボックスが分割されたメッセージの最大数に達しています。
0x50C	1292	0x9811 050C	ROUTERERR_FRAGMENTTIMEOUT	フラグメントタイムアウトが発生しました。
0x50D	1293	0x9811 050D	ROUTERERR_TOBEREMOVED	ポートが削除されました。

一般的なADSエラーコード

Hex	Dec	HRESULT	名前	説明
0x700	1792	0x9811 0700	ADSERR_DEVICE_ERROR	一般的なデバイスエラーです。
0x701	1793	0x9811 0701	ADSERR_DEVICE_SRVNOTSUPP	サービスがサーバでサポートされていません。
0x702	1794	0x9811 0702	ADSERR_DEVICE_INVALIDGRP	インデックスグループが無効です。
0x703	1795	0x9811 0703	ADSERR_DEVICE_INVALIDOFFSET	インデックスオフセットが無効です。
0x704	1796	0x9811 0704	ADSERR_DEVICE_INVALIDACCESS	読み込み/書き込みが許可されていません。
0x705	1797	0x9811 0705	ADSERR_DEVICE_INVALIDSIZE	パラメータのサイズが正しくありません。
0x706	1798	0x9811 0706	ADSERR_DEVICE_INVALIDDATA	データ値が無効です。
0x707	1799	0x9811 0707	ADSERR_DEVICE_NOTREADY	デバイスの操作準備が完了していません。
0x708	1800	0x9811 0708	ADSERR_DEVICE_BUSY	デバイスがビジーです。
0x709	1801	0x9811 0709	ADSERR_DEVICE_INVALIDCONTEXT	オペレーティングシステムコンテキストが無効です。このエラーは、異なるタスクでADSファンクションブロックを使用すると発生することがあります。PLCでのマルチタスク同期によってこのエラーを解決できる場合があります。
0x70A	1802	0x9811 070A	ADSERR_DEVICE_NOMEMORY	メモリが不足しています。
0x70B	1803	0x9811 070B	ADSERR_DEVICE_INVALIDPARM	パラメータ値が無効です。
0x70C	1804	0x9811 070C	ADSERR_DEVICE_NOTFOUND	見つかりません(ファイル...)
0x70D	1805	0x9811 070D	ADSERR_DEVICE_SYNTAX	ファイルまたはコマンドの構文エラーです。
0x70E	1806	0x9811 070E	ADSERR_DEVICE_INCOMPATIBLE	オブジェクトが一致しません。
0x70F	1807	0x9811 070F	ADSERR_DEVICE_EXISTS	オブジェクトは既に存在しています。
0x710	1808	0x9811 0710	ADSERR_DEVICE_SYMBOLNOTFOUND	シンボルが見つかりません。

Hex	Dec	HRESULT	名前	説明
0x711	1809	0x9811 0711	ADSERR_DEVICE_SYMBOLVERSIONINVALID	シンボルのバージョンが無効です。これは、オンラインでの変更により発生する可能性があります。ハンドルを新規作成してください。
0x712	1810	0x9811 0712	ADSERR_DEVICE_INVALIDSTATE	デバイス(サーバ)が無効な状態です。
0x713	1811	0x9811 0713	ADSERR_DEVICE_TRANSMODENOTSUPP	AdsTransModeはサポートされていません。
0x714	1812	0x9811 0714	ADSERR_DEVICE_NOTIFYHANDINVALID	通知ハンドルが無効です。
0x715	1813	0x9811 0715	ADSERR_DEVICE_CLIENTUNKNOWN	通知クライアントが登録されていません。
0x716	1814	0x9811 0716	ADSERR_DEVICE_NOMOREHDL	使用可能な通知ハンドルがこれ以上ありません。
0x717	1815	0x9811 0717	ADSERR_DEVICE_INVALIDWATCHSIZE	通知サイズが大きすぎます。
0x718	1816	0x9811 0718	ADSERR_DEVICE_NOTINIT	デバイスが初期化されていません。
0x719	1817	0x9811 0719	ADSERR_DEVICE_TIMEOUT	デバイスがタイムアウトしています。
0x71A	1818	0x9811 071A	ADSERR_DEVICE_NOINTERFACE	インターフェイス確認に失敗しました。
0x71B	1819	0x9811 071B	ADSERR_DEVICE_INVALIDINTERFACE	間違ったインターフェイスがリクエストされました。
0x71C	1820	0x9811 071C	ADSERR_DEVICE_INVALIDCLSID	クラスIDが無効です。
0x71D	1821	0x9811 071D	ADSERR_DEVICE_INVALIDOBJID	オブジェクトIDが無効です。
0x71E	1822	0x9811 071E	ADSERR_DEVICE_PENDING	リクエストが保留されています。
0x71F	1823	0x9811 071F	ADSERR_DEVICE_ABORTED	リクエストが中止されています。
0x720	1824	0x9811 0720	ADSERR_DEVICE_WARNING	信号警告です。
0x721	1825	0x9811 0721	ADSERR_DEVICE_INVALIDARRAYIDX	配列インデックスが無効です。
0x722	1826	0x9811 0722	ADSERR_DEVICE_SYMBOLNOTACTIVE	シンボルが有効ではありません。
0x723	1827	0x9811 0723	ADSERR_DEVICE_ACCESSDENIED	アクセスが拒否されました。
0x724	1828	0x9811 0724	ADSERR_DEVICE_LICENSENOTFOUND	ライセンスが見つかりません。
0x725	1829	0x9811 0725	ADSERR_DEVICE_LICENSEEXPIRED	ライセンスの期限が切れています。
0x726	1830	0x9811 0726	ADSERR_DEVICE_LICENSEEXCEEDED	ライセンスを超えています。
0x727	1831	0x9811 0727	ADSERR_DEVICE_LICENSEINVALID	ライセンスが無効です。
0x728	1832	0x9811 0728	ADSERR_DEVICE_LICENSESYSTEMID	ライセンスの問題: システムIDが無効です。
0x729	1833	0x9811 0729	ADSERR_DEVICE_LICENSENOTIMELIMIT	ライセンスの期限がありません。
0x72A	1834	0x9811 072A	ADSERR_DEVICE_LICENSEFUTUREISSUE	ライセンスの問題: まだ時刻が経過していません。
0x72B	1835	0x9811 072B	ADSERR_DEVICE_LICENSETIMETOLONG	ライセンス期間が長すぎます。
0x72C	1836	0x9811 072C	ADSERR_DEVICE_EXCEPTION	システムスタートアップ時の例外です。
0x72D	1837	0x9811 072D	ADSERR_DEVICE_LICENSEDUPLICATED	ライセンスファイルを2回読み込みしました。
0x72E	1838	0x9811 072E	ADSERR_DEVICE_SIGNATUREINVALID	シグネチャが無効です。
0x72F	1839	0x9811 072F	ADSERR_DEVICE_CERTIFICATEINVALID	証明書が無効です。
0x730	1840	0x9811 0730	ADSERR_DEVICE_LICENSEOEMNOTFOUND	OEMからの不明な公開キーです。
0x731	1841	0x9811 0731	ADSERR_DEVICE_LICENSERESTRICTED	このシステムIDに対するライセンスが無効です。
0x732	1842	0x9811 0732	ADSERR_DEVICE_LICENSEDEMOTDENIED	デモライセンスは禁止されています。
0x733	1843	0x9811 0733	ADSERR_DEVICE_INVALIDFNCD	ファンクションIDが無効です。
0x734	1844	0x9811 0734	ADSERR_DEVICE_OUTOFRANGE	有効範囲外です。
0x735	1845	0x9811 0735	ADSERR_DEVICE_INVALIDALIGNMENT	アライメントが無効です。
0x736	1846	0x9811 0736	ADSERR_DEVICE_LICENSEPLATFORM	プラットフォームレベルが無効です。
0x737	1847	0x9811 0737	ADSERR_DEVICE_FORWARD_PL	コンテキスト - パッシブレベルに送信。
0x738	1848	0x9811 0738	ADSERR_DEVICE_FORWARD_DL	コンテキスト - ディスパッチレベルに送信。
0x739	1849	0x9811 0739	ADSERR_DEVICE_FORWARD_RT	コンテキスト - リアルタイムに送信。
0x740	1856	0x9811 0740	ADSERR_CLIENT_ERROR	クライアントエラーです。
0x741	1857	0x9811 0741	ADSERR_CLIENT_INVALIDPDM	サービスに無効なパラメータが含まれています。
0x742	1858	0x9811 0742	ADSERR_CLIENT_LISTEMPTY	ポーリングリストが空です。
0x743	1859	0x9811 0743	ADSERR_CLIENT_VARUSED	var接続が既に使用されています。
0x744	1860	0x9811 0744	ADSERR_CLIENT_DUPLINVOKEID	呼び出されたIDは既に使用されています。
0x745	1861	0x9811 0745	ADSERR_CLIENT_SYNCTIMEOUT	タイムアウトが発生しました - 指定されたADSタイムアウトにリモートターミナルがレスポンスしません。リモートターミナルのルート設定が間違っている可能性があります。
0x746	1862	0x9811 0746	ADSERR_CLIENT_W32ERROR	Win32補助システムのエラーです。
0x747	1863	0x9811 0747	ADSERR_CLIENT_TIMEOUTINVALID	クライアントタイムアウト値が無効です。
0x748	1864	0x9811 0748	ADSERR_CLIENT_PORTNOTOPEN	ポートが開いていません。
0x749	1865	0x9811 0749	ADSERR_CLIENT_NOAMSADDR	AMSアドレスがありません。
0x750	1872	0x9811 0750	ADSERR_CLIENT_SYNCINTERNAL	Ads syncの内部エラーです。

Hex	Dec	HRESULT	名前	説明
0x751	1873	0x9811 0751	ADSERR_CLIENT_ADDHASH	ハッシュテーブルがオーバーフローしました。
0x752	1874	0x9811 0752	ADSERR_CLIENT_REMOVEHASH	テーブルにキーが見つかりません。
0x753	1875	0x9811 0753	ADSERR_CLIENT_NOMORESYM	キャッシュにシンボルがありません。
0x754	1876	0x9811 0754	ADSERR_CLIENT_SYNCRESINVALID	無効なレスポンスを受信しました。
0x755	1877	0x9811 0755	ADSERR_CLIENT_SYNCPORTLOCKED	Syncポートがロックされています。

RTIMEエラーコード

Hex	Dec	HRESULT	名前	説明
0x1000	4096	0x9811 1000	RTERR_INTERNAL	リアルタイムシステムの内部エラーです。
0x1001	4097	0x9811 1001	RTERR_BADTIMERPERIODS	タイマ値が無効です。
0x1002	4098	0x9811 1002	RTERR_INVALIDTASKPTR	タスクポインタに無効な値0（ゼロ）が設定されています。
0x1003	4099	0x9811 1003	RTERR_INVALIDSTACKPTR	スタックポインタに無効な値0（ゼロ）が設定されています。
0x1004	4100	0x9811 1004	RTERR_PRIOEXISTS	リクエストタスクの優先順位は既に割り当て済みです。
0x1005	4101	0x9811 1005	RTERR_NOMORETCB	使用可能なTCB(タスク制御ブロック)はこれ以上ありません。TCBの最大数は64です。
0x1006	4102	0x9811 1006	RTERR_NOMORESEMAS	使用可能な排他処理はこれ以上ありません。排他処理の最大数は64です。
0x1007	4103	0x9811 1007	RTERR_NOMOREQUEUES	キューに使用可能な空きスペースがありません。キューの位置の最大数は64です。
0x100D	4109	0x9811 100D	RTERR_EXTIRQALREADYDEF	外部同期割り込みが既に適用されています。
0x100E	4110	0x9811 100E	RTERR_EXTIRQNOTDEF	外部同期割り込みは適用されていません。
0x100F	4111	0x9811 100F	RTERR_EXTIRQINSTALLFAILED	外部同期割り込みの適用に失敗しました。
0x1010	4112	0x9811 1010	RTERR_IRQNOTLESSOREQUAL	間違ったコンテキストでのサービス機能のコール
0x1017	4119	0x9811 1017	RTERR_VMXNOTSUPPORTED	Intel VT-x拡張はサポートされていません。
0x1018	4120	0x9811 1018	RTERR_VMXDISABLED	Intel VT-x拡張はBIOSでは有効ではありません。
0x1019	4121	0x9811 1019	RTERR_VMXCONTROLSMISSING	Intel VT-x拡張でファンクションが見つかりません。
0x101A	4122	0x9811 101A	RTERR_VMXENABLEFAILS	Intel VT-xの有効化に失敗しました。

TCP Winsockのエラーコード

Hex	Dec	名前	説明
0x274C	10060	WSAETIMEDOUT	接続タイムアウトが発生しました - 接続確立中のエラーです。一定時間が経過してもリモートターミナルがレスポンスしなかった、または接続されたホストがレスポンスしなかったために確立された接続を維持できなかったことが原因です。
0x274D	10061	WSAECONNREFUSED	接続が拒否されました - ターゲットコンピュータが明示的に接続を拒否したために、接続を確立できませんでした。通常、このエラーは外部ホスト上で無効なサービス(サーバアプリケーションが実行されていないサービス)への接続を試行すると発生します。
0x2751	10065	WSAEHOSTUNREACH	ホストへの経路がありません - ソケット操作が使用できないホストを参照しています。

その他のWinsockエラーコード: Win32エラーコード [▶ 345]

9.6 Win32エラーコード

次の表にWin32エラーコードのリストを示します。

0 [▶ 345], 100 [▶ 347], 200 [▶ 350], 1001 [▶ 351], 1100 [▶ 353], 1200 [▶ 355], 1400 [▶ 360], 1600 [▶ 362], 1800 [▶ 366], 2000 [▶ 368], 3000 [▶ 369], 5000 [▶ 371], 6000 [▶ 374], 8000 [▶ 376], 8500 [▶ 385], 9001 [▶ 389], 10004 [▶ 391], 12000 [▶ 394],

エラー			説明
10進数	16進数	名前	
0	0x00000000	ERROR_SUCCESS	操作が正常に完了しました。
1	0x00000001	ERROR_INVALID_FUNCTION	ファンクションが正しくありません。

エラー			説明
10進数	16進数	名前	
2	0x00000002	ERROR_FILE_NOT_FOUND	システムは指定されたファイルを見つけることができません。
3	0x00000003	ERROR_PATH_NOT_FOUND	システムは指定されたパスを見つけることができません。
4	0x00000004	ERROR_TOO_MANY_OPEN_FILES	システムはファイルを開くことができません。
5	0x00000005	ERROR_ACCESS_DENIED	アクセスが拒否されました。
6	0x00000006	ERROR_INVALID_HANDLE	ハンドルが無効です。
7	0x00000007	ERROR_ARENA_TRASHED	ストレージ制御ブロックが壊れています。
8	0x00000008	ERROR_NOT_ENOUGH_MEMORY	このコマンドを処理するために十分なストレージを使用できません。
9	0x00000009	ERROR_INVALID_BLOCK	ストレージ制御ブロックのアドレスが無効です。
10	0x0000000A	ERROR_BAD_ENVIRONMENT	環境が正しくありません。
11	0x0000000B	ERROR_BAD_FORMAT	正しくない形式でプログラムをロードしようとしました。
12	0x0000000C	ERROR_INVALID_ACCESS	アクセスコードが無効です。
13	0x0000000D	ERROR_INVALID_DATA	データが無効です。
14	0x0000000E	ERROR_OUTOFMEMORY	この操作を完了するために十分なストレージを使用できません。
15	0x0000000F	ERROR_INVALID_DRIVE	システムは指定されたドライブを見つけることができません。
16	0x00000010	ERROR_CURRENT_DIRECTORY	ディレクトリを削除できません。
17	0x00000011	ERROR_NOT_SAME_DEVICE	システムはファイルを別のディスクドライブに移動できません。
18	0x00000012	ERROR_NO_MORE_FILES	ファイルはこれ以上ありません。
19	0x00000013	ERROR_WRITE_PROTECT	メディアが書き込み保護されています。
20	0x00000014	ERROR_BAD_UNIT	システムは指定されたデバイスを見つけることができません。
21	0x00000015	ERROR_NOT_READY	デバイスがレディ状態ではありません。
22	0x00000016	ERROR_BAD_COMMAND	デバイスがコマンドを認識できません。
23	0x00000017	ERROR_CRC	データエラー(巡回冗長検査)。
24	0x00000018	ERROR_BAD_LENGTH	プログラムがコマンドを発行しましたが、コマンドの長さが正しくありません。
25	0x00000019	ERROR_SEEK	ドライブがディスク上の特定の領域またはトラックを見つけることができません。
26	0x0000001A	ERROR_NOT_DOS_DISK	指定されたディスクにアクセスできません。
27	0x0000001B	ERROR_SECTOR_NOT_FOUND	ドライブは要求されたセクタを見つけることができません。
28	0x0000001C	ERROR_OUT_OF_PAPER	プリンタに用紙がありません。
29	0x0000001D	ERROR_WRITE_FAULT	システムは指定されたデバイスに書き込むことができません。
30	0x0000001E	ERROR_READ_FAULT	システムは指定されたデバイスから読み取ることができません。
31	0x0000001F	ERROR_GEN_FAILURE	システムに接続されたデバイスは機能していません。
32	0x00000020	ERROR_SHARING_VIOLATION	他のプロセスによってファイルが使用されているため、プロセスはファイルにアクセスできません。
33	0x00000021	ERROR_LOCK_VIOLATION	他のプロセスがファイルの一部をロックしているため、プロセスはファイルにアクセスできません。
34	0x00000022	ERROR_WRONG_DISK	間違ったディスクがドライブに入っています。%2 (ボリュームシリアル番号: %3) をドライブ %1 に挿入してください。
36	0x00000024	ERROR_SHARING_BUFFER_EXCEEDED	共有のために開いているファイルが多すぎます。
38	0x00000026	ERROR_HANDLE_EOF	ファイルの末尾に達しています。
39	0x00000027	ERROR_HANDLE_DISK_FULL	ディスクの空き領域がありません。
50	0x00000032	ERROR_NOT_SUPPORTED	リクエストがサポートされていません。
51	0x00000033	ERROR_REM_NOT_LIST	リモートコンピュータが使用できません。
52	0x00000034	ERROR_DUP_NAME	ネットワーク上に重複する名前が存在します。
53	0x00000035	ERROR_BAD_NETPATH	ネットワークパスが見つかりませんでした。

エラー			説明
10進数	16進数	名前	
54	0x00000036	ERROR_NETWORK_BUSY	ネットワークがビジーです。
55	0x00000037	ERROR_DEV_NOT_EXIST	指定されたネットワークリソースまたはデバイスは使用できません。
56	0x00000038	ERROR_TOO_MANY_CMDS	ネットワークBIOSコマンドが制限値に達しています。
57	0x00000039	ERROR_ADAP_HDW_ERR	ネットワークアダプタのハードウェアエラーが発生しました。
58	0x0000003A	ERROR_BAD_NET_RESP	指定されたサーバは要求された動作を実行できません。
59	0x0000003B	ERROR_UNEXP_NET_ERR	予期せぬネットワークエラーが発生しました。
60	0x0000003C	ERROR_BAD_REM_ADAP	リモートアダプタが互換性がありません。
61	0x0000003D	ERROR_PRINTQ_FULL	プリンタキューの空き領域がありません。
62	0x0000003E	ERROR_NO_SPOOL_SPACE	印刷待ちのファイルを保存するスペースがサーバで使用できません。
63	0x0000003F	ERROR_PRINT_CANCELLED	印刷を待機中のファイルが削除されています。
64	0x00000040	ERROR_NETNAME_DELETED	指定されたネットワーク名が使用できません。
65	0x00000041	ERROR_NETWORK_ACCESS_DENIED	ネットワークアクセスが拒否されました。
66	0x00000042	ERROR_BAD_DEV_TYPE	ネットワークのリソースタイプが正しくありません。
67	0x00000043	ERROR_BAD_NET_NAME	ネットワーク名が見つかりません。
68	0x00000044	ERROR_TOO_MANY_NAMES	ローカルコンピュータのネットワークアダプタカードの名前の制限を超えました。
69	0x00000045	ERROR_TOO_MANY_SESS	ネットワークBIOSセッションの制限を超えました。
70	0x00000046	ERROR_SHARING_PAUSED	リモートサーバが停止しているか、または起動プロセス中です。
71	0x00000047	ERROR_REQ_NOT_ACCEP	コンピュータが許容可能な接続数に既に達しているため、現時点でこれ以上多くの接続をこのリモートコンピュータに対して行うことはできません。
72	0x00000048	ERROR_REDIR_PAUSED	指定されたプリンタまたはディスクデバイスが一時停止しています。
80	0x00000050	ERROR_FILE_EXISTS	ファイルは存在しています。
82	0x00000052	ERROR_CANNOT_MAKE	ディレクトリまたはファイルを作成できません。
83	0x00000053	ERROR_FAIL_I24	INT 24での障害。
84	0x00000054	ERROR_OUT_OF_STRUCTURES	このリクエストを処理するためのストレージを使用できません。
85	0x00000055	ERROR_ALREADY_ASSIGNED	ローカルデバイス名は既に使用されています。
86	0x00000056	ERROR_INVALID_PASSWORD	指定されたネットワークのパスワードが正しくありません。
87	0x00000057	ERROR_INVALID_PARAMETER	パラメータが正しくありません。
88	0x00000058	ERROR_NET_WRITE_FAULT	ネットワークで書き込みエラーが発生しました。
89	0x00000059	ERROR_NO_PROC_SLOTS	システムはこの時点で別のプロセスを起動できません。

エラー			説明
10進数	16進数	名前	
100	0x00000064	ERROR_TOO_MANY_SEMAPHORES	別のシステムセマフォを作成できません。
101	0x00000065	ERROR_EXCL_SEM_ALREADY_OWNED	排他的セマフォが他のプロセスによって所有されています。
102	0x00000066	ERROR_SEM_IS_SET	セマフォが設定されていて終了できません。
103	0x00000067	ERROR_TOO_MANY_SEM_REQUESTS	セマフォを再設定できません。
104	0x00000068	ERROR_INVALID_AT_INTERRUPT_TIME	割り込み時に排他的セマフォを要求できません。
105	0x00000069	ERROR_SEM_OWNER_DIED	このセマフォの以前の所有権が終了しています。
106	0x0000006A	ERROR_SEM_USER_LIMIT	ドライブ %1 にディスクを挿入してください。
107	0x0000006B	ERROR_DISK_CHANGE	代替りのディスクが挿入されていないため、プログラムが停止しました。
108	0x0000006C	ERROR_DRIVE_LOCKED	ディスクは他のプロセスによって使用中かロックされています。
109	0x0000006D	ERROR_BROKEN_PIPE	パイプが終了しています。

		エラー		説明
10進数	16進数	名前		
110	0x0000006E	ERROR_OPEN_FAILED		システムは指定されたデバイスまたはファイルを開くことができません。
111	0x0000006F	ERROR_BUFFER_OVERFLOW		ファイル名が長すぎます。
112	0x00000070	ERROR_DISK_FULL		ディスクに十分なスペースがありません。
113	0x00000071	ERROR_NO_MORE_SEARCH_HANDLES		内部ファイルIDはこれ以上使用できません。
114	0x00000072	ERROR_INVALID_TARGET_HANDLE		ターゲット内部ファイルIDが正しくありません。
117	0x00000075	ERROR_INVALID_CATEGORY		アプリケーションプログラムによって行なわれたIOCTL呼び出しが正しくありません。
118	0x00000076	ERROR_INVALID_VERIFY_SWITCH		verify-on-writeスイッチのパラメータ値が正しくありません。
119	0x00000077	ERROR_BAD_DRIVER_LEVEL		システムは要求されたコマンドをサポートしていません。
120	0x00000078	ERROR_CALL_NOT_IMPLEMENTED		この関数はこのシステムではサポートされていません。
121	0x00000079	ERROR_SEM_TIMEOUT		セマフォタイムアウト期間を過ぎてきます。
122	0x0000007A	ERROR_INSUFFICIENT_BUFFER		システムコールに渡されたデータ領域が小さすぎます。
123	0x0000007B	ERROR_INVALID_NAME		ファイル名、ディレクトリ名、またはボリュームラベルの構文が正しくありません。
124	0x0000007C	ERROR_INVALID_LEVEL		システムコールのレベルが正しくありません。
125	0x0000007D	ERROR_NO_VOLUME_LABEL		ディスクにボリュームラベルがありません。
126	0x0000007E	ERROR_MOD_NOT_FOUND		指定されたモジュールが見つかりませんでした。
127	0x0000007F	ERROR_PROC_NOT_FOUND		指定されたプロシージャが見つかりませんでした。
128	0x00000080	ERROR_WAIT_NO_CHILDREN		待機する子プロセスはありません。
129	0x00000081	ERROR_CHILD_NOT_COMPLETE		%1 アプリケーションはWin32モードでは実行できません。
130	0x00000082	ERROR_DIRECT_ACCESS_HANDLE		ローディスクI/O以外の操作のためにディスクパーティションを開くファイルハンドルを使用しようとしています。
131	0x00000083	ERROR_NEGATIVE_SEEK		フィルの開始前にファイルポインタを移動しようとしてしました。
132	0x00000084	ERROR_SEEK_ON_DEVICE		ファイルポインタは指定されたデバイスまたはファイルに設定できません。
133	0x00000085	ERROR_IS_JOIN_TARGET		JOINまたはSUBSTコマンドは、以前に結合されたドライブを含むドライブには使用できません。
134	0x00000086	ERROR_IS_JOINED		既に結合されたドライブでJOINまたはSUBSTコマンドを使用しようとしてしました。
135	0x00000087	ERROR_IS_SUBSTED		既に置換されたドライブでJOINまたはSUBSTコマンドを使用しようとしてしました。
136	0x00000088	ERROR_NOT_JOINED		システムは結合されていないドライブのJOINを削除しようとしてしました。
137	0x00000089	ERROR_NOT_SUBSTED		システムは置換されていないドライブの置換を削除しようとしてしました。
138	0x0000008A	ERROR_JOIN_TO_JOIN		システムは結合されたドライブ上のディレクトリにドライブを結合しようとしてしました。
139	0x0000008B	ERROR_SUBST_TO_SUBST		システムは置換されたドライブ上のディレクトリにドライブを置換しようとしてしました。
140	0x0000008C	ERROR_JOIN_TO_SUBST		システムは置換されたドライブ上のディレクトリにドライブを結合しようとしてしました。
141	0x0000008D	ERROR_SUBST_TO_JOIN		システムは結合されたドライブ上のディレクトリにドライブをSUBST(置換)しようとしてしました。
142	0x0000008E	ERROR_BUSY_DRIVE		システムはこの時点ではJOINまたはSUBSTを実行できません。
143	0x0000008F	ERROR_SAME_DRIVE		システムは、同じドライブ上のディレクトリにドライブを結合または置換できません。
144	0x00000090	ERROR_DIR_NOT_ROOT		ディレクトリはルートディレクトリのサブディレクトリではありません。
145	0x00000091	ERROR_DIR_NOT_EMPTY		ディレクトリが空ではありません。
146	0x00000092	ERROR_IS_SUBST_PATH		指定されたパスは、置換で使用されています。

エラー			説明
10進数	16進数	名前	
147	0x00000093	ERROR_IS_JOIN_PATH	このコマンドを処理するために十分なリソースを使用できません。
148	0x00000094	ERROR_PATH_BUSY	指定されたパスはこの時点では使用できません。
149	0x00000095	ERROR_IS_SUBST_TARGET	ドライブ上のディレクトリが以前の置換のターゲットであったドライブを結合または置換しようとしていました。
150	0x00000096	ERROR_SYSTEM_TRACE	システムトレース情報がCONFIG.SYSファイルに指定されていないか、またはトレースが認められていません。
151	0x00000097	ERROR_INVALID_EVENT_COUNT	DosMuxSemWaitに指定されたセマフォイベントの数が正しくありません。
152	0x00000098	ERROR_TOO_MANY_MUXWAITERS	DosMuxSemWaitが実行されませんでした。既に設定されているセマフォが多すぎます。
153	0x00000099	ERROR_INVALID_LIST_FORMAT	DosMuxSemWaitリストが正しくありません。
154	0x0000009A	ERROR_LABEL_TOO_LONG	入力したボリュームラベルが、ターゲットファイルシステムのラベル文字の制限を超えています。
155	0x0000009B	ERROR_TOO_MANY_TCBS	別のスレッドを作成できません。
156	0x0000009C	ERROR_SIGNAL_REFUSED	受信プロセスが信号の受信を拒否しました。
157	0x0000009D	ERROR_DISCARDED	セグメントは既に破棄されていてロックできません。
158	0x0000009E	ERROR_NOT_LOCKED	セグメントは既にロック解除されています。
159	0x0000009F	ERROR_BAD_THREADID_ADDR	スレッドIDのアドレスが正しくありません。
160	0x000000A0	ERROR_BAD_ARGUMENTS	DosExecPgmに渡された引数文字列が正しくありません。
161	0x000000A1	ERROR_BAD_PATHNAME	指定されたパスが無効です。
162	0x000000A2	ERROR_SIGNAL_PENDING	信号は既に保留中です。
164	0x000000A4	ERROR_MAX_THRDS_REACHED	システムにスレッドをこれ以上作成できません。
167	0x000000A7	ERROR_LOCK_FAILED	ファイルの一部の領域をロックできません。
170	0x000000AA	ERROR_BUSY	要求されたリソースは使用中です。
173	0x000000AD	ERROR_CANCEL_VIOLATION	提供されたキャンセル範囲に対するロックリクエストが未処理の状態ではありませんでした。
174	0x000000AE	ERROR_ATOMIC_LOCKS_NOT_SUPPORTED	ファイルシステムはロックタイプへのアトミックチェンジをサポートしていません。
180	0x000000B4	ERROR_INVALID_SEGMENT_NUMBER	システムが正しくないセグメント番号を検出しました。
182	0x000000B6	ERROR_INVALID_ORDINAL	このオペレーティング システムでは %1 は実行されません。
183	0x000000B7	ERROR_ALREADY_EXISTS	そのファイルが既に存在している場合は、ファイルを作成できません。
186	0x000000BA	ERROR_INVALID_FLAG_NUMBER	渡されたフラグが正しくありません。
187	0x000000BB	ERROR_SEM_NOT_FOUND	指定されたシステムセマフォ名が見つかりませんでした。
188	0x000000BC	ERROR_INVALID_STARTING_CODESEG	このオペレーティング システムでは %1 は実行されません。
189	0x000000BD	ERROR_INVALID_STACKSEG	このオペレーティング システムでは %1 は実行されません。
190	0x000000BE	ERROR_INVALID_MODULETYPE	このオペレーティング システムでは %1 は実行されません。
191	0x000000BF	ERROR_INVALID_EXE_SIGNATURE	Win32モードでは %1 を実行できません。
192	0x000000C0	ERROR_EXE_MARKED_INVALID	このオペレーティング システムでは %1 は実行されません。
193	0x000000C1	ERROR_BAD_EXE_FORMAT	%1 は有効なWin32アプリケーションではありません。
194	0x000000C2	ERROR_ITERATED_DATA_EXCEEDS_64k	このオペレーティング システムでは %1 は実行されません。
195	0x000000C3	ERROR_INVALID_MINALLOCSIZE	このオペレーティング システムでは %1 は実行されません。
196	0x000000C4	ERROR_DYNLINK_FROM_INVALID_RING	このオペレーティング システムではこのアプリケーション プログラムは実行されません。
197	0x000000C5	ERROR_IOPL_NOT_ENABLED	OSは現在、このアプリケーションを実行するように構成されていません。

エラー			説明
10進数	16進数	名前	
198	0x000000C6	ERROR_INVALID_SEGDPL	このオペレーティング システムでは %1 は実行されません。
199	0x000000C7	ERROR_AUTODATASEG_EXCEEDS_64k	OSはこのアプリケーションプログラムを実行できません。

エラー			説明
10進数	16進数	名前	
200	0x000000C8	ERROR_RING2SEG_MUST_BE_MOVABLE	コードセグメントは64K以上にすることはできません。
201	0x000000C9	ERROR_RELOC_CHAIN_XEEDS_SEGLIM	このオペレーティング システムでは %1 は実行されません。
202	0x000000CA	ERROR_INFLOOP_IN_RELOC_CHAIN	このオペレーティング システムでは %1 は実行されません。
203	0x000000CB	ERROR_ENVVAR_NOT_FOUND	システムは入力された環境オプションを見つけることができませんでした。
205	0x000000CD	ERROR_NO_SIGNAL_SENT	コマンドサブツリーには信号ハンドラを持つプロセスがありません。
206	0x000000CE	ERROR_FILENAME_EXCED_RANGE	ファイル名または拡張子が長すぎます。
207	0x000000CF	ERROR_RING2_STACK_IN_USE	リング2スタックは使用中です。
208	0x000000D0	ERROR_META_EXPANSION_TOO_LONG	グローバルファイル名の文字「*」または「?」が誤って入力されたか、または指定されているグローバルファイル名の文字が多すぎます。
209	0x000000D1	ERROR_INVALID_SIGNAL_NUMBER	ポストされている信号が正しくありません。
210	0x000000D2	ERROR_THREAD_1_INACTIVE	信号ハンドラを設定できません。
212	0x000000D4	ERROR_LOCKED	セグメントはロックされていて再割り当てできません。
214	0x000000D6	ERROR_TOO_MANY_MODULES	このプログラムまたはダイナミックリンクモジュールに付加されているダイナミックリンクモジュールが多すぎます。
215	0x000000D7	ERROR_NESTING_NOT_ALLOWED	LoadModuleへの呼び出しをネストできません。
216	0x000000D8	ERROR_EXE_MACHINE_TYPE_MISMATCH	イメージファイル %1 は、現在のマシン以外のマシンタイプに対して有効です。
230	0x000000E6	ERROR_BAD_PIPE	パイプの状態が無効です。
231	0x000000E7	ERROR_PIPE_BUSY	すべてのパイプインスタンスがビジーです。
232	0x000000E8	ERROR_NO_DATA	パイプが閉じられています。
233	0x000000E9	ERROR_PIPE_NOT_CONNECTED	パイプの反対側にプロセスがありません。
234	0x000000EA	ERROR_MORE_DATA	より多くのデータを使用可能です。
240	0x000000F0	ERROR_VC_DISCONNECTED	セッションはキャンセルされました。
254	0x000000FE	ERROR_INVALID_EA_NAME	指定された拡張属性名が無効です。
255	0x000000FF	ERROR_EA_LIST_INCONSISTENT	拡張属性の整合性がありません。
258	0x00000102	WAIT_TIMEOUT	待機動作がタイムアウトしました。
259	0x00000103	ERROR_NO_MORE_ITEMS	データはこれ以上使用できません。
266	0x0000010A	ERROR_CANNOT_COPY	コピー機能は使用できません。
267	0x0000010B	ERROR_DIRECTORY	ディレクトリ名が無効です。
275	0x00000113	ERROR_EAS_DIDNT_FIT	拡張属性がバッファに適合しませんでした。
276	0x00000114	ERROR_EA_FILE_CORRUPT	マウントされたファイルシステムの拡張属性ファイルが破損しています。
277	0x00000115	ERROR_EA_TABLE_FULL	拡張属性テーブルファイルの空き領域がありません。
278	0x00000116	ERROR_INVALID_EA_HANDLE	指定された拡張属性ハンドルが無効です。
282	0x0000011A	ERROR_EAS_NOT_SUPPORTED	マウントされたファイルシステムは拡張属性をサポートしていません。
288	0x00000120	ERROR_NOT_OWNER	呼び出し側によって所有されていないミューテックスを解放しようとしていました。
298	0x0000012A	ERROR_TOO_MANY_POSTS	セマフォに対して行われたポストが多すぎます。
299	0x0000012B	ERROR_PARTIAL_COPY	ReadProcessMemoryまたはWriteProcessMemoryリクエストの一部のみが完了しました。
300	0x0000012C	ERROR_OPLOCK_NOT_GRANTED	oplock要求が拒否されました。
301	0x0000012D	ERROR_INVALID_OPLOCK_PROTOCOL	無効なoplock肯定応答がシステムによって受信されました。

エラー			説明
10進数	16進数	名前	
302	0x0000012E	ERROR_DISK_TOO_FRAGMENTED	この操作を実行するにはボリュームが断片化されすぎています。
303	0x0000012F	ERROR_DELETE_PENDING	ファイルのプロセスが削除されたため、ファイルを開くことができません。
317	0x0000013D	ERROR_MR_MID_NOT_FOUND	システムは、%2 のメッセージファイル内のメッセージ番号 0x%1 に対応するメッセージテキストを見つけることができません。
487	0x000001E7	ERROR_INVALID_ADDRESS	無効なアドレスにアクセスしようとした。
534	0x00000216	ERROR_ARITHMETIC_OVERFLOW	計算結果が32ビットを超えました。
535	0x00000217	ERROR_PIPE_CONNECTED	パイプの反対側にプロセスがあります。
536	0x00000218	ERROR_PIPE_LISTENING	パイプの反対側を開くプロセスを待機中です。
994	0x000003E2	ERROR_EA_ACCESS_DENIED	拡張属性へのアクセスが拒否されました。
995	0x000003E3	ERROR_OPERATION_ABORTED	スレッドの終了またはアプリケーションの要求のため、I/O操作が中止されました。
996	0x000003E4	ERROR_IO_INCOMPLETE	重複するI/Oイベントが信号状態ではありません。
997	0x000003E5	ERROR_IO_PENDING	重複するI/O操作が進行中です。
998	0x000003E6	ERROR_NOACCESS	メモリロケーションへの無効なアクセス。
999	0x000003E7	ERROR_SWAPERROR	inpage操作の実行時エラー

エラー			説明
10進数	16進数	名前	
1001	0x000003E9	ERROR_STACK_OVERFLOW	再帰が深すぎます。スタックがオーバーフローしました。
1002	0x000003EA	ERROR_INVALID_MESSAGE	ウィンドウが送信されたメッセージに回答できません。
1003	0x000003EB	ERROR_CAN_NOT_COMPLETE	この機能を完了できません。
1004	0x000003EC	ERROR_INVALID_FLAGS	無効なフラグ
1005	0x000003ED	ERROR_UNRECOGNIZED_VOLUME	ボリュームには認識されるファイルシステムが含まれていません。必要なすべてのファイルシステムドライバがロードされていることと、ボリュームが破損していないことを確認してください。
1006	0x000003EE	ERROR_FILE_INVALID	ファイルのボリュームが外部から変更されたため、開いているファイルが有効でなくなりました。
1007	0x000003EF	ERROR_FULLSCREEN_MODE	要求された操作は、フルスクリーンモードでは実行できません。
1008	0x000003F0	ERROR_NO_TOKEN	存在しないトークンを参照しようとした。
1009	0x000003F1	ERROR_BADDB	設定レジストリデータベースが破損しています。
1010	0x000003F2	ERROR_BADKEY	設定レジストリキーが無効です。
1011	0x000003F3	ERROR_CANTOPEN	設定レジストリキーを開くことができませんでした。
1012	0x000003F4	ERROR_CANTREAD	設定レジストリキーを読み取ることができませんでした。
1013	0x000003F5	ERROR_CANTWRITE	設定レジストリキーに書き込むことができませんでした。
1014	0x000003F6	ERROR_REGISTRY_RECOVERED	ログまたは代替コピーを使用して、レジストリデータベース内のファイルの1つを復元する必要があるがありました。復元に成功しました。
1015	0x000003F7	ERROR_REGISTRY_CORRUPT	レジストリが破損しています。レジストリデータを含んでいるファイルの1つの構造体が破損しているか、あるいは代替コピーまたはログが存在していないかまたは破損しているため、ファイルに関するシステムのメモリエイメージが復元できません。
1016	0x000003F8	ERROR_REGISTRY_IO_FAILED	レジストリによって開始されたI/O操作が失敗し、回復不能です。レジストリは、レジストリに関するシステムのイメージを含むファイルの1つをリードイン、リードアウト、またはフラッシュすることができませんでした。
1017	0x000003F9	ERROR_NOT_REGISTRY_FILE	システムはファイルをレジストリにロードまたは復元しようとしたが、指定されたファイルがレジストリファイル形式になっていません。

		エラー		説明
10進数	16進数	名前		
1018	0x000003FA	ERROR_KEY_DELETED		削除のマークが付いたレジストリキーに対して、無効な操作を実行しようとした。
1019	0x000003FB	ERROR_NO_LOG_SPACE		システムはレジストリログに必要なスペースを割り当てることができませんでした。
1020	0x000003FC	ERROR_KEY_HAS_CHILDREN		既にサブキーまたは値を持つレジストリキーにはシンボリックリンクを作成できません。
1021	0x000003FD	ERROR_CHILD_MUST_BE_VOLATILE		揮発性親キーの下に安定したサブキーを作成することはできません。
1022	0x000003FE	ERROR_NOTIFY_ENUM_DIR		変更通知リクエストが完了し、その情報が呼び出し側のバッファに返されていません。呼び出し側は、現在、変更を探すファイルを列挙する必要があります。
1051	0x0000041B	ERROR_DEPENDENT_SERVICES_RUNNING		他の実行中のサービスに依存するサービスに対して停止制御が送信されました。
1052	0x0000041C	ERROR_INVALID_SERVICE_CONTROL		要求された操作はこのサービスでは有効ではありません。
1053	0x0000041D	ERROR_SERVICE_REQUEST_TIMEOUT		サービスが所定の時間内に開始または制御リクエストに対して応答しませんでした。
1054	0x0000041E	ERROR_SERVICE_NO_THREAD		このサービスにはスレッドを作成できませんでした。
1055	0x0000041F	ERROR_SERVICE_DATABASE_LOCKED		サービスデータベースがロックされています。
1056	0x00000420	ERROR_SERVICE_ALREADY_RUNNING		サービスのインスタンスが既に実行中です。
1057	0x00000421	ERROR_INVALID_SERVICE_ACCOUNT		アカウント名が無効または存在しないか、または指定されたアカウント名に対してパスワードが無効です。
1058	0x00000422	ERROR_SERVICE_DISABLED		サービスが無効になっているか、または関連付けられているデバイスが有効になっていないため、サービスを開始できません。
1059	0x00000423	ERROR_CIRCULAR_DEPENDENCY		循環するサービス依存関係が指定されました。
1060	0x00000424	ERROR_SERVICE_DOES_NOT_EXIST		指定されたサービスは、インストールされたサービスとして存在していません。
1061	0x00000425	ERROR_SERVICE_CANNOT_ACCEPT_CTRL		サービスはこの時点で制御メッセージを受理できません。
1062	0x00000426	ERROR_SERVICE_NOT_ACTIVE		サービスが開始していません。
1063	0x00000427	ERROR_FAILED_SERVICE_CONTROLLER_CONNECT		サービスプロセスがサービスコントローラに接続できませんでした。
1064	0x00000428	ERROR_EXCEPTION_IN_SERVICE		制御リクエストの処理中にサービスで例外が発生しました。
1065	0x00000429	ERROR_DATABASE_DOES_NOT_EXIST		指定されたデータベースは存在しません。
1066	0x0000042A	ERROR_SERVICE_SPECIFIC_ERROR		サービスがサービス固有のエラーコードを返しました。
1067	0x0000042B	ERROR_PROCESS_ABORTED		プロセスが予期せず終了しました。
1068	0x0000042C	ERROR_SERVICE_DEPENDENCY_FAIL		依存関係サービスまたはグループが開始できませんでした。
1069	0x0000042D	ERROR_SERVICE_LOGON_FAILED		ログオンに失敗したため、サービスが開始しませんでした。
1070	0x0000042E	ERROR_SERVICE_START_HANG		開始後、サービスが開始保留状態でハングしました。
1071	0x0000042F	ERROR_INVALID_SERVICE_LOCK		指定されたサービスデータベースロックは無効です。
1072	0x00000430	ERROR_SERVICE_MARKED_FOR_DELETE		指定されたサービスは、削除のマークが付けられています。
1073	0x00000431	ERROR_SERVICE_EXISTS		指定されたサービスは既に存在しています。
1074	0x00000432	ERROR_ALREADY_RUNNING_LKG		システムは、現在、Last Known Good Configuration (前回正常起動時の構成)で実行中です。
1075	0x00000433	ERROR_SERVICE_DEPENDENCY_DELETED		依存するサービスが存在しないか、または削除のマークが付けられています。
1076	0x00000434	ERROR_BOOT_ALREADY_ACCEPTED		現在のブートが、前回正常起動時の制御セットとして既に使用を承認されています。
1077	0x00000435	ERROR_SERVICE_NEVER_STARTED		前回のブート以後、サービスを開始する試行が行われていません。

エラー			説明
10進数	16進数	名前	
1078	0x00000436	ERROR_DUPLICATE_SERVICE_NAME	その名前は既にサービス名またはサービス表示名として使用されています。
1079	0x00000437	ERROR_DIFFERENT_SERVICE_ACCOUNT	このサービスに対して指定されたアカウントが、同じプロセスで実行中の他のサービスに対して指定されたアカウントと異なります。
1080	0x00000438	ERROR_CANNOT_DETECT_DRIVER_FAILURE	障害アクションは、ドライバに対してではなく、Win32サービスに対してのみ設定できます。
1081	0x00000439	ERROR_CANNOT_DETECT_PROCESS_ABORT	このサービスは、サービス制御マネージャと同じプロセスで実行されています。そのため、サービス制御マネージャは、このサービスのプロセスが予期せず終了した場合にアクションを取ることができません。
1082	0x0000043A	ERROR_NO_RECOVERY_PROGRAM	このサービスには復元プログラムが設定されていません。
1083	0x0000043B	ERROR_SERVICE_NOT_IN_EXE	このサービスを実行するように設定されている実行可能なプログラムにサービスが実装されていません。
1084	0x0000043C	ERROR_NOT_SAFEBOOT_SERVICE	このサービスはセーフモードでは開始できません。

エラー			説明
10進数	16進数	名前	
1100	0x0000044C	ERROR_END_OF_MEDIA	テープの物理的な末端に達しました。
1101	0x0000044D	ERROR_FILEMARK_DETECTED	テープアクセスがファイルマークに達しました。
1102	0x0000044E	ERROR_BEGINNING_OF_MEDIA	テープまたはパーティションの開始位置に達しました。
1103	0x0000044F	ERROR_SETMARK_DETECTED	テープアクセスが一連のファイルの末尾に達しました。
1104	0x00000450	ERROR_NO_DATA_DETECTED	テープにはデータがこれ以上ありません。
1105	0x00000451	ERROR_PARTITION_FAILURE	テープにパーティションを設定できませんでした。
1106	0x00000452	ERROR_INVALID_BLOCK_LENGTH	マルチボリュームパーティションの新しいテープにアクセスする場合、現在のブロックサイズが正しくありません。
1107	0x00000453	ERROR_DEVICE_NOT_PARTITIONED	テープのロード時にテープパーティション情報を見つけることができませんでした。
1108	0x00000454	ERROR_UNABLE_TO_LOCK_MEDIA	メディアイジェクト機構をロックできません。
1109	0x00000455	ERROR_UNABLE_TO_UNLOAD_MEDIA	メディアをアンロードできません。
1110	0x00000456	ERROR_MEDIA_CHANGED	ドライブのメディアが変更された可能性があります。
1111	0x00000457	ERROR_BUS_RESET	I/Oバスがリセットされました。
1112	0x00000458	ERROR_NO_MEDIA_IN_DRIVE	ドライブにメディアがありません。
1113	0x00000459	ERROR_NO_UNICODE_TRANSLATION	Unicode文字のマッピングが、ターゲットのマルチバイトコードページに存在していません。
1114	0x0000045A	ERROR_DLL_INIT_FAILED	ダイナミックリンクライブラリ (DLL) の初期化ルーチンが失敗しました。
1115	0x0000045B	ERROR_SHUTDOWN_IN_PROGRESS	システムをシャットダウン中です。
1116	0x0000045C	ERROR_NO_SHUTDOWN_IN_PROGRESS	進行中のシャットダウンがないため、システムシャットダウンを中止することができません。
1117	0x0000045D	ERROR_IO_DEVICE	I/Oデバイスエラーのため、リクエストを実行できませんでした。
1118	0x0000045E	ERROR_SERIAL_NO_DEVICE	正常に初期化されたシリアルデバイスがありません。シリアルドライバがアンロードされます。
1119	0x0000045F	ERROR_IRQ_BUSY	他のデバイスと割り込みリクエスト (IRQ) を共有するデバイスを開くことができません。IRQを使用する少なくとも1つの他のデバイスが既に開かれています。
1120	0x00000460	ERROR_MORE_WRITES	シリアルI/O操作がこのシリアルポートへの別の書き込みによって完了しています。(IOCTL_SERIAL_XOFF_COUNTERがゼロに達しました。)

エラー			説明
10進数	16進数	名前	
1121	0x00000461	ERROR_COUNTER_TIMEOUT	タイムアウト期限が過ぎているため、シリアルI/O操作が完了しています。(IOCTL_SERIAL_XOFF_COUNTERがゼロに達していません。)
1122	0x00000462	ERROR_FLOPPY_ID_MARK_NOT_FOUND	フロッピーディスク上にIDアドレスマークが見つかりませんでした。
1123	0x00000463	ERROR_FLOPPY_WRONG_CYLINDER	フロッピーディスクのセクタIDフィールドとフロッピーディスクのコントローラトラックアドレスが一致しません。
1124	0x00000464	ERROR_FLOPPY_UNKNOWN_ERROR	フロッピーディスクのコントローラがフロッピーディスクのドライバによって認識されないエラーを報告しました。
1125	0x00000465	ERROR_FLOPPY_BAD_REGISTERS	フロッピーディスクのコントローラがレジスタに整合性のない結果を返しました。
1126	0x00000466	ERROR_DISK_RECALIBRATE_FAILED	ハードディスクへのアクセス時に再校正操作に失敗し、再試行を繰り返しましたが成功しませんでした。
1127	0x00000467	ERROR_DISK_OPERATION_FAILED	ハードディスクへのアクセス時にディスク操作に失敗し、再試行を繰り返しましたが成功しませんでした。
1128	0x00000468	ERROR_DISK_RESET_FAILED	ハードディスクにアクセスする間に、ディスクコントローラのリセットが必要ですが、それにも失敗しました。
1129	0x00000469	ERROR_EOM_OVERFLOW	テープの物理的な終端に達しました。
1130	0x0000046A	ERROR_NOT_ENOUGH_SERVER_MEMORY	このコマンドを処理するために十分なサーバストレージを使用できません。
1131	0x0000046B	ERROR_POSSIBLE_DEADLOCK	潜在的なデッドロック状態が検出されました。
1132	0x0000046C	ERROR_MAPPED_ALIGNMENT	指定されたベースアドレスまたはファイルオフセットのアライメントが適切ではありません。
1140	0x00000474	ERROR_SET_POWER_STATE_VETOED	システムの電源状態を変更しようとしたのですが、別のアプリケーションまたはドライブによって拒否されました。
1141	0x00000475	ERROR_SET_POWER_STATE_FAILED	システムBIOSがシステム電源状態の変更失敗しました。
1142	0x00000476	ERROR_TOO_MANY_LINKS	ファイルシステムがサポートするリンク数より多いリンクをファイルに作成しようとした。
1150	0x0000047E	ERROR_OLD_WIN_VERSION	指定されたプログラムには新しいWindowsバージョンが必要です。
1151	0x0000047F	ERROR_APP_WRONG_OS	指定されたプログラムは、WindowsまたはMS-DOSプログラムではありません。
1152	0x00000480	ERROR_SINGLE_INSTANCE_APP	指定されたプログラムの1よりも多いインスタンスを開始することができません。
1153	0x00000481	ERROR_RMODE_APP	指定されたプログラムは、以前のWindowsバージョン用に書かれています。
1154	0x00000482	ERROR_INVALID_DLL	このアプリケーションを実行するために必要なライブラリファイルの1つが破損しています。
1155	0x00000483	ERROR_NO_ASSOCIATION	この操作に指定されたファイルにはアプリケーションが関連付けられていません。
1156	0x00000484	ERROR_DDE_FAIL	アプリケーションへのコマンド送信時にエラーが発生しました。
1157	0x00000485	ERROR_DLL_NOT_FOUND	このアプリケーションを実行するために必要なライブラリファイルの1つが見つかりません。
1158	0x00000486	ERROR_NO_MORE_USER_HANDLES	現在のプロセスは、ハンドルのすべてのシステム割当量をWindow Managerオブジェクトに使用しています。
1159	0x00000487	ERROR_MESSAGE_SYNC_ONLY	このメッセージは同期動作でのみ使用することができます。
1160	0x00000488	ERROR_SOURCE_ELEMENT_EMPTY	指示されたソースエレメントにはメディアがありません。
1161	0x00000489	ERROR_DESTINATION_ELEMENT_FULL	指示された保存先エレメントには既にメディアが含まれています。
1162	0x0000048A	ERROR_ILLEGAL_ELEMENT_ADDRESS	指示されたエレメントは存在しません。

エラー			説明
10進数	16進数	名前	
1163	0x0000048B	ERROR_MAGAZINE_NOT_PRESENT	指示されたエレメントは、存在しないマガジンの一部です。
1164	0x0000048C	ERROR_DEVICE_REINITIALIZATION_NEEDED	指示されたデバイスは、ハードウェアエラーのため再初期化が必要です。
1165	0x0000048D	ERROR_DEVICE_REQUIRES_CLEANING	デバイスは、操作が試行される前にクリーニングが必要であることを示しています。
1166	0x0000048E	ERROR_DEVICE_DOOR_OPEN	デバイスはドアが開いていることを示しています。
1167	0x0000048F	ERROR_DEVICE_NOT_CONNECTED	デバイスは接続されていません。
1168	0x00000490	ERROR_NOT_FOUND	エレメントが見つかりませんでした。
1169	0x00000491	ERROR_NO_MATCH	インデックスには指定されたキーに一致するものはありません。
1170	0x00000492	ERROR_SET_NOT_FOUND	指定されたプロパティセットはオブジェクトには存在しません。
1171	0x00000493	ERROR_POINT_NOT_FOUND	GetMouseMovePointsExに渡すポイントがバッファ内にありません。
1172	0x00000494	ERROR_NO_TRACKING_SERVICE	トラッキング(ワークステーション)サービスは実行されていません。
1173	0x00000495	ERROR_NO_VOLUME_ID	ボリュームIDが見つかりませんでした。
1175	0x00000497	ERROR_UNABLE_TO_REMOVE_REPLACED	置き換えられるファイルを削除できません。
1176	0x00000498	ERROR_UNABLE_TO_MOVE_REPLACEMENT	置き換えるファイルを置き換えられるファイルに移動できません。置き換えられるファイルは元の名前を保持しています。
1177	0x00000499	ERROR_UNABLE_TO_MOVE_REPLACEMENT_2	置き換えるファイルを置き換えられるファイルに移動できません。置き換えられるファイルは、バックアップ名を使用して名前が変更されています。
1178	0x0000049A	ERROR_JOURNAL_DELETE_IN_PROGRESS	ボリューム変更ジャーナルを削除中です。
1179	0x0000049B	ERROR_JOURNAL_NOT_ACTIVE	ボリューム変更ジャーナルがアクティブではありません。
1180	0x0000049C	ERROR_POTENTIAL_FILE_FOUND	ファイルが見つかりましたが、正しいファイルではない可能性があります。
1181	0x0000049D	ERROR_JOURNAL_ENTRY_DELETED	ジャーナルエントリがジャーナルから削除されています。

エラー			説明
10進数	16進数	名前	
1200	0x000004B0	ERROR_BAD_DEVICE	指定されたデバイス名が無効です。
1201	0x000004B1	ERROR_CONNECTION_UNAVAIL	デバイスは現在接続されていませんが、接続が記憶されています。
1202	0x000004B2	ERROR_DEVICE_ALREADY_REMEMBERED	ローカルデバイスに別のネットワークリソースへの接続が記憶されています。
1203	0x000004B3	ERROR_NO_NET_OR_BAD_PATH	受け取ったネットワークパスを承認したネットワークプロバイダはありません。
1204	0x000004B4	ERROR_BAD_PROVIDER	指定されたネットワークプロバイダ名が無効です。
1205	0x000004B5	ERROR_CANNOT_OPEN_PROFILE	ネットワーク接続プロファイルを開くことができません。
1206	0x000004B6	ERROR_BAD_PROFILE	ネットワーク接続プロファイルが破損しています。
1207	0x000004B7	ERROR_NOT_CONTAINER	非コンテナを列挙できません。
1208	0x000004B8	ERROR_EXTENDED_ERROR	拡張エラーが発生しました。
1209	0x000004B9	ERROR_INVALID_GROUPNAME	指定されたグループ名の形式が無効です。
1210	0x000004BA	ERROR_INVALID_COMPUTERNAME	指定されたコンピュータ名の形式が無効です。
1211	0x000004BB	ERROR_INVALID_EVENTNAME	指定されたイベント名の形式が無効です。
1212	0x000004BC	ERROR_INVALID_DOMAINNAME	指定されたドメイン名の形式が無効です。
1213	0x000004BD	ERROR_INVALID_SERVICENAME	指定されたサービス名の形式が無効です。
1214	0x000004BE	ERROR_INVALID_NETNAME	指定されたネットワーク名の形式が無効です。
1215	0x000004BF	ERROR_INVALID_SHARENAME	指定された共有名の形式が無効です。
1216	0x000004C0	ERROR_INVALID_PASSWORDNAME	指定されたパスワードの形式が無効です。
1217	0x000004C1	ERROR_INVALID_MESSAGE_NAME	指定されたメッセージ名の形式が無効です。

エラー			説明
10進数	16進数	名前	
1218	0x000004C2	ERROR_INVALID_MESSAGEDEST	指定されたメッセージ宛先の形式が無効です。
1219	0x000004C3	ERROR_SESSION_CREDENTIAL_CONFLICT	提供された資格情報が、既存の資格情報のセットと競合しています。
1220	0x000004C4	ERROR_REMOTE_SESSION_LIMIT_EXCEEDED	ネットワークサーバにセッションを確立しようとしたが、サーバに既に確立されているセッションが多すぎます。
1221	0x000004C5	ERROR_DUP_DOMAINNAME	ワークグループまたはドメイン名はネットワーク上の他のコンピュータによって既に使用されています。
1222	0x000004C6	ERROR_NO_NETWORK	ネットワークは存在しないか、または開始されていません。
1223	0x000004C7	ERROR_CANCELLED	操作はユーザによってキャンセルされました。
1224	0x000004C8	ERROR_USER_MAPPED_FILE	要求された操作は、ユーザマッピングされたセクションが開いたファイルで実行できません。
1225	0x000004C9	ERROR_CONNECTION_REFUSED	リモートシステムがネットワーク接続を拒否しました。
1226	0x000004CA	ERROR_GRACEFUL_DISCONNECT	ネットワーク接続は正常に終了しました。
1227	0x000004CB	ERROR_ADDRESS_ALREADY_ASSOCIATED	ネットワーク転送エンドポイントは既にアドレスが関連付けられています。
1228	0x000004CC	ERROR_ADDRESS_NOT_ASSOCIATED	ネットワークエンドポイントにはまだアドレスが関連付けられていません。
1229	0x000004CD	ERROR_CONNECTION_INVALID	存在しないネットワーク接続に操作を実行しようとした。
1230	0x000004CE	ERROR_CONNECTION_ACTIVE	アクティブなネットワーク接続で無効な操作を実行しようとした。
1231	0x000004CF	ERROR_NETWORK_UNREACHABLE	ネットワークロケーションに到達できません。ネットワークのトラブルシューティングについては、Windows Helpを参照してください。
1232	0x000004D0	ERROR_HOST_UNREACHABLE	ネットワークロケーションに到達できません。ネットワークのトラブルシューティングについては、Windows Helpを参照してください。
1233	0x000004D1	ERROR_PROTOCOL_UNREACHABLE	ネットワークロケーションに到達できません。ネットワークのトラブルシューティングについては、Windows Helpを参照してください。
1234	0x000004D2	ERROR_PORT_UNREACHABLE	リモートシステム上の宛先ネットワークエンドポイントでサービスが動作していません。
1235	0x000004D3	ERROR_REQUEST_ABORTED	リクエストは中止されました。
1236	0x000004D4	ERROR_CONNECTION_ABORTED	ネットワーク接続はローカルシステムによって中止されました。
1237	0x000004D5	ERROR_RETRY	操作を完了できませんでした。再試行してください。
1238	0x000004D6	ERROR_CONNECTION_COUNT_LIMIT	このアカウントに対する現在の接続数の制限に達しているため、サーバへの接続ができませんでした。
1239	0x000004D7	ERROR_LOGIN_TIME_RESTRICTION	このアカウントに対して承認されている時間以外にログインしようとした。
1240	0x000004D8	ERROR_LOGIN_WKSTA_RESTRICTION	アカウントはこのステーションからのログインを承認されていません。
1241	0x000004D9	ERROR_INCORRECT_ADDRESS	要求された操作にネットワークアドレスを使用できませんでした。
1242	0x000004DA	ERROR_ALREADY_REGISTERED	サービスは既に登録されています。
1243	0x000004DB	ERROR_SERVICE_NOT_FOUND	指定されたサービスは存在しません。
1244	0x000004DC	ERROR_NOT_AUTHENTICATED	ユーザが認証されていないため、要求された操作は実行されませんでした。
1245	0x000004DD	ERROR_NOT_LOGGED_ON	ユーザがネットワークにログオンしていないため、要求された操作は実行されませんでした。指定されたサービスは存在しません。
1246	0x000004DE	ERROR_CONTINUE	進行中の作業を続けます。
1247	0x000004DF	ERROR_ALREADY_INITIALIZED	初期化が既に完了しているのに初期化を実行しようとした。
1248	0x000004E0	ERROR_NO_MORE_DEVICES	ローカルデバイスはこれ以上ありません。
1249	0x000004E1	ERROR_NO_SUCH_SITE	指定されたサイトは存在しません。

エラー			説明
10進数	16進数	名前	
1250	0x000004E2	ERROR_DOMAIN_CONTROLLER_EXISTS	指定された名前のドメインコントローラは既に存在しています。
1251	0x000004E3	ERROR_ONLY_IF_CONNECTED	この操作は、サーバに接続している場合にのみサポートされます。
1252	0x000004E4	ERROR_OVERRIDE_NOCHANGES	グループポリシーフレームワークは、変更がない場合でも拡張を呼び出す必要があります。
1253	0x000004E5	ERROR_BAD_USER_PROFILE	指定されたユーザには有効なプロファイルがありません。
1254	0x000004E6	ERROR_NOT_SUPPORTED_ON_SBS	この操作は、Microsoft Small Business Serverではサポートされていません。
1255	0x000004E7	ERROR_SERVER_SHUTDOWN_IN_PROGRESS	サーバマシンはシャットダウン中です。
1256	0x000004E8	ERROR_HOST_DOWN	リモートシステムが使用できません。ネットワークのトラブルシューティングについては、Windows Helpを参照してください。
1257	0x000004E9	ERROR_NON_ACCOUNT_SID	提供されたセキュリティIDは、アカウントドメインのものではありません。
1258	0x000004EA	ERROR_NON_DOMAIN_SID	提供されたセキュリティIDにはドメインコンポーネントがありません。
1259	0x000004EB	ERROR_APPHELP_BLOCK	AppHelpダイアログがキャンセルされたため、アプリケーションが開始できません。
1260	0x000004EC	ERROR_ACCESS_DISABLED_BY_POLICY	要求されたリソースへのアクセスは、管理者によって無効になっています。
1261	0x000004ED	ERROR_REG_NAT_CONSUMPTION	プログラムが無効なレジスタ値を使用しようとしてしました。通常、このエラーは初期化されていないレジスタによって発生します。このエラーはItanium固有のエラーです。
1262	0x000004EE	ERROR_CSCSHARE_OFFLINE	共有は現在オフラインか、または存在しません。
1300	0x00000514	ERROR_NOT_ALL_ASSIGNED	参照されたすべての権限が呼び出し側に割り当てられていません。
1301	0x00000515	ERROR_SOME_NOT_MAPPED	アカウント名とセキュリティIDのマッピングの一部が完了していません。
1302	0x00000516	ERROR_NO_QUOTAS_FOR_ACCOUNT	システムの割り当て制限が、このアカウントに対して明確に設定されていません。
1303	0x00000517	ERROR_LOCAL_USER_SESSION_KEY	暗号化鍵が使用できません。周知の暗号化鍵は返されました。
1304	0x00000518	ERROR_NULL_LM_PASSWORD	パスワードが複雑すぎてLAN Managerパスワードに変換できません。LAN ManagerパスワードがNULL文字列を返しました。
1305	0x00000519	ERROR_UNKNOWN_REVISION	リビジョンレベルが不明です。
1306	0x0000051A	ERROR_REVISION_MISMATCH	2つのリビジョンレベルが互換性がないことを示しています。
1307	0x0000051B	ERROR_INVALID_OWNER	このセキュリティIDは、このオブジェクトの所有者として割り当てられていない可能性があります。
1308	0x0000051C	ERROR_INVALID_PRIMARY_GROUP	このセキュリティIDは、オブジェクトのプライマリグループとして割り当てられていない可能性があります。
1309	0x0000051D	ERROR_NO_IMPERSONATION_TOKEN	現在クライアントに偽装していないスレッドによって、偽装トークンを操作しようとしてしました。
1310	0x0000051E	ERROR_CANT_DISABLE_MANDATORY	グループが無効になっていない可能性があります。
1311	0x0000051F	ERROR_NO_LOGON_SERVERS	ログオンリクエストに対応する現在使用できるログオンサーバがありません。
1312	0x00000520	ERROR_NO_SUCH_LOGON_SESSION	指定されたログオンセッションは存在しません。既に終了している可能性があります。
1313	0x00000521	ERROR_NO_SUCH_PRIVILEGE	指定された権限は存在しません。
1314	0x00000522	ERROR_PRIVILEGE_NOT_HELD	クライアントは要求された権限を持っていません。
1315	0x00000523	ERROR_INVALID_ACCOUNT_NAME	提供された名前は適切に構成されたアカウント名ではありません。
1316	0x00000524	ERROR_USER_EXISTS	指定されたユーザは既に存在しています。
1317	0x00000525	ERROR_NO_SUCH_USER	指定されたユーザは存在しません。

エラー			説明
10進数	16進数	名前	
1318	0x00000526	ERROR_GROUP_EXISTS	指定されたグループは既に存在しています。
1319	0x00000527	ERROR_NO_SUCH_GROUP	指定されたグループは存在しません。
1320	0x00000528	ERROR_MEMBER_IN_GROUP	指定されたユーザアカウントが既に指定されたグループのメンバであるか、または指定されたグループがメンバを含んでいるため削除できません。
1321	0x00000529	ERROR_MEMBER_NOT_IN_GROUP	指定されたユーザアカウントは、指定されたグループアカウントのメンバではありません。
1322	0x0000052A	ERROR_LAST_ADMIN	最後に残った管理アカウントは無効化または削除できません。
1323	0x0000052B	ERROR_WRONG_PASSWORD	パスワードを更新できません。現在のパスワードとして提供された値が正しくありません。
1324	0x0000052C	ERROR_ILL_FORMED_PASSWORD	パスワードを更新できません。新しいパスワードとして提供された値にパスワードに使用できない値が含まれています。
1325	0x0000052D	ERROR_PASSWORD_RESTRICTION	パスワードを更新できません。新しいパスワードとして提供された値が、ドメインの長さ、複雑さ、または履歴要件を満たしていません。
1326	0x0000052E	ERROR_LOGON_FAILURE	ログオンエラー： ユーザ名を認識できないか、またはパスワードが間違っています。
1327	0x0000052F	ERROR_ACCOUNT_RESTRICTION	ログオンエラー： ユーザアカウントの制限。
1328	0x00000530	ERROR_INVALID_LOGON_HOURS	ログオンエラー： アカウントログオン時間制限の違反。
1329	0x00000531	ERROR_INVALID_WORKSTATION	ログオンエラー： ユーザはこのコンピュータへのログオンが許可されていません。
1330	0x00000532	ERROR_PASSWORD_EXPIRED	ログオンエラー： 指定されたアカウントパスワードの期限が過ぎています。
1331	0x00000533	ERROR_ACCOUNT_DISABLED	ログオンエラー： アカウントは現在無効になっています。
1332	0x00000534	ERROR_NONE_MAPPED	アカウント名とセキュリティIDのマッピングが完了していません。
1333	0x00000535	ERROR_TOO_MANY_LUIDS_REQUESTED	一度に要求したローカルユーザID (LUID) が多すぎます。
1334	0x00000536	ERROR_LUIDS_EXHAUSTED	ローカルユーザID (LUID) はこれ以上使用できません。
1335	0x00000537	ERROR_INVALID_SUB_AUTHORITY	セキュリティIDのサブオーソリティ部分が、この特定の使用に対して無効です。
1336	0x00000538	ERROR_INVALID_ACL	アクセス制御リスト (ACL) 構造体が無効です。
1337	0x00000539	ERROR_INVALID_SID	セキュリティID構造体が無効です。
1338	0x0000053A	ERROR_INVALID_SECURITY_DESCR	セキュリティ記述子構造体が無効です。
1340	0x0000053C	ERROR_BAD_INHERITANCE_ACL	継承したアクセス制御リスト (ACL) またはアクセス制御エントリ (ACE) を構築できませんでした。
1341	0x0000053D	ERROR_SERVER_DISABLED	サーバは現在無効になっています。
1342	0x0000053E	ERROR_SERVER_NOT_DISABLED	サーバは現在有効になっています。
1343	0x0000053F	ERROR_INVALID_ID_AUTHORITY	提供された値は、ID発行機関に対して無効な値です。
1344	0x00000540	ERROR_ALLOTTED_SPACE_EXCEEDED	セキュリティ情報の更新にメモリはこれ以上使用できません。
1345	0x00000541	ERROR_INVALID_GROUP_ATTRIBUTES	指定された属性は無効であるか、全体としてグループの属性との互換性がありません。
1346	0x00000542	ERROR_BAD_IMPERSONATION_LEVEL	必要な偽装レベルが提供されていないか、または提供された偽装レベルが無効です。
1347	0x00000543	ERROR_CANT_OPEN_ANONYMOUS	匿名レベルのセキュリティトークンを開くことができません。
1348	0x00000544	ERROR_BAD_VALIDATION_CLASS	要求された検証情報クラスが無効です。
1349	0x00000545	ERROR_BAD_TOKEN_TYPE	トークンのタイプが用途に適していません。
1350	0x00000546	ERROR_NO_SECURITY_ON_OBJECT	セキュリティが関連付けられていないオブジェクトでのセキュリティ操作を実行できません。
1351	0x00000547	ERROR_CANT_ACCESS_DOMAIN_INFO	マシンが使用不能であるか、またはアクセスが拒否されたため、構成情報をドメインコントローラから読み出すことができませんでした。

エラー			説明
10進数	16進数	名前	
1352	0x00000548	ERROR_INVALID_SERVER_STATE	セキュリティアカウントマネージャ (SAM) またはローカルセキュリティオーソリティ (LSA) サーバが、セキュリティ操作を実行する適切な状態ではありませんでした。
1353	0x00000549	ERROR_INVALID_DOMAIN_STATE	ドメインがセキュリティ操作を実行する適切な状態ではありませんでした。
1354	0x0000054A	ERROR_INVALID_DOMAIN_ROLE	この操作は、ドメインのプライマドメインコントローラの対してのみ許可されています。
1355	0x0000054B	ERROR_NO_SUCH_DOMAIN	指定されたドメインは存在しないか、または接続できません。
1356	0x0000054C	ERROR_DOMAIN_EXISTS	指定されたドメインは既に存在しています。
1357	0x0000054D	ERROR_DOMAIN_LIMIT_EXCEEDED	サーバ1つあたりのドメイン数の制限を超えようとしてしました。
1358	0x0000054E	ERROR_INTERNAL_DB_CORRUPTION	致命的なメディア障害またはディスク上のデータ構造体の破損のため、要求された操作を完了できません。
1359	0x0000054F	ERROR_INTERNAL_ERROR	内部エラーが発生しました。
1360	0x00000550	ERROR_GENERIC_NOT_MAPPED	非汎用タイプが既にマッピングされているアクセスマスクに汎用アクセスタイプが含まれていました。
1361	0x00000551	ERROR_BAD_DESCRIPTOR_FORMAT	セキュリティ記述子が、正しい形式 (絶対または自己相対形式) ではありません。
1362	0x00000552	ERROR_NOT_LOGON_PROCESS	要求されたアクションは、ログオンプロセスによる使用のみに制限されています。この呼び出しプロセスは、ログオンプロセスとして登録されていません。
1363	0x00000553	ERROR_LOGON_SESSION_EXISTS	新しいログオンセッションは既に使用中のIDでは開始できません。
1364	0x00000554	ERROR_NO_SUCH_PACKAGE	指定された認証パッケージが不明です。
1365	0x00000555	ERROR_BAD_LOGON_SESSION_STATE	ログオンセッションが要求された操作に適した状態ではありません。
1366	0x00000556	ERROR_LOGON_SESSION_COLLISION	ログオンセッションIDは既に使用されています。
1367	0x00000557	ERROR_INVALID_LOGON_TYPE	ログオンリクエストに無効なログオンタイプの値が含まれています。
1368	0x00000558	ERROR_CANNOT_IMPERSONATE	そのパイプからデータが読み出されるまでは名前付きパイプを使用してなりません。
1369	0x00000559	ERROR_RXACT_INVALID_STATE	レジストリサブツリーのトランザクション状態が、要求された操作と一致しません。
1370	0x0000055A	ERROR_RXACT_COMMIT_FAILURE	内部セキュリティデータベースの破損が発生しました。
1371	0x0000055B	ERROR_SPECIAL_ACCOUNT	ビルトインアカウントでこの操作を実行することはできません。
1372	0x0000055C	ERROR_SPECIAL_GROUP	このビルトイン特別グループでこの操作を実行することはできません。
1373	0x0000055D	ERROR_SPECIAL_USER	このビルトイン特別ユーザでこの操作を実行することはできません。
1374	0x0000055E	ERROR_MEMBERS_PRIMARY_GROUP	グループは現在ユーザのプライマリグループであるため、ユーザはグループから削除できません。
1375	0x0000055F	ERROR_TOKEN_ALREADY_IN_USE	このトークンはプライマリトークンとして既に使用されています。
1376	0x00000560	ERROR_NO_SUCH_ALIAS	指定されたローカルグループは存在しません。
1377	0x00000561	ERROR_MEMBER_NOT_IN_ALIAS	指定されたアカウント名は、ローカルグループのメンバではありません。
1378	0x00000562	ERROR_MEMBER_IN_ALIAS	指定されたアカウント名は、既にローカルグループのメンバです。
1379	0x00000563	ERROR_ALIAS_EXISTS	指定されたローカルグループは既に存在しています。
1380	0x00000564	ERROR_LOGON_NOT_GRANTED	ログオンエラー: ユーザはこのコンピュータでは要求されるログオンタイプを付与されていません。
1381	0x00000565	ERROR_TOO_MANY_SECRETS	1つのシステム内に保存できる秘密の最大数を超えました。

エラー			説明
10進数	16進数	名前	
1382	0x00000566	ERROR_SECRET_TOO_LONG	秘密の長さが許容される最大長を超えています。
1383	0x00000567	ERROR_INTERNAL_DB_ERROR	ローカルセキュリティオーソリティデータベースに内部的な不整合が含まれています。
1384	0x00000568	ERROR_TOO_MANY_CONTEXT_IDS	ログオンの試行中、ユーザのセキュリティコンテキストに蓄積されたセキュリティIDが多すぎます。
1385	0x00000569	ERROR_LOGON_TYPE_NOT_GRANTED	ログオンエラー: ユーザはこのコンピュータでは要求されるログオンタイプを付与されていません。
1386	0x0000056A	ERROR_NT_CROSS_ENCRYPTION_REQUIRED	ユーザパスワードを変更するには、クロス暗号化パスワードが必要です。
1387	0x0000056B	ERROR_NO_SUCH_MEMBER	メンバが存在しないため、新しいメンバをローカルグループに追加または削除することができませんでした。
1388	0x0000056C	ERROR_INVALID_MEMBER	メンバが間違ったアカウントタイプを持っているため、新しいメンバをローカルグループに追加することができませんでした。
1389	0x0000056D	ERROR_TOO_MANY_SIDS	指定されているセキュリティIDが多すぎます。
1390	0x0000056E	ERROR_LM_CROSS_ENCRYPTION_REQUIRED	このユーザパスワードを変更するには、クロス暗号化パスワードが必要です。
1391	0x0000056F	ERROR_NO_INHERITANCE	ACLが継承可能なコンポーネントを含んでいないことを示します。
1392	0x00000570	ERROR_FILE_CORRUPT	ファイルまたはディレクトリが破損し、読み取ることができません。
1393	0x00000571	ERROR_DISK_CORRUPT	ディスク構造が破損し、読み取ることができません。
1394	0x00000572	ERROR_NO_USER_SESSION_KEY	指定されたログオンセッションに対するユーザセッションキーがありません。
1395	0x00000573	ERROR_LICENSE_QUOTA_EXCEEDED	アクセス中のサービスは、特定の接続数に対してライセンスされています。サービスが許容可能な接続数に既に達しているため、現時点でこれ以上多くの接続をこのサービスに対して行うことはできません。
1396	0x00000574	ERROR_WRONG_TARGET_NAME	ログオンエラー: ターゲットアカウント名が正しくありません。
1397	0x00000575	ERROR_MUTUAL_AUTH_FAILED	相互認証に失敗しました。サーバのパスワードがドメインコントローラで期限が過ぎています。
1398	0x00000576	ERROR_TIME_SKEW	クライアントとサーバ間で時間差があります。
1399	0x00000577	ERROR_CURRENT_DOMAIN_NOT_ALLOWED	この操作は現在のドメインでは実行できません。

エラー			説明
10進数	16進数	名前	
1400	0x00000578	ERROR_INVALID_WINDOW_HANDLE	無効なウィンドウハンドル
1401	0x00000579	ERROR_INVALID_MENU_HANDLE	無効なメニューハンドル
1402	0x0000057A	ERROR_INVALID_CURSOR_HANDLE	無効なカーソルハンドル
1403	0x0000057B	ERROR_INVALID_ACCEL_HANDLE	無効なアクセラレータテーブルハンドル
1404	0x0000057C	ERROR_INVALID_HOOK_HANDLE	無効なフックハンドル
1405	0x0000057D	ERROR_INVALID_DWP_HANDLE	複数のウィンドウ位置構造に対する無効なハンドル。
1406	0x0000057E	ERROR_TLW_WITH_WSCHILD	トップレベルの子ウィンドウを作成できません。
1407	0x0000057F	ERROR_CANNOT_FIND_WND_CLASS	ウィンドウクラスを見つけることができません。
1408	0x00000580	ERROR_WINDOW_OF_OTHER_THREAD	無効なウィンドウです。他のスレッドに属しています。
1409	0x00000581	ERROR_HOTKEY_ALREADY_REGISTERED	ホットキーは既に登録されています。
1410	0x00000582	ERROR_CLASS_ALREADY_EXISTS	クラスは既に存在しています。
1411	0x00000583	ERROR_CLASS_DOES_NOT_EXIST	クラスが存在しません。
1412	0x00000584	ERROR_CLASS_HAS_WINDOWS	クラスはまだウィンドウで開かれています。
1413	0x00000585	ERROR_INVALID_INDEX	無効なインデックス
1414	0x00000586	ERROR_INVALID_ICON_HANDLE	無効なアイコンハンドル
1415	0x00000587	ERROR_PRIVATE_DIALOG_INDEX	プライベートのDIALOGウィンドウワードを使用。
1416	0x00000588	ERROR_LISTBOX_ID_NOT_FOUND	リストボックスIDが見つかりませんでした。

		エラー		説明
10進数	16進数	名前		
1417	0x00000589	ERROR_NO_WILDCARD_CHARACTERS		ワイルドカードが見つかりませんでした。
1418	0x0000058A	ERROR_CLIPBOARD_NOT_OPEN		スレッドはクリップボードを開いていません。
1419	0x0000058B	ERROR_HOTKEY_NOT_REGISTERED		ホットキーは登録されていません。
1420	0x0000058C	ERROR_WINDOW_NOT_DIALOG		ウィンドウは有効なダイアログウィンドウではありません。
1421	0x0000058D	ERROR_CONTROL_ID_NOT_FOUND		制御IDが見つかりませんでした。
1422	0x0000058E	ERROR_INVALID_COMBOBOX_MESSAGE		コンボボックスは編集制御を持っていないため無効なメッセージです。
1423	0x0000058F	ERROR_WINDOW_NOT_COMBOBOX		このウィンドウはコンボボックスではありません。
1424	0x00000590	ERROR_INVALID_EDIT_HEIGHT		高さは256未満である必要があります。
1425	0x00000591	ERROR_DC_NOT_FOUND		無効なデバイスコンテキスト (DC) ハンドル。
1426	0x00000592	ERROR_INVALID_HOOK_FILTER		無効なフックプロシージャタイプ。
1427	0x00000593	ERROR_INVALID_FILTER_PROC		無効なフックプロシージャ。
1428	0x00000594	ERROR_HOOK_NEEDS_HMOD		モジュールハンドルのない非ローカルフックを設定できません。
1429	0x00000595	ERROR_GLOBAL_ONLY_HOOK		このフックプロシージャグローバルでのみ設定できます。
1430	0x00000596	ERROR_JOURNAL_HOOK_SET		ジャーナルフックプロシージャは既にインストールされています。
1431	0x00000597	ERROR_HOOK_NOT_INSTALLED		フックプロシージャはインストールされていません。
1432	0x00000598	ERROR_INVALID_LB_MESSAGE		単一選択のリストボックスに対する無効なメッセージ。
1433	0x00000599	ERROR_SETCOUNT_ON_BAD_LB		LB_SETCOUNTが非レイジー型リストボックスに送信されました。
1434	0x0000059A	ERROR_LB_WITHOUT_TABSTOPS		このリストボックスはタブストップをサポートしていません。
1435	0x0000059B	ERROR_DESTROY_OBJECT_OF_OTHER_THREAD		他のスレッドによって作成されたオブジェクトを削除できません。
1436	0x0000059C	ERROR_CHILD_WINDOW_MENU		子ウィンドウはメニューを持つことができません。
1437	0x0000059D	ERROR_NO_SYSTEM_MENU		このウィンドウにはシステムメニューがありません。
1438	0x0000059E	ERROR_INVALID_MSGBOX_STYLE		無効なメッセージボックスのスタイル。
1439	0x0000059F	ERROR_INVALID_SPI_VALUE		無効なシステム幅 (SPI_*) パラメータ。
1440	0x000005A0	ERROR_SCREEN_ALREADY_LOCKED		スクリーンは既にロックされています。
1441	0x000005A1	ERROR_HWNDS_HAVE_DIFF_PARENT		複数のウィンドウ位置構造ではウィンドウに対するすべてのハンドルが同じ親を持つ必要があります。
1442	0x000005A2	ERROR_NOT_CHILD_WINDOW		このウィンドウは子ウィンドウではありません。
1443	0x000005A3	ERROR_INVALID_GW_COMMAND		無効なGW_* コマンド
1444	0x000005A4	ERROR_INVALID_THREAD_ID		無効なスレッドID
1445	0x000005A5	ERROR_NON_MDICHILD_WINDOW		マルチドキュメントインターフェイス (MDI) ウィンドウではないウィンドウからのメッセージは処理できません。
1446	0x000005A6	ERROR_POPUP_ALREADY_ACTIVE		ポップアップメニューは既にアクティブです。
1447	0x000005A7	ERROR_NO_SCROLLBARS		このウィンドウにはスクロールバーがありません。
1448	0x000005A8	ERROR_INVALID_SCROLLBAR_RANGE		スクロールバーの範囲は、MAXLONGより大きくできません。
1449	0x000005A9	ERROR_INVALID_SHOWWIN_COMMAND		指定された方法ではウィンドウを表示または削除できません。
1450	0x000005AA	ERROR_NO_SYSTEM_RESOURCES		要求されたサービスを完了するには存在するシステムリソースが不十分です。
1451	0x000005AB	ERROR_NONPAGED_SYSTEM_RESOURCES		要求されたサービスを完了するには存在するシステムリソースが不十分です。
1452	0x000005AC	ERROR_PAGED_SYSTEM_RESOURCES		要求されたサービスを完了するには存在するシステムリソースが不十分です。
1453	0x000005AD	ERROR_WORKING_SET_QUOTA		要求されたサービスを完了するには割り当てが不十分です。

エラー			説明
10進数	16進数	名前	
1454	0x000005AE	ERROR_PAGEFILE_QUOTA	要求されたサービスを完了するには割り当てが不十分です。
1455	0x000005AF	ERROR_COMMITMENT_LIMIT	この操作を完了するにはページファイルが小さすぎます。
1456	0x000005B0	ERROR_MENU_ITEM_NOT_FOUND	メニューアイテムが見つかりませんでした。
1457	0x000005B1	ERROR_INVALID_KEYBOARD_HANDLE	無効なキーボードレイアウトハンドル。
1458	0x000005B2	ERROR_HOOK_TYPE_NOT_ALLOWED	フックタイプが許可されていません。
1459	0x000005B3	ERROR_REQUIRES_INTERACTIVE_WINDOWSTATION	この操作には対話的なウィンドウステーションが必要です。
1460	0x000005B4	ERROR_TIMEOUT	タイムアウト期限が過ぎたため、この操作は返されました。
1461	0x000005B5	ERROR_INVALID_MONITOR_HANDLE	無効なモニタハンドル
1500	0x000005DC	ERROR_EVENTLOG_FILE_CORRUPT	イベントログファイルが破損しています。
1501	0x000005DD	ERROR_EVENTLOG_CANT_START	イベントログファイルを開くことができないため、イベントログサービスが開始されませんでした。
1502	0x000005DE	ERROR_LOG_FILE_FULL	イベントログファイルの空き領域がありません。
1503	0x000005DF	ERROR_EVENTLOG_FILE_CHANGED	イベントログファイルが読み取り操作の間に変更されています。

エラー			説明
10進数	16進数	名前	
1601	0x00000641	ERROR_INSTALL_SERVICE_FAILURE	Windows Installerサービスにアクセスできませんでした。サポート担当者に連絡し、Windows Installerサービスが適切に登録されていることを確認してください。
1602	0x00000642	ERROR_INSTALL_USEREXIT	ユーザがインストールをキャンセルしました。
1603	0x00000643	ERROR_INSTALL_FAILURE	インストール中の致命的なエラー。
1604	0x00000644	ERROR_INSTALL_SUSPEND	インストールが中止され、完了していません。
1605	0x00000645	ERROR_UNKNOWN_PRODUCT	この操作は、現在インストールされている製品に対してのみ有効です。
1606	0x00000646	ERROR_UNKNOWN_FEATURE	機能IDが登録されていません。
1607	0x00000647	ERROR_UNKNOWN_COMPONENT	コンポーネントIDが登録されていません。
1608	0x00000648	ERROR_UNKNOWN_PROPERTY	不明なプロパティ。
1609	0x00000649	ERROR_INVALID_HANDLE_STATE	ハンドルが無効な状態です
1610	0x0000064A	ERROR_BAD_CONFIGURATION	この製品の設定データが破損しています。サポート担当者にお問い合わせください。
1611	0x0000064B	ERROR_INDEX_ABSENT	コンポーネント修飾子がありません。
1612	0x0000064C	ERROR_INSTALL_SOURCE_ABSENT	この製品のインストールソースが使用できません。ソースが存在し、それにアクセスできることを確認してください。
1613	0x0000064D	ERROR_INSTALL_PACKAGE_VERSION	このインストールパッケージは、Windows Installerサービスではインストールすることができません。新しいバージョンのWindows Installerサービスを含むWindowsサービスパックをインストールする必要があります。
1614	0x0000064E	ERROR_PRODUCT_UNINSTALLED	製品がアンインストールされています。
1615	0x0000064F	ERROR_BAD_QUERY_SYNTAX	SQLクエリー構文が無効またはサポートされていません。
1616	0x00000650	ERROR_INVALID_FIELD	レコードフィールドが存在しません。
1617	0x00000651	ERROR_DEVICE_REMOVED	デバイスが削除されています。
1618	0x00000652	ERROR_INSTALL_ALREADY_RUNNING	別のインストールが既に進行中です。このインストールを開始する前に別のインストールを完了してください。
1619	0x00000653	ERROR_INSTALL_PACKAGE_OPEN_FAILED	このインストールパッケージを開くことができませんでした。パッケージが存在し、それにアクセスできることを確認するか、またはアプリケーションベンダに連絡し、これが有効なWindows Installerパッケージであることを確認してください。

エラー			説明
10進数	16進数	名前	
1620	0x00000654	ERROR_INSTALL_PACKAGE_INVALID	このインストールパッケージを開くことができませんでした。アプリケーションベンダに連絡し、これが有効なWindows Installerパッケージであることを確認してください。
1621	0x00000655	ERROR_INSTALL_UI_FAILURE	Windows Installerサービスのユーザインターフェイス開始時にエラーが発生しました。サポート担当者にお問い合わせください。
1622	0x00000656	ERROR_INSTALL_LOG_FAILURE	インストールログファイルを開くときにエラーが発生しました。指定されたログファイルの場所が存在し、それに書き込むことができることを確認してください。
1623	0x00000657	ERROR_INSTALL_LANGUAGE_UNSUPPORTED	このインストールパッケージの言語がシステムでサポートされていません。
1624	0x00000658	ERROR_INSTALL_TRANSFORM_FAILURE	変換を適用するときにエラーが発生しました。指定された変換パスが有効であることを確認してください。
1625	0x00000659	ERROR_INSTALL_PACKAGE_REJECTED	このインストールは、システムポリシーによって禁止されています。システム管理者にお問い合わせください。
1626	0x0000065A	ERROR_FUNCTION_NOT_CALLED	ファンクションが実行できませんでした。
1627	0x0000065B	ERROR_FUNCTION_FAILED	ファンクションが実行中に失敗しました。
1628	0x0000065C	ERROR_INVALID_TABLE	無効または不明なテーブルが指定されています。
1629	0x0000065D	ERROR_DATATYPE_MISMATCH	提供されたデータのタイプが間違っています。
1630	0x0000065E	ERROR_UNSUPPORTED_TYPE	このタイプのデータはサポートされていません。
1631	0x0000065F	ERROR_CREATE_FAILED	Windows Installerサービスが開始できませんでした。サポート担当者にお問い合わせください。
1632	0x00000660	ERROR_INSTALL_TEMP_UNWRITABLE	tempフォルダの空き領域がないか、またはアクセスできません。tempフォルダが存在し、それに書き込むことができることを確認してください。
1633	0x00000661	ERROR_INSTALL_PLATFORM_UNSUPPORTED	このインストールパッケージは、このプロセッサタイプではサポートされていません。製品ベンダにお問い合わせください。
1634	0x00000662	ERROR_INSTALL_NOTUSED	コンポーネントはこのコンピュータで使用されていません。
1635	0x00000663	ERROR_PATCH_PACKAGE_OPEN_FAILED	このパッチパッケージを開くことができませんでした。パッチパッケージが存在し、それにアクセスできることを確認するか、またはアプリケーションベンダに連絡し、これが有効なWindows Installerパッチパッケージであることを確認してください。
1636	0x00000664	ERROR_PATCH_PACKAGE_INVALID	このパッチパッケージを開くことができませんでした。アプリケーションベンダに連絡し、これが有効なWindows Installerパッチパッケージであることを確認してください。
1637	0x00000665	ERROR_PATCH_PACKAGE_UNSUPPORTED.	このパッチパッケージは、Windows Installerサービスでは処理することができません。新しいバージョンのWindows Installerサービスを含むWindowsサービスパックをインストールする必要があります。
1638	0x00000666	ERROR_PRODUCT_VERSION	この製品の別のバージョンが既にインストールされています。このバージョンのインストールは継続できません。この製品の既存のバージョンを設定または削除するには、コントロールパネルで「プログラムの追加/削除」を使用してください。
1639	0x00000667	ERROR_INVALID_COMMAND_LINE	無効なコマンドライン引数。コマンドラインヘルプの詳細については、Windows Installer SDKを参照してください。
1640	0x00000668	ERROR_INSTALL_REMOTE_DISALLOWED	ターミナルサービスのリモートセッションでは、管理者のみがサーバソフトウェアの追加、削除または設定を行う権限があります。サーバにソフトウェアをインストールまたは設定したい場合は、ネットワーク管理者にお問い合わせください。
1641	0x00000669	ERROR_SUCCESS_REBOOT_INITIATED	要求された操作が正常に完了しました。システムが再起動され、変更が反映されます。

		エラー		説明
10進数	16進数	名前		
1642	0x0000066A	ERROR_PATCH_TARGET_NOT_FOUND		アップグレードするプログラムが見つからない、またはアップグレードパッチがプログラムの違うバージョンをアップグレードするためのものであるため、アップグレードパッチはWindows Installerサービスではインストールできません。アップグレードするプログラムがコンピュータに存在し、正しいアップグレードパッチを使用していることを確認してください。
1643	0x0000066B	ERROR_PATCH_PACKAGE_REJECTED		このパッチパッケージはシステムポリシーでは許可されていません。パッケージが適切な証明書によって署名されていません。
1644	0x0000066C	ERROR_INSTALL_TRANSFORM_REJECTED		1つ以上のカスタマイズはシステムポリシーでは許可されていません。カスタマイズが適切な証明書によって署名されていません。
1700	0x000006A4	RPC_S_INVALID_STRING_BINDING		文字列のバインドが無効です。
1701	0x000006A5	RPC_S_WRONG_KIND_OF_BINDING		バインディングハンドルが正しいタイプではありません。
1702	0x000006A6	RPC_S_INVALID_BINDING		バインディングハンドルが無効です。
1703	0x000006A7	RPC_S_PROTSEQ_NOT_SUPPORTED		RPCプロトコルシーケンスがサポートされていません。
1704	0x000006A8	RPC_S_INVALID_RPC_PROTSEQ		RPCプロトコルシーケンスが無効です。
1705	0x000006A9	RPC_S_INVALID_STRING_UUID		文字列のUniversal Unique Identifier (UUID)が無効です。
1706	0x000006AA	RPC_S_INVALID_ENDPOINT_FORMAT		エンドポイント形式が無効です。
1707	0x000006AB	RPC_S_INVALID_NET_ADDR		ネットワークアドレスが無効です。
1708	0x000006AC	RPC_S_NO_ENDPOINT_FOUND		エンドポイントが見つかりませんでした。
1709	0x000006AD	RPC_S_INVALID_TIMEOUT		タイムアウト値が無効です。
1710	0x000006AE	RPC_S_OBJECT_NOT_FOUND		オブジェクトのUniversal Unique Identifier (UUID)が見つかりませんでした。
1711	0x000006AF	RPC_S_ALREADY_REGISTERED		オブジェクトのUniversal Unique Identifier (UUID)が既に登録されています。
1712	0x000006B0	RPC_S_TYPE_ALREADY_REGISTERED		タイプのUniversal Unique Identifier (UUID)が既に登録されています。
1713	0x000006B1	RPC_S_ALREADY_LISTENING		RPCサーバは既に稼動しています。
1714	0x000006B2	RPC_S_NO_PROTSEQS_REGISTERED		プロトコルシーケンスが登録されていません。
1715	0x000006B3	RPC_S_NOT_LISTENING		RPCサーバが稼動していません。
1716	0x000006B4	RPC_S_UNKNOWN_MGR_TYPE		マネージャのタイプが不明です。
1717	0x000006B5	RPC_S_UNKNOWN_IF		インターフェイスが不明です。
1718	0x000006B6	RPC_S_NO_BINDINGS		バインディングがありません。
1719	0x000006B7	RPC_S_NO_PROTSEQS		プロトコルシーケンスがありません。
1720	0x000006B8	RPC_S_CANT_CREATE_ENDPOINT		エンドポイントを作成できません。
1721	0x000006B9	RPC_S_OUT_OF_RESOURCES		この操作を完了するために十分なリソースを使用できません。
1722	0x000006BA	RPC_S_SERVER_UNAVAILABLE		RPCサーバが使用できません。
1723	0x000006BB	RPC_S_SERVER_TOO_BUSY		この操作を完了するにはRPCサーバが過度にビジーです。
1724	0x000006BC	RPC_S_INVALID_NETWORK_OPTIONS		ネットワークオプションが無効です。
1725	0x000006BD	RPC_S_NO_CALL_ACTIVE		このスレッドでアクティブなリモートプロシージャの呼び出しがありません。
1726	0x000006BE	RPC_S_CALL_FAILED		リモートプロシージャの呼び出しに失敗しました。
1727	0x000006BF	RPC_S_CALL_FAILED_DNE		リモートプロシージャの呼び出しに失敗し、実行できませんでした。
1728	0x000006C0	RPC_S_PROTOCOL_ERROR		リモートプロシージャの呼び出し (RPC) プロトコルにエラーが発生しました。
1730	0x000006C2	RPC_S_UNSUPPORTED_TRANS_SYN		この転送構文は、RPCサーバではサポートされていません。
1732	0x000006C4	RPC_S_UNSUPPORTED_TYPE		Universal Unique Identifier (UUID) のタイプがサポートされていません。
1733	0x000006C5	RPC_S_INVALID_TAG		タグが無効です。
1734	0x000006C6	RPC_S_INVALID_BOUND		配列の境界が無効です。

		エラー	説明
10進数	16進数	名前	
1735	0x000006C7	RPC_S_NO_ENTRY_NAME	バインディングにエントリ名が含まれていません。
1736	0x000006C8	RPC_S_INVALID_NAME_SYNTAX	名前の構文が無効です。
1737	0x000006C9	RPC_S_UNSUPPORTED_NAME_SYNTAX	名前の構文がサポートされていません。
1739	0x000006CB	RPC_S_UUID_NO_ADDRESS	Universal Unique Identifier (UUID) を構築するためにネットワークアドレスが使用できません。
1740	0x000006CC	RPC_S_DUPLICATE_ENDPOINT	エンドポイントが重複しています。
1741	0x000006CD	RPC_S_UNKNOWN_AUTHN_TYPE	認証タイプが不明です。
1742	0x000006CE	RPC_S_MAX_CALLS_TOO_SMALL	呼び出しの最大数が小さすぎます。
1743	0x000006CF	RPC_S_STRING_TOO_LONG	文字列が長すぎます。
1744	0x000006D0	RPC_S_PROTSEQ_NOT_FOUND	RPCプロトコルシーケンスが見つかりませんでした。
1745	0x000006D1	RPC_S_PROCNUM_OUT_OF_RANGE	プロシージャ番号が範囲外です。
1746	0x000006D2	RPC_S_BINDING_HAS_NO_AUTH	バインディングに認証情報が含まれていません。
1747	0x000006D3	RPC_S_UNKNOWN_AUTHN_SERVICE	認証サービスが不明です。
1748	0x000006D4	RPC_S_UNKNOWN_AUTHN_LEVEL	認証レベルが不明です。
1749	0x000006D5	RPC_S_INVALID_AUTH_IDENTITY	セキュリティコンテキストが無効です。
1750	0x000006D6	RPC_S_UNKNOWN_AUTHZ_SERVICE	認証サービスが不明です。
1751	0x000006D7	EPT_S_INVALID_ENTRY	エントリが無効です。
1752	0x000006D8	EPT_S_CANT_PERFORM_OP	サーバエンドポイントが操作を実行できません。
1753	0x000006D9	EPT_S_NOT_REGISTERED	エンドポイントマッパーからエンドポイントはこれ以上使用できません。
1754	0x000006DA	RPC_S_NOTHING_TO_EXPORT	インターフェイスがエクスポートされていません。
1755	0x000006DB	RPC_S_INCOMPLETE_NAME	エントリ名が不完全です。
1756	0x000006DC	RPC_S_INVALID_VERS_OPTION	バージョンオプションが無効です。
1757	0x000006DD	RPC_S_NO_MORE_MEMBERS	メンバーはこれ以上ありません。
1758	0x000006DE	RPC_S_NOT_ALL_OBJS_UNEXPORTED	アンエクスポートするものは何もありません。
1759	0x000006DF	RPC_S_INTERFACE_NOT_FOUND	インターフェイスが見つかりませんでした。
1760	0x000006E0	RPC_S_ENTRY_ALREADY_EXISTS	エントリは既に存在しています。
1761	0x000006E1	RPC_S_ENTRY_NOT_FOUND	エントリが見つかりませんでした。
1762	0x000006E2	RPC_S_NAME_SERVICE_UNAVAILABLE	名前サービスが使用できません。
1763	0x000006E3	RPC_S_INVALID_NAF_ID	ネットワークアドレスファミリが無効です。
1764	0x000006E4	RPC_S_CANNOT_SUPPORT	要求された操作がサポートされていません。
1765	0x000006E5	RPC_S_NO_CONTEXT_AVAILABLE	偽装を許可するためにセキュリティコンテキストが使用できません。
1766	0x000006E6	RPC_S_INTERNAL_ERROR	リモートプロシージャの呼び出し (RPC) で内部エラーが発生しました。
1767	0x000006E7	RPC_S_ZERO_DIVIDE	RPCサーバが整数のゼロ除算を実行しようとしてしました。
1768	0x000006E8	RPC_S_ADDRESS_ERROR	RPCサーバでアドレスエラーが発生しました。
1769	0x000006E9	RPC_S_FP_DIV_ZERO	RPCサーバでの浮動小数点数操作がゼロ除算を発生させました。
1770	0x000006EA	RPC_S_FP_UNDERFLOW	RPCサーバで浮動小数点数のアンダーフローが発生しました。
1771	0x000006EB	RPC_S_FP_OVERFLOW	RPCサーバで浮動小数点数のオーバーフローが発生しました。
1772	0x000006EC	RPC_X_NO_MORE_ENTRIES	自動ハンドルのバインディングに使用可能なRPCサーバのリストがすべて使われています。
1773	0x000006ED	RPC_X_SS_CHAR_TRANS_OPEN_FAIL	文字変換テーブルファイルを開くことができません。
1774	0x000006EE	RPC_X_SS_CHAR_TRANS_SHORT_FILE	文字変換テーブルを含むファイルが512バイト未満です。
1775	0x000006EF	RPC_X_SS_IN_NULL_CONTEXT	リモートプロシージャ呼び出し中にヌルコンテキストハンドルがクライアントからホストに渡されました。
1777	0x000006F1	RPC_X_SS_CONTEXT_DAMAGED	リモートプロシージャ呼び出し中にコンテキストハンドルが変更されました。

エラー			説明
10進数	16進数	名前	
1778	0x000006F2	RPC_X_SS_HANDLES_MISMATCH	リモートプロシージャ呼び出しに渡されたバインディングハンドルが一致しません。
1779	0x000006F3	RPC_X_SS_CANNOT_GET_CALL_HANDLE	スタブがリモートプロシージャ呼び出しハンドルを取得できません。
1780	0x000006F4	RPC_X_NULL_REF_POINTER	ヌル参照ポインタがスタブに渡されました。
1781	0x000006F5	RPC_X_ENUM_VALUE_OUT_OF_RANGE	列挙値が範囲外です。
1782	0x000006F6	RPC_X_BYTE_COUNT_TOO_SMALL	バイトカウントが小さすぎます。
1783	0x000006F7	RPC_X_BAD_STUB_DATA	スタブが無効なデータを受信しました。
1784	0x000006F8	ERROR_INVALID_USER_BUFFER	提供されたユーザバッファが要求された操作に対して有効ではありません。
1785	0x000006F9	ERROR_UNRECOGNIZED_MEDIA	ディスクメディアが認識されません。初期化されていない可能性があります。
1786	0x000006FA	ERROR_NO_TRUST_LSA_SECRET	ワークステーションに信頼できる秘密がありません。
1787	0x000006FB	ERROR_NO_TRUST_SAM_ACCOUNT	サーバ上のセキュリティデータベースにこのワークステーションの信頼できる関係を示すコンピュータアカウントがありません。
1788	0x000006FC	ERROR_TRUSTED_DOMAIN_FAILURE	プライマリドメインと信頼できるドメインの信頼関係に失敗しました。
1789	0x000006FD	ERROR_TRUSTED_RELATIONSHIP_FAILURE	ワークステーションとプライマリドメインの信頼関係に失敗しました。
1790	0x000006FE	ERROR_TRUST_FAILURE	ネットワークのログオンに失敗しました。
1791	0x000006FF	RPC_S_CALL_IN_PROGRESS	リモートプロシージャ呼び出しはこのスレッドで既に進行中です。
1792	0x00000700	ERROR_NETLOGON_NOT_STARTED	ログオンしようとしたのですが、ネットワークログオンサービスが開始していませんでした。
1793	0x00000701	ERROR_ACCOUNT_EXPIRED	ユーザのアカウントの期限が過ぎています。
1794	0x00000702	ERROR_REDIRECTOR_HAS_OPEN_HANDLES	リダイレクタが使用中でアンロードできません。
1795	0x00000703	ERROR_PRINTER_DRIVER_ALREADY_INSTALLED	指定されたプリンタドライバは既にインストールされています。
1796	0x00000704	ERROR_UNKNOWN_PORT	指定されたポートが不明です。
1797	0x00000705	ERROR_UNKNOWN_PRINTER_DRIVER	プリンタドライバが不明です。
1798	0x00000706	ERROR_UNKNOWN_PRINTPROCESSOR	印刷プロセッサが不明です。
1799	0x00000707	ERROR_INVALID_SEPARATOR_FILE	指定されたセパレータファイルが無効です。

エラー			説明
10進数	16進数	名前	
1800	0x00000708	ERROR_INVALID_PRIORITY	指定されたプライオリティが無効です。
1801	0x00000709	ERROR_INVALID_PRINTER_NAME	プリンタ名が無効です。
1802	0x0000070A	ERROR_PRINTER_ALREADY_EXISTS	プリンタは既に存在しています。
1803	0x0000070B	ERROR_INVALID_PRINTER_COMMAND	プリンタコマンドが無効です。
1804	0x0000070C	ERROR_INVALID_DATATYPE	指定されたデータタイプが無効です。
1805	0x0000070D	ERROR_INVALID_ENVIRONMENT	指定された環境が無効です。
1806	0x0000070E	RPC_S_NO_MORE_BINDINGS	バインディングはこれ以上ありません。
1807	0x0000070F	ERROR_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT	使用されたアカウントがドメイン間のトラストアカウントです。グローバルユーザアカウントまたはローカルユーザアカウントを使用してこのサーバにアクセスしてください。
1808	0x00000710	ERROR_NOLOGON_WORKSTATION_TRUST_ACCOUNT	使用されたアカウントはコンピュータアカウントです。グローバルユーザアカウントまたはローカルユーザアカウントを使用してこのサーバにアクセスしてください。
1809	0x00000711	ERROR_NOLOGON_SERVER_TRUST_ACCOUNT	使用されたアカウントはサーバトラストアカウントです。グローバルユーザアカウントまたはローカルユーザアカウントを使用してこのサーバにアクセスしてください。
1810	0x00000712	ERROR_DOMAIN_TRUST_INCONSISTENT	指定されたドメインの名前またはセキュリティID (SID)が、ドメインの信頼情報と一致しません。
1811	0x00000713	ERROR_SERVER_HAS_OPEN_HANDLES	サーバが使用中でアンロードできません。
1812	0x00000714	ERROR_RESOURCE_DATA_NOT_FOUND	指定されたイメージファイルにリソースセクションが含まれていませんでした。

		エラー		説明
10進数	16進数	名前		
1813	0x00000715	ERROR_RESOURCE_TYPE_NOT_FOUND		指定されたリソースタイプがイメージファイルで見つかりませんでした。
1814	0x00000716	ERROR_RESOURCE_NAME_NOT_FOUND		指定されたリソース名がイメージファイルで見つかりませんでした。
1815	0x00000717	ERROR_RESOURCE_LANG_NOT_FOUND		指定されたリソース言語IDがイメージファイルで見つかりませんでした。
1816	0x00000718	ERROR_NOT_ENOUGH_QUOTA		このコマンドを処理するために十分な割り当てを使用できません。
1817	0x00000719	RPC_S_NO_INTERFACES		インターフェイスが登録されていません。
1818	0x0000071A	RPC_S_CALL_CANCELLED		リモートプロシージャ呼び出しがキャンセルされました。
1819	0x0000071B	RPC_S_BINDING_INCOMPLETE		バインディングハンドルに必要なすべての情報が含まれていません。
1820	0x0000071C	RPC_S_COMM_FAILURE		リモートプロシージャ呼び出し中に通信エラーが発生しました。
1821	0x0000071D	RPC_S_UNSUPPORTED_AUTHN_LEVEL		要求された認証レベルはサポートされていません。
1822	0x0000071E	RPC_S_NO_PRINC_NAME		プリンシパル名が登録されていません。
1823	0x0000071F	RPC_S_NOT_RPC_ERROR		指定されたエラーは、有効なWindows RPCエラーコードではありません。
1824	0x00000720	RPC_S_UUID_LOCAL_ONLY		このコンピュータのみで有効なUUIDが割り当てられています。
1825	0x00000721	RPC_S_SEC_PKG_ERROR		セキュリティパッケージ固有のエラーが発生しました。
1826	0x00000722	RPC_S_NOT_CANCELLED		スレッドがキャンセルされていません。
1827	0x00000723	RPC_X_INVALID_ES_ACTION		エンコード/デコードハンドルの無効な操作。
1828	0x00000724	RPC_X_WRONG_ES_VERSION		互換性のないシリアライズパッケージのバージョン。
1829	0x00000725	RPC_X_WRONG_STUB_VERSION		互換性のないRPCスタブのバージョン。
1830	0x00000726	RPC_X_INVALID_PIPE_OBJECT		RPCパイプオブジェクトが無効または破損しています。
1831	0x00000727	RPC_X_WRONG_PIPE_ORDER		RPCパイプオブジェクトに無効な操作を実行しようとした。
1832	0x00000728	RPC_X_WRONG_PIPE_VERSION		サポートされていないRPCパイプバージョン。
1898	0x0000076A	RPC_S_GROUP_MEMBER_NOT_FOUND		グループメンバが見つかりませんでした。
1899	0x0000076B	EPT_S_CANT_CREATE		エンドポイントマッパーのデータベースエントリが作成できませんでした。
1900	0x0000076C	RPC_S_INVALID_OBJECT		オブジェクトのUniversal Unique Identifier (UUID) が空のUUIDです。
1901	0x0000076D	ERROR_INVALID_TIME		指定された時間が無効です。
1902	0x0000076E	ERROR_INVALID_FORM_NAME		指定されたフォーム名が無効です。
1903	0x0000076F	ERROR_INVALID_FORM_SIZE		指定されたフォームサイズが無効です。
1904	0x00000770	ERROR_ALREADY_WAITING		指定されたプリンタハンドルが既に待機中です。
1905	0x00000771	ERROR_PRINTER_DELETED		指定されたプリンタが削除されています。
1906	0x00000772	ERROR_INVALID_PRINTER_STATE		プリンタの状態が無効です。
1907	0x00000773	ERROR_PASSWORD_MUST_CHANGE		ユーザのパスワードは最初にログオンする前に変更する必要があります。
1908	0x00000774	ERROR_DOMAIN_CONTROLLER_NOT_FOUND		このドメインのドメインコントローラが見つかりませんでした。
1909	0x00000775	ERROR_ACCOUNT_LOCKED_OUT		参照されたアカウントは現在ロックされ、ログオンできません。
1910	0x00000776	OR_INVALID_OXID		指定されたオブジェクトエクスポートが見つかりませんでした。
1911	0x00000777	OR_INVALID_OID		指定されたオブジェクトが見つかりませんでした。
1912	0x00000778	OR_INVALID_SET		指定されたオブジェクトレゾルバセットが見つかりませんでした。
1913	0x00000779	RPC_S_SEND_INCOMPLETE		送信するデータの一部がリクエストバッファに残されています。
1914	0x0000077A	RPC_S_INVALID_ASYNC_HANDLE		無効な非同期リモートプロシージャ呼び出しハンドル。

エラー			説明
10進数	16進数	名前	
1915	0x0000077B	RPC_S_INVALID_ASYNC_CALL	この操作に対する無効な非同期RPC呼び出しハンドドル。
1916	0x0000077C	RPC_X_PIPE_CLOSED	RPCパイプオブジェクトは既に閉じています。
1917	0x0000077D	RPC_X_PIPE_DISCIPLINE_ERROR	RPC呼び出しがすべてのパイプが処理される前に完了しました。
1918	0x0000077E	RPC_X_PIPE_EMPTY	RPCパイプからデータはこれ以上使用できません。
1919	0x0000077F	ERROR_NO_SITENAME	このマシンで使用できるサイト名がありません。
1920	0x00000780	ERROR_GANT_ACCESS_FILE	ファイルがシステムによってアクセスできません。
1921	0x00000781	ERROR_GANT_RESOLVE_FILENAME	ファイルの名前がシステムによって解決できません。
1922	0x00000782	RPC_S_ENTRY_TYPE_MISMATCH	エントリが所期のタイプではありません。
1923	0x00000783	RPC_S_NOT_ALL_OBJS_EXPORTED	すべてのオブジェクトUUIDが指定されたエントリにエクスポートできませんでした。
1924	0x00000784	RPC_S_INTERFACE_NOT_EXPORTED	インターフェイスが指定されたエントリにエクスポートできませんでした。
1925	0x00000785	RPC_S_PROFILE_NOT_ADDED	指定されたプロファイルエントリが追加できませんでした。
1926	0x00000786	RPC_S_PRF_ELT_NOT_ADDED	指定されたプロファイルエレメントが追加できませんでした。
1927	0x00000787	RPC_S_PRF_ELT_NOT_REMOVED	指定されたプロファイルエレメントが削除できませんでした。
1928	0x00000788	RPC_S_GRP_ELT_NOT_ADDED	グループエレメントが追加できませんでした。
1929	0x00000789	RPC_S_GRP_ELT_NOT_REMOVED	グループエレメントが削除できませんでした。
1930	0x0000078A	ERROR_KM_DRIVER_BLOCKED	プリンタドライバがコンピュータで有効なポリシーと互換性がなく、NT 4.0ドライバをブロックしています。

エラー			説明
10進数	16進数	名前	
2000	0x000007D0	ERROR_INVALID_PIXEL_FORMAT	ピクセル形式が無効です。
2001	0x000007D1	ERROR_BAD_DRIVER	指定されたドライバが無効です。
2002	0x000007D2	ERROR_INVALID_WINDOW_STYLE	ウィンドウスタイルまたはクラス属性がこの操作に対して無効です。
2003	0x000007D3	ERROR_METAFILE_NOT_SUPPORTED	要求されたメタファイル操作がサポートされていません。
2004	0x000007D4	ERROR_TRANSFORM_NOT_SUPPORTED	要求された変換操作がサポートされていません。
2005	0x000007D5	ERROR_CLIPPING_NOT_SUPPORTED	要求されたクリッピング操作がサポートされていません。
2010	0x000007DA	ERROR_INVALID_CMM	指定された色管理モジュールが無効です。
2011	0x000007DB	ERROR_INVALID_PROFILE	指定されたカラープロファイルが無効です。
2012	0x000007DC	ERROR_TAG_NOT_FOUND	指定されたタグが見つかりませんでした。
2013	0x000007DD	ERROR_TAG_NOT_PRESENT	必要なタグがありません。
2014	0x000007DE	ERROR_DUPLICATE_TAG	指定されたタグは既に存在しています。
2015	0x000007DF	ERROR_PROFILE_NOT_ASSOCIATED_WITH_DEVICE	指定された色プロファイルがどのデバイスにも関連付けられていません。
2016	0x000007E0	ERROR_PROFILE_NOT_FOUND	指定されたカラープロファイルが見つかりませんでした。
2017	0x000007E1	ERROR_INVALID_COLORSPACE	指定された色空間が無効です。
2018	0x000007E2	ERROR_ICM_NOT_ENABLED	画像色管理が有効になっていません。
2019	0x000007E3	ERROR_DELETING_ICM_XFORM	色変換の削除中にエラーが発生しました。
2020	0x000007E4	ERROR_INVALID_TRANSFORM	指定された色変換が無効です。
2021	0x000007E5	ERROR_COLORSPACE_MISMATCH	指定された変換がビットマップの色空間と一致しません。
2022	0x000007E6	ERROR_INVALID_COLORINDEX	指定された名前付き色インデックスがプロファイルに存在しません。
2108	0x0000083C	ERROR_CONNECTED_OTHER_PASSWORD	ネットワーク接続は正常に確立されましたが、ユーザは始めに指定されたパスワード以外のパスワードを入力する必要があります。

エラー			説明
10進数	16進数	名前	
2202	0x0000089A	ERROR_BAD_USERNAME	指定されたユーザ名が無効です。
2250	0x000008CA	ERROR_NOT_CONNECTED	このネットワーク接続は存在しません。
2401	0x00000961	ERROR_OPEN_FILES	このネットワーク接続でファイルが開いているか、またはリクエストが保留されています。
2402	0x00000962	ERROR_ACTIVE_CONNECTIONS	アクティブな接続がまだ存続しています。
2404	0x00000964	ERROR_DEVICE_IN_USE	デバイスがアクティブなプロセスによって使用中のため、切断できません。
2500	0x000009C4	ERROR_PKINIT_FAILURE	スマートカードログオン中にKDC証明書の検証時にkerberosプロトコルにエラーが発生しました。
2501	0x000009C5	ERROR_SMARTCARD_SUBSYSTEM_FAILURE	スマートカードサブシステムの使用をしようとしたときに、Kerberosプロトコルにエラーが発生しました。

エラー			説明
10進数	16進数	名前	
3000	0x00000BB8	ERROR_UNKNOWN_PRINT_MONITOR	指定された印刷モニタが不明です。
3001	0x00000BB9	ERROR_PRINTER_DRIVER_IN_USE	指定されたプリンタドライバが不明です。
3002	0x00000BBA	ERROR_SPOOL_FILE_NOT_FOUND	スプールファイルが見つかりませんでした。
3003	0x00000BBB	ERROR_SPL_NO_STARTDOC	StartDocPrinter呼び出しが発行されませんでした。
3004	0x00000BBC	ERROR_SPL_NO_ADDJOB	AddJob呼び出しが発行されませんでした。
3005	0x00000BBD	ERROR_PRINT_PROCESSOR_ALREADY_INSTALLED	指定された印刷プロセッサは既にインストールされています。
3006	0x00000BBE	ERROR_PRINT_MONITOR_ALREADY_INSTALLED	指定された印刷モニタは既にインストールされています。
3007	0x00000BBF	ERROR_INVALID_PRINT_MONITOR	指定された印刷モニタに必要な機能がありません。
3008	0x00000BC0	ERROR_PRINT_MONITOR_IN_USE	指定された印刷モニタが不明です。
3009	0x00000BC1	ERROR_PRINTER_HAS_JOBS_QUEUED	プリンタへのキューにジョブがある場合は、要求された操作は許可されません。
3010	0x00000BC2	ERROR_SUCCESS_REBOOT_REQUIRED	要求された操作に成功しました。システムを再起動するまで変更は反映されません。
3011	0x00000BC3	ERROR_SUCCESS_RESTART_REQUIRED	要求された操作に成功しました。サービスを再起動するまで変更は反映されません。
3012	0x00000BC4	ERROR_PRINTER_NOT_FOUND	プリンタが見つかりませんでした。
4000	0x00000FA0	ERROR_WINS_INTERNAL	コマンド処理中にWINSにエラーが発生しました。
4001	0x00000FA1	ERROR_CAN_NOT_DEL_LOCAL_WINS	ローカルWINSが削除できません。
4002	0x00000FA2	ERROR_STATIC_INIT	ファイルからのインポートが失敗しました。
4003	0x00000FA3	ERROR_INC_BACKUP	バックアップに失敗しました。以前にフルバックアップを実行しましたか？
4004	0x00000FA4	ERROR_FULL_BACKUP	バックアップに失敗しました。データベースのバックアップを実行するディレクトリをチェックしてください。
4005	0x00000FA5	ERROR_REC_NON_EXISTENT	名前がWINSデータベースに存在しません。
4006	0x00000FA6	ERROR_RPL_NOT_ALLOWED	設定されていないパートナーとの複製は許可されていません。
4100	0x00001004	ERROR_DHCP_ADDRESS_CONFLICT	DHCPクライアントがネットワークで既に使用中のIPアドレスを取得しました。DHCPクライアントが新しいアドレスを取得するまで、ローカルインターフェイスは無効になります。
4200	0x00001068	ERROR_WMI_GUID_NOT_FOUND	渡されたGUIDがWMIデータプロバイダによって有効と認識されませんでした。
4201	0x00001069	ERROR_WMI_INSTANCE_NOT_FOUND	渡されたインスタンス名がWMIデータプロバイダによって有効と認識されませんでした。
4202	0x0000106A	ERROR_WMI_ITEMID_NOT_FOUND	渡されたデータアイテムIDがWMIデータプロバイダによって有効と認識されませんでした。
4203	0x0000106B	ERROR_WMI_TRY_AGAIN	WMIリクエストを完了できなかったため、再試行してください。
4204	0x0000106C	ERROR_WMI_DP_NOT_FOUND	WMIデータプロバイダが特定できませんでした。
4205	0x0000106D	ERROR_WMI_UNRESOLVED_INSTANCE_REF	WMIデータプロバイダが登録されていないインスタンスセットを参照しています。

エラー			説明
10進数	16進数	名前	
4206	0x0000106E	ERROR_WMI_ALREADY_ENABLED	WMIデータブロックまたはイベント通知は既に有効になっています。
4207	0x0000106F	ERROR_WMI_GUID_DISCONNECTED	WMIデータブロックは使用できなくなりました。
4208	0x00001070	ERROR_WMI_SERVER_UNAVAILABLE	WMIデータサービスは使用できません。
4209	0x00001071	ERROR_WMI_DP_FAILED	WMIデータプロバイダがリクエストの実行に失敗しました。
4210	0x00001072	ERROR_WMI_INVALID_MOF	WMI MOF情報が有効ではありません。
4211	0x00001073	ERROR_WMI_INVALID_REGINFO	WMI登録情報が有効ではありません。
4212	0x00001074	ERROR_WMI_ALREADY_DISABLED	WMIデータブロックまたはイベント通知は既に無効になっています。
4213	0x00001075	ERROR_WMI_READ_ONLY	WMIデータアイテムまたはデータブロックは読み取り専用です。
4214	0x00001076	ERROR_WMI_SET_FAILURE	WMIデータアイテムまたはデータブロックが変更できませんでした。
4300	0x000010CC	ERROR_INVALID_MEDIA	メディアIDが有効なメディアを表示していません。
4301	0x000010CD	ERROR_INVALID_LIBRARY	ライブラリIDが有効なライブラリを表示していません。
4302	0x000010CE	ERROR_INVALID_MEDIA_POOL	メディアプールIDが有効なメディアプールを表示していません。
4303	0x000010CF	ERROR_DRIVE_MEDIA_MISMATCH	ドライブとメディアの互換性がないか、または異なるライブラリに存在しています。
4304	0x000010D0	ERROR_MEDIA_OFFLINE	メディアは現在オフラインライブラリに存在しており、この操作を実行するにはオンラインにする必要があります。
4305	0x000010D1	ERROR_LIBRARY_OFFLINE	この操作はオフラインライブラリでは実行できません。
4306	0x000010D2	ERROR_EMPTY	ライブラリ、ドライブ、またはメディアプールが空です。
4307	0x000010D3	ERROR_NOT_EMPTY	この操作を実行するには、ライブラリ、ドライブ、またはメディアプールが空である必要があります。
4308	0x000010D4	ERROR_MEDIA_UNAVAILABLE	このメディアプールまたはライブラリではメディアは現在使用できません。
4309	0x000010D5	ERROR_RESOURCE_DISABLED	この操作に必要なリソースが無効になっています。
4310	0x000010D6	ERROR_INVALID_CLEANER	メディアIDが有効なクリーナを表示していません。
4311	0x000010D7	ERROR_UNABLE_TO_CLEAN	ドライブがクリーニングできない、またはクリーニングをサポートしていません。
4312	0x000010D8	ERROR_OBJECT_NOT_FOUND	オブジェクトIDが有効なオブジェクトを表示していません。
4313	0x000010D9	ERROR_DATABASE_FAILURE	データベースからの読み取りまたはデータベースへの書き込みができません。
4314	0x000010DA	ERROR_DATABASE_FULL	データベースの空き領域がありません。
4315	0x000010DB	ERROR_MEDIA_INCOMPATIBLE	メディアに、デバイスまたはメディアプールとの互換性がありません。
4316	0x000010DC	ERROR_RESOURCE_NOT_PRESENT	この操作に必要なリソースが存在しません。
4317	0x000010DD	ERROR_INVALID_OPERATION	操作IDが有効ではありません。
4318	0x000010DE	ERROR_MEDIA_NOT_AVAILABLE	メディアがマウントされていないか、または準備ができていません。
4319	0x000010DF	ERROR_DEVICE_NOT_AVAILABLE	デバイスの準備ができていません。
4320	0x000010E0	ERROR_REQUEST_REFUSED	オペレータまたは管理者がリクエストを拒否しました。
4321	0x000010E1	ERROR_INVALID_DRIVE_OBJECT	ドライブIDが有効なドライブを表示していません。
4322	0x000010E2	ERROR_LIBRARY_FULL	ライブラリの空き領域がありません。使用できるスロットがありません。
4323	0x000010E3	ERROR_MEDIUM_NOT_ACCESSIBLE	転送がメディアにアクセスできません。
4324	0x000010E4	ERROR_UNABLE_TO_LOAD_MEDIUM	メディアをドライブにロードできません。

エラー			説明
10進数	16進数	名前	
4325	0x000010E5	ERROR_UNABLE_TO_INVENTORY_DRIVE	ドライブに関するステータスを読み出すことができません。
4326	0x000010E6	ERROR_UNABLE_TO_INVENTORY_SLOT	スロットに関するステータスを読み出すことができません。
4327	0x000010E7	ERROR_UNABLE_TO_INVENTORY_TRANSPORT	転送に関するステータスを読み出すことができません。
4328	0x000010E8	ERROR_TRANSPORT_FULL	既に使用されているため、転送を使用できません。
4329	0x000010E9	ERROR_CONTROLLING_IEPORT	挿入/排出ポートを開くまたは閉じることができません。
4330	0x000010EA	ERROR_UNABLE_TO_EJECT_MOUNTED_MEDIA	メディアがドライブの中にあるため、メディアを排出できません。
4331	0x000010EB	ERROR_CLEANER_SLOT_SET	クリーナスロットは既に予約されています。
4332	0x000010EC	ERROR_CLEANER_SLOT_NOT_SET	クリーナスロットは予約されていません。
4333	0x000010ED	ERROR_CLEANER_CARTRIDGE_SPENT	クリーナカートリッジがドライブクリーニングの最大数を実行しました。
4334	0x000010EE	ERROR_UNEXPECTED_OMID	不明なメディア上の識別子
4335	0x000010EF	ERROR_GANT_DELETE_LAST_ITEM	このグループまたはリソースで最後に残ったアイテムは削除できません。
4336	0x000010F0	ERROR_MESSAGE_EXCEEDS_MAX_SIZE	提供されたメッセージがこのパラメータに許される最大サイズを超えています。
4337	0x000010F1	ERROR_VOLUME_CONTAINS_SYS_FILES	ボリュームにシステムまたはページファイルが含まれています。
4338	0x000010F2	ERROR_INDIGENOUS_TYPE	ライブラリの少なくとも1つのドライブがこのメディアタイプをしていることを報告するまでは、メディアタイプをこのライブラリから削除することはできません。
4339	0x000010F3	ERROR_NO_SUPPORTING_DRIVES	使用可能な有効なドライブが存在しないため、このオフラインメディアはこのシステムにマウントできません。
4340	0x000010F4	ERROR_CLEANER_CARTRIDGE_INSTALLED	クリーナカートリッジがテープライブラリに存在します。
4350	0x000010FE	ERROR_FILE_OFFLINE	リモートストレージサービスがファイルを回収することができませんでした。
4351	0x000010FF	ERROR_REMOTE_STORAGE_NOT_ACTIVE	リモートストレージサービスはこの時点で使用できません。
4352	0x00001100	ERROR_REMOTE_STORAGE_MEDIA_ERROR	リモートストレージサービスでメディアエラーが発生しました。
4390	0x00001126	ERROR_NOT_A_REPARSE_POINT	このファイルまたはディレクトリはリパースポイントではありません。
4391	0x00001127	ERROR_REPARSE_ATTRIBUTE_CONFLICT	既存の属性と競合するため、リパースポイント属性が設定できません。
4392	0x00001128	ERROR_INVALID_REPARSE_DATA	リパースポイントバッファに存在するデータが無効です。
4393	0x00001129	ERROR_REPARSE_TAG_INVALID	リパースポイントバッファに存在するタグが無効です。
4394	0x0000112A	ERROR_REPARSE_TAG_MISMATCH	リクエストで指定されたタグとリパースポイントに存在するタグには不一致があります。
4500	0x00001194	ERROR_VOLUME_NOT_SIS_ENABLED	シングルインスタンスストレージはこのボリュームでは使用できません。

エラー			説明
10進数	16進数	名前	
5001	0x00001389	ERROR_DEPENDENT_RESOURCE_EXISTS	他のリソースが依存しているため、クラスタリソースを他のグループに移動できません。
5002	0x0000138A	ERROR_DEPENDENCY_NOT_FOUND	クラスタリソースの依存性が見つかりませんでした。
5003	0x0000138B	ERROR_DEPENDENCY_ALREADY_EXISTS	既に依存関係が存在するため、指定されたリソースに対してクラスタリソースの依存関係を作成することはできません。
5004	0x0000138C	ERROR_RESOURCE_NOT_ONLINE	クラスタリソースがオンラインではありません。
5005	0x0000138D	ERROR_HOST_NODE_NOT_AVAILABLE	クラスタノードはこの操作に使用できません。

エラー			説明
10進数	16進数	名前	
5006	0x0000138E	ERROR_RESOURCE_NOT_AVAILABLE	クラスタリソースが使用できません。
5007	0x0000138F	ERROR_RESOURCE_NOT_FOUND	クラスタリソースが見つかりませんでした。
5008	0x00001390	ERROR_SHUTDOWN_CLUSTER	クラスタがシャットダウン中です。
5009	0x00001391	ERROR_CANT_EVICT_ACTIVE_NODE	ノードを停止しなければ、クラスタノードはクラスタから削除できません。
5010	0x00001392	ERROR_OBJECT_ALREADY_EXISTS	オブジェクトは既に存在しています。
5011	0x00001393	ERROR_OBJECT_IN_LIST	オブジェクトはリストに既に含まれています。
5012	0x00001394	ERROR_GROUP_NOT_AVAILABLE	このクラスタグループは、新しいリクエストには使用できません。
5013	0x00001395	ERROR_GROUP_NOT_FOUND	クラスタグループが見つかりませんでした。
5014	0x00001396	ERROR_GROUP_NOT_ONLINE	クラスタグループがオンラインでないため、この操作を完了できませんでした。
5015	0x00001397	ERROR_HOST_NODE_NOT_RESOURCE_OWNER	クラスタノードはリソースの所有者ではありません。
5016	0x00001398	ERROR_HOST_NODE_NOT_GROUP_OWNER	クラスタノードはグループの所有者ではありません。
5017	0x00001399	ERROR_RESMON_CREATE_FAILED	クラスタリソースが指定されたリソースモニタに作成できませんでした。
5018	0x0000139A	ERROR_RESMON_ONLINE_FAILED	クラスタリソースがリソースモニタによってオンラインにできませんでした。
5019	0x0000139B	ERROR_RESOURCE_ONLINE	クラスタリソースがオンラインでないため、この操作を完了できませんでした。
5020	0x0000139C	ERROR_QUORUM_RESOURCE	クォーラムリソースであるため、クラスタリソースが削除またはオンラインにできませんでした。
5021	0x0000139D	ERROR_NOT_QUORUM_CAPABLE	指定されたリソースがクォーラムリソースになれないため、クラスタがそのリソースをクォーラムリソースにすることができませんでした。
5022	0x0000139E	ERROR_CLUSTER_SHUTTING_DOWN	クラスタソフトウェアがシャットダウン中です。
5023	0x0000139F	ERROR_INVALID_STATE	グループまたはリソースが要求された操作を実行するために正しい状態ではありません。
5024	0x000013A0	ERROR_RESOURCE_PROPERTIES_STORED	プロパティは保存されますが、次にリソースがオンラインになるまではすべての変更が反映されません。
5025	0x000013A1	ERROR_NOT_QUORUM_CLASS	指定されたリソースが共有ストレージクラスに属していないため、クラスタがそのリソースをクォーラムリソースにすることができませんでした。
5026	0x000013A2	ERROR_CORE_RESOURCE	クラスリソースがコアリソースであるため削除できませんでした。
5027	0x000013A3	ERROR_QUORUM_RESOURCE_ONLINE_FAILED	クォーラムリソースがオンラインになるのに失敗しました。
5028	0x000013A4	ERROR_QUORUMLOG_OPEN_FAILED	クォーラムログが正常に作成またはマウントできませんでした。
5029	0x000013A5	ERROR_CLUSTERLOG_CORRUPT	クラスタログが破損しています。
5030	0x000013A6	ERROR_CLUSTERLOG_RECORD_EXCEEDS_MAXSIZE	最大サイズを超えているため、レコードがクラスタログに書き込むことができませんでした。
5031	0x000013A7	ERROR_CLUSTERLOG_EXCEEDS_MAXSIZE	クラスタログが実際サイズを超えています。
5032	0x000013A8	ERROR_CLUSTERLOG_CHKPOINT_NOT_FOUND	チェックポイントレコードがクラスタログに見つかりませんでした。
5033	0x000013A9	ERROR_CLUSTERLOG_NOT_ENOUGH_SPACE	ロギングに必要な最小ディスクスペースが使用できません。
5034	0x000013AA	ERROR_QUORUM_OWNER_ALIVE	リソースが他のアクティブノードによって所有されているため、クラスタノードがクォーラムリソースを制御することに失敗しました。
5035	0x000013AB	ERROR_NETWORK_NOT_AVAILABLE	クラスタネットワークはこの操作に使用できません。
5036	0x000013AC	ERROR_NODE_NOT_AVAILABLE	クラスタノードはこの操作に使用できません。
5037	0x000013AD	ERROR_ALL_NODES_NOT_AVAILABLE	すべてのクラスタノードがこの操作を実行するために実行中である必要があります。
5038	0x000013AE	ERROR_RESOURCE_FAILED	クラスタリソースが失敗しました。
5039	0x000013AF	ERROR_CLUSTER_INVALID_NODE	クラスタノードが有効ではありません。

		エラー		説明
10進数	16進数	名前		
5040	0x000013B0	ERROR_CLUSTER_NODE_EXISTS		クラスタノードは既に存在しています。
5041	0x000013B1	ERROR_CLUSTER_JOIN_IN_PROGRESS		ノードはクラスタに結合中です。
5042	0x000013B2	ERROR_CLUSTER_NODE_NOT_FOUND		クラスタノードが見つかりませんでした。
5043	0x000013B3	ERROR_CLUSTER_LOCAL_NODE_NOT_FOUND		クラスタローカルノード情報が見つかりませんでした。
5044	0x000013B4	ERROR_CLUSTER_NETWORK_EXISTS		クラスタネットワークは既に存在しています。
5045	0x000013B5	ERROR_CLUSTER_NETWORK_NOT_FOUND		クラスタネットワークが見つかりませんでした。
5046	0x000013B6	ERROR_CLUSTER_NETINTERFACE_EXISTS		クラスタネットワークインターフェイスは既に存在しています。
5047	0x000013B7	ERROR_CLUSTER_NETINTERFACE_NOT_FOUND		クラスタネットワークインターフェイスが見つかりませんでした。
5048	0x000013B8	ERROR_CLUSTER_INVALID_REQUEST		クラスタリクエストはこのオブジェクトでは有効ではありません。
5049	0x000013B9	ERROR_CLUSTER_INVALID_NETWORK_PROVIDER		クラスタネットワークプロバイダが有効ではありません。
5050	0x000013BA	ERROR_CLUSTER_NODE_DOWN		クラスタノードが停止しています。
5051	0x000013BB	ERROR_CLUSTER_NODE_UNREACHABLE		クラスタノードに到達できません。
5052	0x000013BC	ERROR_CLUSTER_NODE_NOT_MEMBER		クラスタノードはクラスタのメンバではありません。
5053	0x000013BD	ERROR_CLUSTER_JOIN_NOT_IN_PROGRESS		クラスタ結合操作は進行中ではありません。
5054	0x000013BE	ERROR_CLUSTER_INVALID_NETWORK		クラスタネットワークが有効ではありません。
5056	0x000013C0	ERROR_CLUSTER_NODE_UP		クラスタノードが起動しています。
5057	0x000013C1	ERROR_CLUSTER_IPADDR_IN_USE		クラスタのIPアドレスが既に使用されています。
5058	0x000013C2	ERROR_CLUSTER_NODE_NOT_PAUSED		クラスタノードが一時停止していません。
5059	0x000013C3	ERROR_CLUSTER_NO_SECURITY_CONTEXT		クラスタセキュリティコンテキストが使用できません。
5060	0x000013C4	ERROR_CLUSTER_NETWORK_NOT_INTERNAL		クラスタネットワークが内部クラスタ通信用に構成されていません。
5061	0x000013C5	ERROR_CLUSTER_NODE_ALREADY_UP		クラスタノードが既に起動しています。
5062	0x000013C6	ERROR_CLUSTER_NODE_ALREADY_DOWN		クラスタノードが既に停止しています。
5063	0x000013C7	ERROR_CLUSTER_NETWORK_ALREADY_ONLINE		クラスタネットワークが既にオンラインになっています。
5064	0x000013C8	ERROR_CLUSTER_NETWORK_ALREADY_OFFLINE		クラスタネットワークが既にオフラインになっています。
5065	0x000013C9	ERROR_CLUSTER_NODE_ALREADY_MEMBER		クラスタノードは既にクラスタのメンバです。
5066	0x000013CA	ERROR_CLUSTER_LAST_INTERNAL_NETWORK		クラスタネットワークは、2つ以上のクラスタノード間の内部クラスタ通信用にのみ構成されています。内部通信機能は、ネットワークから削除できません。
5067	0x000013CB	ERROR_CLUSTER_NETWORK_HAS_DEPENDENTS		1つ以上のクラスタリソースが、クライアントにサービスを提供するためにネットワークに依存しています。クライアントアクセス機能は、ネットワークから削除できません。
5068	0x000013CC	ERROR_INVALID_OPERATION_ON_QUORUM		この操作は、クォラムリソースとしてクラスタリソース上で実行できません。クォラムリソースをオフラインにしたり、またはその所有者リストを変更したりすることはできません。
5069	0x000013CD	ERROR_DEPENDENCY_NOT_ALLOWED		クラスタのクォラムリソースには、依存関係を持たせることはできません。
5070	0x000013CE	ERROR_CLUSTER_NODE_PAUSED		クラスタノードが一時停止しています。
5071	0x000013CF	ERROR_NODE_CANT_HOST_RESOURCE		クラスタリソースはオンラインにできません。所有者ノードはこのリソースを実行できません。
5072	0x000013D0	ERROR_CLUSTER_NODE_NOT_READY		クラスタノードが要求された操作を実行するための準備ができていません。
5073	0x000013D1	ERROR_CLUSTER_NODE_SHUTTING_DOWN		クラスタノードがシャットダウン中です。
5074	0x000013D2	ERROR_CLUSTER_JOIN_ABORTED		クラスタの結合操作が中止されました。
5075	0x000013D3	ERROR_CLUSTER_INCOMPATIBLE_VERSIONS		結合するノードとそのスポンサの間でソフトウェアバージョンの互換性がないために、クラスタ結合操作が失敗しました。
5076	0x000013D4	ERROR_CLUSTER_MAXNUM_OF_RESOURCES_EXCEEDED		クラスタがモニタできるリソースの限界値に達しているため、このリソースは作成できません。

エラー			説明
10進数	16進数	名前	
5077	0x000013D5	ERROR_CLUSTER_SYSTEM_CONFIG_CHANGED	システム設定がクラスタの結合または構築操作中に変更されました。結合または構築操作が中止されました。
5078	0x000013D6	ERROR_CLUSTER_RESOURCE_TYPE_NOT_FOUND	指定されたリソースタイプが見つかりませんでした。
5079	0x000013D7	ERROR_CLUSTER_RESTYPE_NOT_SUPPORTED	指定されたノードはこのタイプのリソースをサポートしていません。これは、バージョンの不整合やこのノードにリソースDLLが存在していないためです。
5080	0x000013D8	ERROR_CLUSTER_RESNAME_NOT_FOUND	指定されたリソース名は、このリソースDLLでサポートされています。これは、リソースDLLに無効な名前(または変更された名前)が提供されたことが原因の可能性あります。
5081	0x000013D9	ERROR_CLUSTER_NO_RPC_PACKAGES_REGISTERED	認証パッケージがRPCサーバに登録できませんでした。
5082	0x000013DA	ERROR_CLUSTER_OWNER_NOT_IN_PREFLIST	グループの所有者がグループの推奨リストに含まれていないため、グループをオンラインにすることができません。グループの所有者ノードを変更するには、グループを移動します。
5083	0x000013DB	ERROR_CLUSTER_DATABASE_SEQMISMATCH	クラスタデータベースのシーケンス番号が変更されたか、またはロッカーノードと互換性がないため、結合操作が失敗しました。このエラーは、クラスタデータベースが結合中に変更される場合、結合操作中に発生する場合があります。
5084	0x000013DC	ERROR_RESMON_INVALID_STATE	リソースモニタは、リソースが現在の状態である間は、フェイルオペレーションの実行を許可しません。このエラーは、リソースが保留状態であるときに発生する場合があります。
5085	0x000013DD	ERROR_CLUSTER_GUM_NOT_LOCKER	非ロッカーノードが、グローバルアップデートを実行するロックを予約するリクエスト取得しました。
5086	0x000013DE	ERROR_QUORUM_DISK_NOT_FOUND	クォーラムディスクがクラスタサービスによって特定できませんでした。
5087	0x000013DF	ERROR_DATABASE_BACKUP_CORRUPT	バックアップクラスタデータベースがおそらく破損しています。
5088	0x000013E0	ERROR_CLUSTER_NODE_ALREADY_HAS_DFS_ROOT	DFSルートは既にクラスタノードに存在しています。
5089	0x000013E1	ERROR_RESOURCE_PROPERTY_UNCHANGEABLE	他の既存のプロパティと競合するため、リソースプロパティの変更に失敗しました。
5890	0x00001702	ERROR_CLUSTER_MEMBERSHIP_INVALID_STATE	ノードの現在のメンバーシップ状態と互換性のない操作を実行しようとして失敗しました。
5891	0x00001703	ERROR_CLUSTER_QUORUMLOG_NOT_FOUND	クォーラムリソースにクォーラムログが含まれていません。
5892	0x00001704	ERROR_CLUSTER_MEMBERSHIP_HALT	メンバーシップエンジンがこのノードでクラスタサービスのシャットダウンを要求しました。
5893	0x00001705	ERROR_CLUSTER_INSTANCE_ID_MISMATCH	結合するノードのクラスタインスタンスIDがスポンサードノードのクラスタインスタンスIDと一致しないため、結合操作が失敗しました。
5894	0x00001706	ERROR_CLUSTER_NETWORK_NOT_FOUND_FOR_IP	指定されたIPアドレスに一致するネットワークが見つかりませんでした。サブネットマスクとクラスタネットワークも指定してください。
5895	0x00001707	ERROR_CLUSTER_PROPERTY_DATA_TYPE_MISMATCH	プロパティの実際のデータタイプがプロパティの所期のデータタイプと一致しませんでした。
5896	0x00001708	ERROR_CLUSTER_EVICT_WITHOUT_CLEANUP	クラスタノードが正常にクラスタから削除されました。このノードは、削除クリーンアップ機能をサポートしていないため、クリーンアップされませんでした。

エラー			説明
10進数	16進数	名前	
6000	0x00001770	ERROR_ENCRYPTION_FAILED	指定されたファイルが暗号化できませんでした。
6001	0x00001771	ERROR_DECRYPTION_FAILED	指定されたファイルが復号できませんでした。
6002	0x00001772	ERROR_FILE_ENCRYPTED	指定されたファイルが暗号化され、ユーザが復号することはできません。

エラー			説明
10進数	16進数	名前	
6003	0x00001773	ERROR_NO_RECOVERY_POLICY	このシステムには有効な暗号化復元ポリシーは設定されていません。
6004	0x00001774	ERROR_NO_EFS	必要な暗号化ドライバがこのシステムにロードされていません。
6005	0x00001775	ERROR_WRONG_EFS	ファイルが現在ロードされている暗号化ドライバとは異なるドライバを使用して暗号化されました。
6006	0x00001776	ERROR_NO_USER_KEYS	このユーザにはEFSキーが定義されていません。
6007	0x00001777	ERROR_FILE_NOT_ENCRYPTED	指定されたファイルが暗号化されていません。
6008	0x00001778	ERROR_NOT_EXPORT_FORMAT	指定されたファイルが定義されたEFSエクスポート形式ではありません。
6009	0x00001779	ERROR_FILE_READ_ONLY	指定されたファイルは読み取り専用です。
6010	0x0000177A	ERROR_DIR_EFS_DISALLOWED	ディレクトリが暗号化のために無効になっています。
6011	0x0000177B	ERROR_EFS_SERVER_NOT_TRUSTED	サーバがリモート暗号化操作に必要な信頼性を備えていません。
6012	0x0000177C	ERROR_BAD_RECOVERY_POLICY	このシステムに設定されている復元ポリシーに無効な復元証明書が含まれています。
6013	0x0000177D	ERROR_EFS_ALG_BLOB_TOO_BIG	ソースファイルに使用されている暗号化アルゴリズムには、宛先ファイルのものより大きな鍵バッファが必要です。
6014	0x0000177E	ERROR_VOLUME_NOT_SUPPORT_EFS	ディスクパーティションがファイルの暗号化をサポートしていません。
6118	0x000017E6	ERROR_NO_BROWSER_SERVERS_FOUND	このワークグループのサーバリストは現在使用できません。
6200	0x00001838	SCHED_E_SERVICE_NOT_LOCALSYSTEM	Task Schedulerサービスは、システムアカウントで実行し適切に機能するように設定する必要があります。個々のタスクは、別のアカウントで実行するように設定できます。
7001	0x00001B59	ERROR_CTX_WINSTATION_NAME_INVALID	指定されたセッション名が無効です。
7002	0x00001B5A	ERROR_CTX_INVALID_PD	指定されたプロトコルドライバが無効です。
7003	0x00001B5B	ERROR_CTX_PD_NOT_FOUND	指定されたプロトコルドライバがシステムパスで見つかりませんでした。
7004	0x00001B5C	ERROR_CTX_WD_NOT_FOUND	指定されたターミナル接続ドライバがシステムパスで見つかりませんでした。
7005	0x00001B5D	ERROR_CTX_CANNOT_MAKE_EVENTLOG_ENTRY	イベントログ用のレジストリキーがこのセッションで作成できませんでした。
7006	0x00001B5E	ERROR_CTX_SERVICE_NAME_COLLISION	同じ名前のサービスがシステム上に既に存在しています。
7007	0x00001B5F	ERROR_CTX_CLOSE_PENDING	セッションの終了操作は保留されています。
7008	0x00001B60	ERROR_CTX_NO_OUTBUF	使用可能な出力バッファがありません。
7009	0x00001B61	ERROR_CTX_MODEM_INF_NOT_FOUND	MODEM.INFファイルが見つかりませんでした。
7010	0x00001B62	ERROR_CTX_INVALID_MODEMNAME	MODEM.INFにモデム名が見つかりませんでした。
7011	0x00001B63	ERROR_CTX_MODEM_RESPONSE_ERROR	モデムが送信されたコマンドを受理しませんでした。設定されているモデム名と接続されているモデムが一致することを確認してください。
7012	0x00001B64	ERROR_CTX_MODEM_RESPONSE_TIMEOUT	モデムが送信されたコマンドに回答しませんでした。モデムのケーブルが正しく接続され、電源がオンになっていることを確認してください。
7013	0x00001B65	ERROR_CTX_MODEM_RESPONSE_NO_CARRIER	キャリア検出が失敗したか、またはキャリアが切断によって停止しています。
7014	0x00001B66	ERROR_CTX_MODEM_RESPONSE_NO_DIALTONE	所定の時間内にダイヤルトーンが検出されませんでした。電話ケーブルが正しく接続され作動していることを確認してください。
7015	0x00001B67	ERROR_CTX_MODEM_RESPONSE_BUSY	コールバック時にリモートサイトでビジー信号が検出されました。
7016	0x00001B68	ERROR_CTX_MODEM_RESPONSE_VOICE	コールバック時にリモートサイトで音声を検出されました。
7017	0x00001B69	ERROR_CTX_TD_ERROR	転送ドライバエラー
7022	0x00001B6E	ERROR_CTX_WINSTATION_NOT_FOUND	指定されたセッションが見つかりません。
7023	0x00001B6F	ERROR_CTX_WINSTATION_ALREADY_EXISTS	指定されたセッション名は既に使用されていません。

エラー			説明
10進数	16進数	名前	
7024	0x00001B70	ERROR_CTX_WINSTATION_BUSY	接続、切断、リセット、または削除操作を処理するためにターミナル接続が現在ビジーであるため、要求された操作を完了できませんでした。
7025	0x00001B71	ERROR_CTX_BAD_VIDEO_MODE	現在のクライアントではサポートされていないビデオモードのセッションを接続しようとしました。
7035	0x00001B7B	ERROR_CTX_GRAPHICS_INVALID	アプリケーションがDOSグラフィックモードを有効にしようしました。DOSグラフィックモードはサポートされていません
7037	0x00001B7D	ERROR_CTX_LOGON_DISABLED	対話的なログオン権限が無効になっています。システム管理者にお問い合わせください。
7038	0x00001B7E	ERROR_CTX_NOT_CONSOLE	要求された操作は、システムコンソールでのみ実行できます。これは、コンソールへの直接アクセスが必要なドライバまたはシステムDLLによくある結果です。
7040	0x00001B80	ERROR_CTX_CLIENT_QUERY_TIMEOUT	クライアントがサーバ接続メッセージへの応答に失敗しました。
7041	0x00001B81	ERROR_CTX_CONSOLE_DISCONNECT	コンソールセッションの切断はサポートされていません。
7042	0x00001B82	ERROR_CTX_CONSOLE_CONNECT	切断されたセッションのコンソールへの再接続はサポートされていません。
7044	0x00001B84	ERROR_CTX_SHADOW_DENIED	別のセッションをリモートで制御するリクエストは拒否されました。
7045	0x00001B85	ERROR_CTX_WINSTATION_ACCESS_DENIED	要求されたセッションアクセスが拒否されました。
7049	0x00001B89	ERROR_CTX_INVALID_WD	指定されたターミナル接続ドライバが無効です。
7050	0x00001B8A	ERROR_CTX_SHADOW_INVALID	要求されたセッションはリモートで制御できません。これは、セッションが切断されたか、または現在ユーザがログオンしていないことが原因である場合があります。
7051	0x00001B8B	ERROR_CTX_SHADOW_DISABLED	要求されたセッションは、リモート制御を許可するように設定されていません。
7052	0x00001B8C	ERROR_CTX_CLIENT_LICENSE_IN_USE	このターミナルサーバに接続するリクエストは拒否されました。ターミナルサーバのクライアントライセンス番号は、現在、別のユーザによって使用されています。固有のライセンス番号を取得するようにシステム管理者に依頼してください。
7053	0x00001B8D	ERROR_CTX_CLIENT_LICENSE_NOT_SET	このターミナルサーバに接続するリクエストは拒否されました。ターミナルサーバのクライアントライセンス番号が、ターミナルサーバクライアントのこのコピーに入力されていません。システム管理者にお問い合わせください。
7054	0x00001B8E	ERROR_CTX_LICENSE_NOT_AVAILABLE	システムはライセンスされているログオン制限に達しました。後にもう一度お試しください。
7055	0x00001B8F	ERROR_CTX_LICENSE_CLIENT_INVALID	使用中のクライアントには、このシステムを使用するライセンスがありません。ログオンのリクエストは拒否されました。
7056	0x00001B90	ERROR_CTX_LICENSE_EXPIRED	システムライセンスの期限が過ぎています。ログオンのリクエストは拒否されました。
7057	0x00001B91	ERROR_CTX_SHADOW_NOT_RUNNING	指定されたセッションが現在リモートで制御されていないため、リモート制御が終了できませんでした。

エラー			説明
10進数	16進数	名前	
8001	0x00001F41	FRS_ERR_INVALID_API_SEQUENCE	ファイル複製サービスAPIが誤って呼び出されました。
8002	0x00001F42	FRS_ERR_STARTING_SERVICE	ファイル複製サービスを開始できません。
8003	0x00001F43	FRS_ERR_STOPPING_SERVICE	ファイル複製サービスを停止できません。
8004	0x00001F44	FRS_ERR_INTERNAL_API	ファイル複製サービスAPIがリクエストを終了しました。イベントログから詳細情報が得られる場合があります。
8005	0x00001F45	FRS_ERR_INTERNAL	ファイル複製サービスがリクエストを終了しました。イベントログから詳細情報が得られる場合があります。

エラー			説明
10進数	16進数	名前	
8006	0x00001F46	FRS_ERR_SERVICE_COMM	ファイル複製サービスに接続できません。イベントログから詳細情報が得られる場合があります。
8007	0x00001F47	FRS_ERR_INSUFFICIENT_PRIV	ユーザの権限が十分ではないため、ファイル複製サービスがリクエストに対応できません。イベントログから詳細情報が得られる場合があります。
8008	0x00001F48	FRS_ERR_AUTHENTICATION	認証されたRPCが使用できないため、ファイル複製サービスがリクエストに対応できません。イベントログから詳細情報が得られる場合があります。
8009	0x00001F49	FRS_ERR_PARENT_INSUFFICIENT_PRIV	ドメインコントローラでのユーザの権限が十分ではないため、ファイル複製サービスがリクエストに対応できません。イベントログから詳細情報が得られる場合があります。
8010	0x00001F4A	FRS_ERR_PARENT_AUTHENTICATION	ドメインコントローラで認証されたRPCが使用できないため、ファイル複製サービスがリクエストに対応できません。イベントログから詳細情報が得られる場合があります。
8011	0x00001F4B	FRS_ERR_CHILD_TO_PARENT_COMM	ファイル複製サービスがドメインコントローラ上のファイル複製サービスと通信できません。イベントログから詳細情報が得られる場合があります。
8012	0x00001F4C	FRS_ERR_PARENT_TO_CHILD_COMM	ドメインコントローラ上のファイル複製サービスがこのコンピュータ上のファイル複製サービスと通信できません。イベントログから詳細情報が得られる場合があります。
8013	0x00001F4D	FRS_ERR_SYSVOL_POPULATE	内部エラーのため、ファイル複製サービスがシステムボリュームを格納できません。イベントログから詳細情報が得られる場合があります。
8014	0x00001F4E	FRS_ERR_SYSVOL_POPULATE_TIMEOUT	内部タイムアウトのため、ファイル複製サービスがシステムボリュームを格納できません。イベントログから詳細情報が得られる場合があります。
8015	0x00001F4F	FRS_ERR_SYSVOL_IS_BUSY	ファイル複製サービスはリクエストを処理できません。システムボリュームが前のリクエストによってビジー状態です。
8016	0x00001F50	FRS_ERR_SYSVOL_DEMOTE	内部エラーのため、ファイル複製サービスがシステムボリュームの複製を停止できません。イベントログから詳細情報が得られる場合があります。
8017	0x00001F51	FRS_ERR_INVALID_SERVICE_PARAMETER	ファイル複製サービスが無効なパラメータを検出しました。
8200	0x00002008	ERROR_DS_NOT_INSTALLED	ディレクトリサービスをインストール中にエラーが発生しました。詳細情報については、イベントログを参照してください。
8201	0x00002009	ERROR_DS_MEMBERSHIP_EVALUATED_LOCALLY	ディレクトリサービスがローカルでグループメンバーシップを判別しました。
8202	0x0000200A	ERROR_DS_NO_ATTRIBUTE_OR_VALUE	指定されたディレクトリサービス属性または値は存在しません。
8203	0x0000200B	ERROR_DS_INVALID_ATTRIBUTE_SYNTAX	ディレクトリサービスに対して指定された属性構文が無効です。
8204	0x0000200C	ERROR_DS_ATTRIBUTE_TYPE_UNDEFINED	ディレクトリサービスに対して指定された属性タイプが定義されていません。
8205	0x0000200D	ERROR_DS_ATTRIBUTE_OR_VALUE_EXISTS	指定されたディレクトリサービス属性または値は既に存在します。
8206	0x0000200E	ERROR_DS_BUSY	ディレクトリサービスがビジーです。
8207	0x0000200F	ERROR_DS_UNAVAILABLE	ディレクトリサービスが使用できません。
8208	0x00002010	ERROR_DS_NO_RIDS_ALLOCATED	ディレクトリサービスは相対IDを割り当てるのができませんでした。
8209	0x00002011	ERROR_DS_NO_MORE_RIDS	ディレクトリサービスが相対IDのプールを使い果たしています。
8210	0x00002012	ERROR_DS_INCORRECT_ROLE_OWNER	ディレクトリサービスがこのタイプの操作のマスターではないため、要求されたオペレーションが実行できませんでした。
8211	0x00002013	ERROR_DS_RIDMGR_INIT_ERROR	ディレクトリサービスが相対IDを割り当てるサブシステムを初期化できませんでした。

		エラー		説明
10進数	16進数	名前		
8212	0x00002014	ERROR_DS_OBJ_CLASS_VIOLATION		要求された操作が、オブジェクトのクラスに関連付けられた1つ以上の制約条件を満たすことができませんでした。
8213	0x00002015	ERROR_DS_CANT_ON_NON_LEAF		ディレクトリサービスは、リーフオブジェクトでのみ要求された操作を実行できます。
8214	0x00002016	ERROR_DS_CANT_ON_RDN		ディレクトリサービスがオブジェクトのRDN属性に関する要求された操作を実行できません。
8215	0x00002017	ERROR_DS_CANT_MOD_OBJ_CLASS		ディレクトリサービスがオブジェクトのオブジェクトクラスを変更しようとした。
8216	0x00002018	ERROR_DS_CROSS_DOM_MOVE_ERROR		要求されたクロスドメイン移動操作が実行できませんでした。
8217	0x00002019	ERROR_DS_GC_NOT_AVAILABLE		グローバルカタログサーバに接続できません。
8218	0x0000201A	ERROR_SHARED_POLICY		ポリシーオブジェクトが共有されており、ルートでのみ変更できます。
8219	0x0000201B	ERROR_POLICY_OBJECT_NOT_FOUND		ポリシーオブジェクトが存在しません。
8220	0x0000201C	ERROR_POLICY_ONLY_IN_DS		要求されたポリシー情報がディレクトリサービスのみにあります。
8221	0x0000201D	ERROR_PROMOTION_ACTIVE		ドメインコントローラの昇格が現在アクティブです。
8222	0x0000201E	ERROR_NO_PROMOTION_ACTIVE		ドメインコントローラの昇格が現在アクティブではありません。
8224	0x00002020	ERROR_DS_OPERATIONS_ERROR		操作エラーが発生しました。
8225	0x00002021	ERROR_DS_PROTOCOL_ERROR		プロトコルエラーが発生しました。
8226	0x00002022	ERROR_DS_TIMELIMIT_EXCEEDED		このリクエストに対する時間制限を超えました。
8227	0x00002023	ERROR_DS_SIZELIMIT_EXCEEDED		このリクエストに対するサイズ制限を超えました。
8228	0x00002024	ERROR_DS_ADMIN_LIMIT_EXCEEDED		このリクエストに対する管理制限を超えました。
8229	0x00002025	ERROR_DS_COMPARE_FALSE		この比較応答はfalse(偽)です。
8230	0x00002026	ERROR_DS_COMPARE_TRUE		この比較応答はtrue(真)です。
8231	0x00002027	ERROR_DS_AUTH_METHOD_NOT_SUPPORTED		要求された認証方法が、サーバではサポートされていません。
8232	0x00002028	ERROR_DS_STRONG_AUTH_REQUIRED		このサーバにはこれよりセキュアな認証方法が必要です。
8233	0x00002029	ERROR_DS_INAPPROPRIATE_AUTH		不適切な認証。
8234	0x0000202A	ERROR_DS_AUTH_UNKNOWN		認証メカニズムが不明です。
8235	0x0000202B	ERROR_DS_REFERRAL		照会がサーバから返されました。
8236	0x0000202C	ERROR_DS_UNAVAILABLE_CRIT_EXTENSION		サーバは要求されたクリティカル拡張をサポートしていません。
8237	0x0000202D	ERROR_DS_CONFIDENTIALITY_REQUIRED		このリクエストにはセキュアな接続が必要です。
8238	0x0000202E	ERROR_DS_INAPPROPRIATE_MATCHING		不適切な一致。
8239	0x0000202F	ERROR_DS_CONSTRAINT_VIOLATION		制約条件の違反が発生しました。
8240	0x00002030	ERROR_DS_NO_SUCH_OBJECT		そのようなオブジェクトはサーバ上にありません。
8241	0x00002031	ERROR_DS_ALIAS_PROBLEM		エイリアスに問題があります。
8242	0x00002032	ERROR_DS_INVALID_DN_SYNTAX		無効なdn構文が指定されています。
8243	0x00002033	ERROR_DS_IS_LEAF		このオブジェクトはリーフオブジェクトです。
8244	0x00002034	ERROR_DS_ALIAS_DEREF_PROBLEM		エイリアスの逆参照に問題があります。
8245	0x00002035	ERROR_DS_UNWILLING_TO_PERFORM		サーバがリクエストを処理しようとしません。
8246	0x00002036	ERROR_DS_LOOP_DETECT		ループが検出されました。
8247	0x00002037	ERROR_DS_NAMING_VIOLATION		命名規則の違反があります。
8248	0x00002038	ERROR_DS_OBJECT_RESULTS_TOO_LARGE		結果セットが大きすぎます。
8249	0x00002039	ERROR_DS_AFFECTS_MULTIPLE_DSAS		操作が複数のDSAに影響しています
8250	0x0000203A	ERROR_DS_SERVER_DOWN		サーバが作動していません。
8251	0x0000203B	ERROR_DS_LOCAL_ERROR		ローカルエラーが発生しました。
8252	0x0000203C	ERROR_DS_ENCODING_ERROR		エンコードエラーが発生しました。
8253	0x0000203D	ERROR_DS_DECODING_ERROR		デコードエラーが発生しました。
8254	0x0000203E	ERROR_DS_FILTER_UNKNOWN		検索フィルタが認識できません。
8255	0x0000203F	ERROR_DS_PARAM_ERROR		1つ以上のパラメータが無効です。
8256	0x00002040	ERROR_DS_NOT_SUPPORTED		指定された方法がサポートされていません。

エラー			説明
10進数	16進数	名前	
8257	0x00002041	ERROR_DS_NO_RESULTS_RETURNED	結果が返されませんでした。
8258	0x00002042	ERROR_DS_CONTROL_NOT_FOUND	指定された制御が、サーバではサポートされていません。
8259	0x00002043	ERROR_DS_CLIENT_LOOP	照会ループがクライアントによって検出されました。
8260	0x00002044	ERROR_DS_REFERRAL_LIMIT_EXCEEDED	事前設定された照会制限を超えました。
8261	0x00002045	ERROR_DS_SORT_CONTROL_MISSING	検索にはSORT制御が必要です。
8262	0x00002046	ERROR_DS_OFFSET_RANGE_ERROR	検索結果が指定されたオフセット範囲を超えています。
8301	0x0000206D	ERROR_DS_ROOT_MUST_BE_NC	ルートオブジェクトは、名前付けコンテキストのヘッドに配置する必要があります。ルートオブジェクトはインスタンス化された親を持つことはできません。
8302	0x0000206E	ERROR_DS_ADD_REPLICA_INHIBITED	複製追加操作は実行できません。名前付けコンテキストは、複製を作成するために書き込み可能である必要があります。
8303	0x0000206F	ERROR_DS_ATT_NOT_DEF_IN_SCHEMA	スキーマに定義されていない属性への参照が発生しました。
8304	0x00002070	ERROR_DS_MAX_OBJ_SIZE_EXCEEDED	オブジェクトの最大サイズを超えました。
8305	0x00002071	ERROR_DS_OBJ_STRING_NAME_EXISTS	既に使用されている名前でもオブジェクトをディレクトリに追加しようとしました。
8306	0x00002072	ERROR_DS_NO_RDN_DEFINED_IN_SCHEMA	スキーマに定義されたRDNを持っていないクラスのオブジェクトを追加しようとしました。
8307	0x00002073	ERROR_DS_RDN_DOESNT_MATCH_SCHEMA	スキーマに定義されたRDNではないRDNを使用してオブジェクトを追加しようとしました。
8308	0x00002074	ERROR_DS_NO_REQUESTED_ATTRS_FOUND	要求された属性がオブジェクトに何も見つかりませんでした。
8309	0x00002075	ERROR_DS_USER_BUFFER_TOO_SMALL	ユーザバッファが小さすぎます。
8310	0x00002076	ERROR_DS_ATT_IS_NOT_ON_OBJ	操作で指定された属性がオブジェクトに存在しません。
8311	0x00002077	ERROR_DS_ILLEGAL_MOD_OPERATION	この変更操作は無効です。一部の変更は許可されていません。
8312	0x00002078	ERROR_DS_OBJ_TOO_LARGE	指定されたオブジェクトが大きすぎます。
8313	0x00002079	ERROR_DS_BAD_INSTANCE_TYPE	指定されたインスタンスタイプが有効ではありません。
8314	0x0000207A	ERROR_DS_MASTERDSA_REQUIRED	この操作はマスタDSAで実行する必要があります。
8315	0x0000207B	ERROR_DS_OBJECT_CLASS_REQUIRED	オブジェクトのクラス属性を指定する必要があります。
8316	0x0000207C	ERROR_DS_MISSING_REQUIRED_ATT	必要な属性が見つかりません。
8317	0x0000207D	ERROR_DS_ATT_NOT_DEF_FOR_CLASS	オブジェクトを変更してそのクラスに許可されていない属性を含めようとしました。
8318	0x0000207E	ERROR_DS_ATT_ALREADY_EXISTS	指定された属性は既にオブジェクトに存在しています。
8320	0x00002080	ERROR_DS_CANT_ADD_ATT_VALUES	指定された属性が存在しないか、または値がありません。
8321	0x00002081	ERROR_DS_SINGLE_VALUE_CONSTRAINT	1つの値のみを持つことができる属性に複数の値が指定されています。
8322	0x00002082	ERROR_DS_RANGE_CONSTRAINT	属性の値が値の許容範囲を超えています。
8323	0x00002083	ERROR_DS_ATT_VAL_ALREADY_EXISTS	指定された値は既に存在しています。
8324	0x00002084	ERROR_DS_CANT_REM_MISSING_ATT	その属性がオブジェクトに存在しないため、属性を削除できません。
8325	0x00002085	ERROR_DS_CANT_REM_MISSING_ATT_VAL	その属性値がオブジェクトに存在しないため、属性値を削除できません。
8326	0x00002086	ERROR_DS_ROOT_CANT_BE_SUBREF	指定されたルートオブジェクトはsubrefになることができません。
8327	0x00002087	ERROR_DS_NO_CHAINING	連鎖は許可されていません。
8328	0x00002088	ERROR_DS_NO_CHAINED_EVAL	連鎖した評価は許可されていません。
8329	0x00002089	ERROR_DS_NO_PARENT_OBJECT	オブジェクトの親がインスタンス化されていない、または削除されているため、操作が実行できませんでした。

エラー			説明
10進数	16進数	名前	
8330	0x0000208A	ERROR_DS_PARENT_IS_AN_ALIAS	エイリアスである親を持つことは許可されていません。エイリアスはリーフオブジェクトです。
8331	0x0000208B	ERROR_DS_CANT_MIX_MASTER_AND_REPS	オブジェクトと親は同じタイプである必要があり、両方がマスタまたは両方が複製である必要があります。
8332	0x0000208C	ERROR_DS_CHILDREN_EXIST	子オブジェクトが存在するため、操作が実行できません。この操作は、リーフオブジェクトでのみ実行できます。
8333	0x0000208D	ERROR_DS_OBJ_NOT_FOUND	ディレクトリオブジェクトが見つかりません。
8334	0x0000208E	ERROR_DS_ALIASSED_OBJ_MISSING	エイリアスを持つオブジェクトが見つかりません。
8335	0x0000208F	ERROR_DS_BAD_NAME_SYNTAX	オブジェクト名の構文が間違っています。
8336	0x00002090	ERROR_DS_ALIAS_POINTS_TO_ALIAS	エイリアスが他のエイリアスを参照することは許可されていません。
8337	0x00002091	ERROR_DS_CANT_DEREF_ALIAS	エイリアスは逆参照できません。
8338	0x00002092	ERROR_DS_OUT_OF_SCOPE	操作がスコープを外れています。
8339	0x00002093	ERROR_DS_OBJECT_BEING_REMOVED	オブジェクトが削除中のため、操作を継続できません。
8340	0x00002094	ERROR_DS_CANT_DELETE_DSA_OBJ	DSAオブジェクトは削除できません。
8341	0x00002095	ERROR_DS_GENERIC_ERROR	ディレクトリサービスエラーが発生しました。
8342	0x00002096	ERROR_DS_DSA_MUST_BE_INT_MASTER	この操作は、内部マスタDSAオブジェクトでのみ実行できます。
8343	0x00002097	ERROR_DS_CLASS_NOT_DSA	このオブジェクトはクラスDSAである必要があります。
8344	0x00002098	ERROR_DS_INSUFF_ACCESS_RIGHTS	操作を実行するにはアクセス権が不足しています。
8345	0x00002099	ERROR_DS_ILLEGAL_SUPERIOR	親が可能な上位リストに含まれていないため、オブジェクトが追加できません。
8346	0x0000209A	ERROR_DS_ATTRIBUTE_OWNED_BY_SAM	この属性はSecurity Accounts Manager (SAM)によって所有されているため、属性へのアクセスが許可されていません。
8347	0x0000209B	ERROR_DS_NAME_TOO_MANY_PARTS	名前の部分が多すぎます。
8348	0x0000209C	ERROR_DS_NAME_TOO_LONG	名前が長すぎます。
8349	0x0000209D	ERROR_DS_NAME_VALUE_TOO_LONG	名前の値が長すぎます。
8350	0x0000209E	ERROR_DS_NAME_UNPARSEABLE	名前の解析時にディレクトリサービスにエラーが発生しました。
8351	0x0000209F	ERROR_DS_NAME_TYPE_UNKNOWN	ディレクトリサービスが名前の属性タイプを取得できません。
8352	0x000020A0	ERROR_DS_NOT_AN_OBJECT	この名前はオブジェクトを識別しません。(the name identifies a phantom.)
8353	0x000020A1	ERROR_DS_SEC_DESC_TOO_SHORT	セキュリティ記述子が短すぎます。
8354	0x000020A2	ERROR_DS_SEC_DESC_INVALID	セキュリティ記述子が無効です。
8355	0x000020A3	ERROR_DS_NO_DELETED_NAME	削除したオブジェクトの名前の作成に失敗しました。
8356	0x000020A4	ERROR_DS_SUBREF_MUST_HAVE_PARENT	新しいsubrefの親が存在する必要があります。
8357	0x000020A5	ERROR_DS_NCNAME_MUST_BE_NC	このオブジェクトは名前付けコンテキストである必要があります。
8358	0x000020A6	ERROR_DS_CANT_ADD_SYSTEM_ONLY	システムによって所有されている属性を追加することは許可されていません。
8359	0x000020A7	ERROR_DS_CLASS_MUST_BE_CONCRETE	オブジェクトのクラスは構造的である必要があります。抽象的なクラスをインスタンス化することはできません。
8360	0x000020A8	ERROR_DS_INVALID_DMD	スキーマオブジェクトが見つかりませんでした。
8361	0x000020A9	ERROR_DS_OBJ_GUID_EXISTS	このGUID (deadまたはalive)を持つローカルオブジェクトは既に存在しています。
8362	0x000020AA	ERROR_DS_NOT_ON_BACKLINK	この操作はバックリンクでは実行できません。
8363	0x000020AB	ERROR_DS_NO_CROSSREF_FOR_NC	指定された名前付けコンテキストに対するクロス参照が見つかりませんでした。
8364	0x000020AC	ERROR_DS_SHUTTING_DOWN	ディレクトリサービスがシャットダウン中であるため、操作が実行できませんでした。
8365	0x000020AD	ERROR_DS_UNKNOWN_OPERATION	ディレクトリサービスリクエストが無効です。

		エラー	説明
10進数	16進数	名前	
8366	0x000020AE	ERROR_DS_INVALID_ROLE_OWNER	役割所有者の属性を読み出すことができませんでした。
8367	0x000020AF	ERROR_DS_COULDNT_CONTACT_FSMO	要求されたFSMO操作に失敗しました。現在のFSMOホルダに到達できませんでした。
8368	0x000020B0	ERROR_DS_CROSS_NC_DN_RENAME	名前付けコンテキストでのDNの変更は許可されていません。
8369	0x000020B1	ERROR_DS_CANT_MOD_SYSTEM_ONLY	この属性はシステムによって所有されているため変更できません。
8370	0x000020B2	ERROR_DS_REPLICATOR_ONLY	レプリケータのみがこの機能を実行できます。
8371	0x000020B3	ERROR_DS_OBJ_CLASS_NOT_DEFINED	指定されたクラスは定義されていません。
8372	0x000020B4	ERROR_DS_OBJ_CLASS_NOT_SUBCLASS	指定されたクラスはサブクラスではありません。
8373	0x000020B5	ERROR_DS_NAME_REFERENCE_INVALID	名前の参照が無効です。
8374	0x000020B6	ERROR_DS_CROSS_REF_EXISTS	クロス参照は既に存在しています。
8375	0x000020B7	ERROR_DS_CANT_DEL_MASTER_CROSSREF	マスタのクロス参照を削除することは許可されていません。
8376	0x000020B8	ERROR_DS_SUBTREE_NOTIFY_NOT_NC_HEAD	サブツリー通知は、NCヘッドでのみサポートされています。
8377	0x000020B9	ERROR_DS_NOTIFY_FILTER_TOO_COMPLEX	通知フィルタが複雑すぎます。
8378	0x000020BA	ERROR_DS_DUP_RDN	スキーマ更新の失敗：RDNの重複。
8379	0x000020BB	ERROR_DS_DUP_OID	スキーマ更新の失敗：OIDの重複。
8380	0x000020BC	ERROR_DS_DUP_MAPI_ID	スキーマ更新の失敗：MAPI IDの重複。
8381	0x000020BD	ERROR_DS_DUP_SCHEMA_ID_GUID	スキーマ更新の失敗：スキーマID GUIDの重複。
8382	0x000020BE	ERROR_DS_DUP_LDAP_DISPLAY_NAME	スキーマ更新の失敗：LDAP表示名の重複。
8383	0x000020BF	ERROR_DS_SEMANTIC_ATT_TEST	スキーマ更新の失敗：範囲の下限値を上限値より小さくする
8384	0x000020C0	ERROR_DS_SYNTAX_MISMATCH	スキーマ更新の失敗：構文の不一致
8385	0x000020C1	ERROR_DS_EXISTS_IN_MUST_HAVE	スキーマ削除の失敗：属性がmust-containで使用されている
8386	0x000020C2	ERROR_DS_EXISTS_IN_MAY_HAVE	スキーマ削除の失敗：属性がmay-containで使用されている
8387	0x000020C3	ERROR_DS_NONEXISTENT_MAY_HAVE	スキーマ更新の失敗：may-containに属性が存在しない
8388	0x000020C4	ERROR_DS_NONEXISTENT_MUST_HAVE	スキーマ更新の失敗：must-containに属性が存在しない
8389	0x000020C5	ERROR_DS_AUX_CLS_TEST_FAIL	スキーマ更新の失敗：aux-classリストのクラスが存在しない、または補助クラスではない
8390	0x000020C6	ERROR_DS_NONEXISTENT_POSS_SUP	スキーマ更新の失敗：poss-superiorsにクラスが存在しない
8391	0x000020C7	ERROR_DS_SUB_CLS_TEST_FAIL	スキーマ更新の失敗：subclassofリストのクラスが存在しない、または階層規則に従っていない
8392	0x000020C8	ERROR_DS_BAD_RDN_ATT_ID_SYNTAX	スキーマ更新の失敗：Rdn-Att-Idの構文が間違っている
8393	0x000020C9	ERROR_DS_EXISTS_IN_AUX_CLS	スキーマ削除の失敗：クラスがauxiliary classとして使用されている
8394	0x000020CA	ERROR_DS_EXISTS_IN_SUB_CLS	スキーマ削除の失敗：クラスがsub classとして使用されている
8395	0x000020CB	ERROR_DS_EXISTS_IN_POSS_SUP	スキーマ削除の失敗：クラスがposs superiorとして使用されている
8396	0x000020CC	ERROR_DS_RECALCSHEMA_FAILED	検証キャッシュの再計算でスキーマ更新に失敗しました。
8397	0x000020CD	ERROR_DS_TREE_DELETE_NOT_FINISHED	ツリー削除が終了していません。
8398	0x000020CE	ERROR_DS_CANT_DELETE	要求された削除操作を実行できませんでした。
8399	0x000020CF	ERROR_DS_ATT_SCHEMA_REQ_ID	スキーマレコードのgovernsクラスIDを読み出すことができません。
8400	0x000020D0	ERROR_DS_BAD_ATT_SCHEMA_SYNTAX	属性スキーマの構文が間違っています。
8401	0x000020D1	ERROR_DS_CANT_CACHE_ATT	属性をキャッシュできませんでした。
8402	0x000020D2	ERROR_DS_CANT_CACHE_CLASS	クラスをキャッシュできませんでした。
8403	0x000020D3	ERROR_DS_CANT_REMOVE_ATT_CACHE	属性をキャッシュから削除できませんでした。
8404	0x000020D4	ERROR_DS_CANT_REMOVE_CLASS_CACHE	クラスをキャッシュから削除できませんでした。

		エラー		説明
10進数	16進数	名前		
8405	0x000020D5	ERROR_DS_CANT_RETRIEVE_DN		識別された名前属性を読み出すことができませんでした。
8406	0x000020D6	ERROR_DS_MISSING_SUPREF		必要なsubrefが見つかりません。
8407	0x000020D7	ERROR_DS_CANT_RETRIEVE_INSTANCE		インスタンスタイプの属性を読み出すことができませんでした。
8408	0x000020D8	ERROR_DS_CODE_INCONSISTENCY		内部エラーが発生しました。
8409	0x000020D9	ERROR_DS_DATABASE_ERROR		データベースエラーが発生しました。
8410	0x000020DA	ERROR_DS_GOVERNSID_MISSING		属性GOVERNSIDが見つかりません。
8411	0x000020DB	ERROR_DS_MISSING_EXPECTED_ATT		所期の属性が見つかりません。
8412	0x000020DC	ERROR_DS_NCNAME_MISSING_CR_REF		指定された名前付けコンテキストにクロス参照が見つかりません。
8413	0x000020DD	ERROR_DS_SECURITY_CHECKING_ERROR		セキュリティチェックエラーが発生しました。
8414	0x000020DE	ERROR_DS_SCHEMA_NOT_LOADED		スキーマがロードされていません。
8415	0x000020DF	ERROR_DS_SCHEMA_ALLOC_FAILED		スキーマ割り当てに失敗しました。マシンのメモリが不足していないかチェックしてください。
8416	0x000020E0	ERROR_DS_ATT_SCHEMA_REQ_SYNTAX		属性スキーマに必要な構文の取得に失敗しました。
8417	0x000020E1	ERROR_DS_GCVERIFY_ERROR		グローバルカタログの検証に失敗しました。グローバルカタログが使用できないか、またはこの操作をサポートしていません。ディレクトリの一部が現在使用できません。
8418	0x000020E2	ERROR_DS_DRA_SCHEMA_MISMATCH		関連するサーバ間のスキーマ不一致のため、複製操作が失敗しました。
8419	0x000020E3	ERROR_DS_CANT_FIND_DSA_OBJ		DSAオブジェクトが見つかりませんでした。
8420	0x000020E4	ERROR_DS_CANT_FIND_EXPECTED_NC		名前付けコンテキストが見つかりませんでした。
8421	0x000020E5	ERROR_DS_CANT_FIND_NC_IN_CACHE		名前付けコンテキストがキャッシュ内に見つかりませんでした。
8422	0x000020E6	ERROR_DS_CANT_RETRIEVE_CHILD		子オブジェクトが見つかりませんでした。
8423	0x000020E7	ERROR_DS_SECURITY_ILLEGAL_MODIFY		セキュリティ上の理由からこの変更は許可されていません。
8424	0x000020E8	ERROR_DS_CANT_REPLACE_HIDDEN_REC		この操作は非表示のレコードを置き換えることができません。
8425	0x000020E9	ERROR_DS_BAD_HIERARCHY_FILE		階層ファイルが無効です。
8426	0x000020EA	ERROR_DS_BUILD_HIERARCHY_TABLE_FAILED		階層テーブルの構築に失敗しました。
8427	0x000020EB	ERROR_DS_CONFIG_PARAM_MISSING		ディレクトリ設定パラメータがレジストリに見つかりません。
8428	0x000020EC	ERROR_DS_COUNTING_AB_INDICES_FAILED		アドレス帳のインデックスのカウン트에失敗しました。
8429	0x000020ED	ERROR_DS_HIERARCHY_TABLE_MALLOC_FAILED		階層テーブルの割り当てに失敗しました。
8430	0x000020EE	ERROR_DS_INTERNAL_FAILURE		ディレクトリサービスに内部エラーが発生しました。
8431	0x000020EF	ERROR_DS_UNKNOWN_ERROR		ディレクトリサービスに不明なエラーが発生しました。
8432	0x000020F0	ERROR_DS_ROOT_REQUIRES_CLASS_TOP		ルートオブジェクトには「最上位」のクラスが必要です。
8433	0x000020F1	ERROR_DS_REFUSING_FSMO_ROLES		このディレクトリサーバがシャットダウン中で、新しい浮動小数点数シングルマスタ操作役割の所有権を取得できません。
8434	0x000020F2	ERROR_DS_MISSING_FSMO_SETTINGS		ディレクトリサービスに必須の構成情報を見つからず、浮動小数点数シングルマスタ操作役割の所有権を決定できません。
8435	0x000020F3	ERROR_DS_UNABLE_TO_SURRENDER_ROLES		ディレクトリサービスは、1つ以上の浮動小数点数シングルマスタ操作役割の所有権を他のサーバに転送できませんでした。
8436	0x000020F4	ERROR_DS_DRA_GENERIC		複製操作に失敗しました。
8437	0x000020F5	ERROR_DS_DRA_INVALID_PARAMETER		無効なパラメータがこの複製操作に指定されています。
8438	0x000020F6	ERROR_DS_DRA_BUSY		ディレクトリサービスがビジーすぎるため、この時点で複製操作を完了できません。
8439	0x000020F7	ERROR_DS_DRA_BAD_DN		この複製操作に指定された識別名が無効です。

エラー			説明
10進数	16進数	名前	
8440	0x000020F8	ERROR_DS_DRA_BAD_NC	この複製操作に関する名前付けコンテキストが無効です。
8441	0x000020F9	ERROR_DS_DRA_DN_EXISTS	この複製操作に指定された識別名は既に存在しています。
8442	0x000020FA	ERROR_DS_DRA_INTERNAL_ERROR	複製システムで内部エラーが発生しました。
8443	0x000020FB	ERROR_DS_DRA_INCONSISTENT_DIT	複製操作でデータベースの不整合が発生しました。
8444	0x000020FC	ERROR_DS_DRA_CONNECTION_FAILED	この複製操作に指定されたサーバに接続できませんでした。
8445	0x000020FD	ERROR_DS_DRA_BAD_INSTANCE_TYPE	複製操作で無効なインスタンスタイプのオブジェクトが見つかりました。
8446	0x000020FE	ERROR_DS_DRA_OUT_OF_MEM	複製操作がメモリの割り当てに失敗しました。
8447	0x000020FF	ERROR_DS_DRA_MAIL_PROBLEM	複製操作でメールシステムにエラーが発生しました。
8448	0x00002100	ERROR_DS_DRA_REF_ALREADY_EXISTS	ターゲットサーバに関する複製参照情報は既に存在しています。
8449	0x00002101	ERROR_DS_DRA_REF_NOT_FOUND	ターゲットサーバに関する複製参照情報が存在しません。
8450	0x00002102	ERROR_DS_DRA_OBJ_IS_REP_SOURCE	別のサーバに複製されているため、名前付けコンテキストが削除できません。
8451	0x00002103	ERROR_DS_DRA_DB_ERROR	複製操作でデータベースエラーが発生しました。
8452	0x00002104	ERROR_DS_DRA_NO_REPLICA	名前付けコンテキストが削除中であるか、または指定されたサーバから複製されていません。
8453	0x00002105	ERROR_DS_DRA_ACCESS_DENIED	複製アクセスが拒否されました。
8454	0x00002106	ERROR_DS_DRA_NOT_SUPPORTED	要求された操作が、このバージョンのディレクトリサービスではサポートされていません。
8455	0x00002107	ERROR_DS_DRA_RPC_CANCELLED	複製リモートプロシージャ呼び出しがキャンセルされました。
8456	0x00002108	ERROR_DS_DRA_SOURCE_DISABLED	ソースサーバが現在、複製リクエストを拒否しています。
8457	0x00002109	ERROR_DS_DRA_SINK_DISABLED	デスティネーションサーバが現在、複製リクエストを拒否しています。
8458	0x0000210A	ERROR_DS_DRA_NAME_COLLISION	オブジェクト名の競合のため、複製操作に失敗しました。
8459	0x0000210B	ERROR_DS_DRA_SOURCE_REINSTALLED	複製ソースが再インストールされました。
8460	0x0000210C	ERROR_DS_DRA_MISSING_PARENT	必要な親オブジェクトが見つからないため、複製操作が失敗しました。
8461	0x0000210D	ERROR_DS_DRA_PREEMPTED	複製操作が他のプロセスによって中断されました。
8462	0x0000210E	ERROR_DS_DRA_ABANDON_SYNC	アップデートの欠如のため、複製同期の試行が破棄されました。
8463	0x0000210F	ERROR_DS_DRA_SHUTDOWN	システムがシャットダウン中のため、複製操作が終了しました。
8464	0x00002110	ERROR_DS_DRA_INCOMPATIBLE_PARTIAL_SET	デスティネーションの部分属性セットがソースの部分属性セットのサブセットではないため、複製同期に失敗しました。
8465	0x00002111	ERROR_DS_DRA_SOURCE_IS_PARTIAL_REPLICA	マスタ複製が部分複製から同期しようとしたため、複製同期に失敗しました。
8466	0x00002112	ERROR_DS_DRA_EXTN_CONNECTION_FAILED	この複製操作に指定されたサーバには接続しましたが、このサーバが操作を完了するために必要な別のサーバに接続できませんでした。
8467	0x00002113	ERROR_DS_INSTALL_SCHEMA_MISMATCH	ソースフォレストのアクティブディレクトリスキーマのバージョンが、このコンピュータアクティブディレクトリのバージョンと互換性がありません。このコンピュータをドメインコントローラとしてソースフォレストに追加する前に、ソースフォレストでドメインコントローラのOSをアップグレードする必要があります。
8468	0x00002114	ERROR_DS_DUP_LINK_ID	スキーマ更新の失敗: 同じリンクIDを持つ属性は既に存在しています。
8469	0x00002115	ERROR_DS_NAME_ERROR_RESOLVING	名前変換: 一般的な処理エラー。
8470	0x00002116	ERROR_DS_NAME_ERROR_NOT_FOUND	名前変換: 名前が見つからない、または名前を確認するための権限が不足しています。

エラー			説明
10進数	16進数	名前	
8471	0x00002117	ERROR_DS_NAME_ERROR_NOT_UNIQUE	名前変換: 入力名が複数の出力名にマッピングされています。
8472	0x00002118	ERROR_DS_NAME_ERROR_NO_MAPPING	名前変換: 入力名は見つかりましたが、関連する出力形式が見つかりませんでした。
8473	0x00002119	ERROR_DS_NAME_ERROR_DOMAIN_ONLY	名前変換: 完全に解決することができず、ドメインのみが見つかりました。
8474	0x0000211A	ERROR_DS_NAME_ERROR_NO_SYNTACTICAL_MAPPING	名前変換: 通信を行わずに、純粋に構文的なマッピングをクライアントで実行できません。
8475	0x0000211B	ERROR_DS_CONSTRUCTED_ATT_MOD	構築されたattの変更は許可されていません。
8476	0x0000211C	ERROR_DS_WRONG_OM_OBJ_CLASS	指定された構文を持つ属性に対して、指定されたOMオブジェクトクラスが正しくありません。
8477	0x0000211D	ERROR_DS_DRA_REPL_PENDING	複製リクエストがPOST送信され、応答を待っています。
8478	0x0000211E	ERROR_DS_DS_REQUIRED	要求された操作にはディレクトリサービスが必要ですが、利用できませんでした。
8479	0x0000211F	ERROR_DS_INVALID_LDAP_DISPLAY_NAME	クラスまたは属性のLDAP表示名に非ASCII文字が含まれています。
8480	0x00002120	ERROR_DS_NON_BASE_SEARCH	要求された検索操作では基本検索機能のみがサポートされています。
8481	0x00002121	ERROR_DS_CANT_RETRIEVE_ATTS	検索がデータベースからの属性の読み出しに失敗しました。
8482	0x00002122	ERROR_DS_BACKLINK_WITHOUT_LINK	スキーマ更新操作が、対応するフォワードリンクがないバックワードリンク属性を追加しようとしました。
8483	0x00002123	ERROR_DS_EPOCH_MISMATCH	クロスドメイン移動のソースとデスティネーションの間でオブジェクトのエPOCH番号が一致していません。ソースまたはデスティネーションのいずれかが、オブジェクトの最新バージョンを持っていません。
8484	0x00002124	ERROR_DS_SRC_NAME_MISMATCH	クロスドメイン移動のソースとデスティネーションの間でオブジェクトの現在の名前が一致していません。ソースまたはデスティネーションのいずれかが、オブジェクトの最新バージョンを持っていません。
8485	0x00002125	ERROR_DS_SRC_AND_DST_NC_IDENTICAL	クロスドメイン移動操作のソースとデスティネーションが同一です。呼び出し側は、クロスドメイン移動操作ではなく、ローカル移動操作を使用する必要があります。
8486	0x00002126	ERROR_DS_DST_NC_MISMATCH	クロスドメイン移動のソースとデスティネーションは、フォレストの名前付けコンテキストが一致していません。ソースまたはデスティネーションのいずれかが、Partitionsコンテナの最新バージョンを持っていません。
8487	0x00002127	ERROR_DS_NOT_AUTHORITY_FOR_DST_NC	クロスドメイン移動のデスティネーションは、デスティネーションの名前付けコンテキストに関するコンテンツ情報を持っていません。
8488	0x00002128	ERROR_DS_SRC_GUID_MISMATCH	クロスドメイン移動のソースとデスティネーションの間でソースオブジェクトのIDが一致していません。ソースまたはデスティネーションのいずれかが、ソースオブジェクトの最新バージョンを持っていません。
8489	0x00002129	ERROR_DS_CANT_MOVE_DELETED_OBJECT	ドメイン間を移動するオブジェクトは、デスティネーションサーバによって削除されることが既に知られています。ソースサーバが、ソースオブジェクトの最新バージョンを持っていません。
8490	0x0000212A	ERROR_DS_PDC_OPERATION_IN_PROGRESS	PDC PSMOへの排他的アクセスが必要な他の操作が既に進行中です。
8491	0x0000212B	ERROR_DS_CROSS_DOMAIN_CLEANUP_REQD	クロスドメイン移動操作が失敗し、ソースドメインとデスティネーションドメインにそれぞれ1つずつ、移動したオブジェクトの2つのバージョンが存在しています。デスティネーションオブジェクトを削除し、一貫性のある状態にシステムを復元する必要があります。
8492	0x0000212C	ERROR_DS_ILLEGAL_XDOM_MOVE_OPERATION	このクラスのクロスドメイン移動が許可されていないか、またはオブジェクトに特殊文字が含まれているため、このオブジェクトはドメインの境界

エラー			説明
10進数	16進数	名前	
			を越えて移動できません(例えば、トラストアカウントや制限付きのRIDによって移動ができません)。
8493	0x0000212D	ERROR_DS_CANT_WITH_ACCT_GROUP_MEMBERSHPS	メンバーシップがドメイン境界を越えるオブジェクトは移動できません。移動すれば、アカウントグループのメンバーシップ条件の違反になります。アカウントグループのメンバーシップからオブジェクトを削除し、再試行してください。
8494	0x0000212E	ERROR_DS_NC_MUST_HAVE_NC_PARENT	名前付けコンテキストのヘッドは、内部ノードではなく、別の名前付けコンテキストのヘッドの直接の子である必要があります。
8495	0x0000212F	ERROR_DS_CR_IMPOSSIBLE_TO_VALIDATE	ディレクトリが提示された名前付けコンテキストの上に名前付けコンテキストの複製を保持していないため、提示された名前付けコンテキストを検証できません。ドメインの名前付けマスタの役割が、後ローバルカタログサーバとして構成されたサーバによって保持されていること、およびこのサーバがその複製パートナーに対して最新の状態であることを確認してください。(Windows 2000のドメイン名前付けマスタにのみ適用されます)
8496	0x00002130	ERROR_DS_DST_DOMAIN_NOT_NATIVE	デスティネーションドメインはネイティブモードである必要があります。
8497	0x00002131	ERROR_DS_MISSING_INFRASTRUCTURE_CONTAINER	サーバが関連するドメインのインフラストラクチャコンテナを持っていないため、操作を実行できません。
8498	0x00002132	ERROR_DS_CANT_MOVE_ACCOUNT_GROUP	空でないアカウントグループのクロスドメイン移動は許可されていません。
8499	0x00002133	ERROR_DS_CANT_MOVE_RESOURCE_GROUP	空でないリソースグループのクロスドメイン移動は許可されていません。

エラー			説明
10進数	16進数	名前	
8500	0x00002134	ERROR_DS_INVALID_SEARCH_FLAG	この属性の検索フラグは無効です。ANRビットは、UnicodeまたはTeletex文字列の属性についてのみ有効です。
8501	0x00002135	ERROR_DS_NO_TREE_DELETE_ABOVE_NC	NCヘッドを子孫として持つオブジェクトからツリー削除を開始することは許可されていません。
8502	0x00002136	ERROR_DS_COULDNT_LOCK_TREE_FOR_DELETE	ツリーが使用中であるため、ディレクトリサービスがツリーの削除を準備するためにツリーをロックすることができませんでした。
8503	0x00002137	ERROR_DS_COULDNT_IDENTIFY_OBJECTS_FOR_TREE_DELETE	ディレクトリサービスがツリーを削除しようとするときに、オブジェクトのリストを識別することができませんでした。
8504	0x00002138	ERROR_DS_SAM_INIT_FAILURE	以下のエラーのために、Security Accounts Managerの初期化に失敗しました: %1. エラーステータス: 0x%2. OKをクリックしてシステムをシャットダウンし、Directory Services Restoreモードで再起動してください。詳細情報をイベントログでチェックしてください。
8505	0x00002139	ERROR_DS_SENSITIVE_GROUP_VIOLATION	管理者のみが管理者グループのメンバーシップリストを変更できます。
8506	0x0000213A	ERROR_DS_CANT_MOD_PRIMARYGROUPID	ドメインコントローラアカウントのプライマリグループIDを変更できません。
8507	0x0000213B	ERROR_DS_ILLEGAL_BASE_SCHEMA_MOD	ベーススキーマを変更しようとしてしました。
8508	0x0000213C	ERROR_DS_NONSAFE_SCHEMA_CHANGE	新しいメンバーシップ属性を既存のクラスに追加すること、既存のクラスから必須属性を削除すること、またはオプション属性を(直接または継承を通じて、例えば、補助クラスを追加または削除することによって)バックリンク属性を持っていない特殊クラスTopに追加することは許可されていません。
8509	0x0000213D	ERROR_DS_SCHEMA_UPDATE_DISALLOWED	このDCはスキーマFSMO Role Ownerではないため、スキーマの更新はこのDCでは許可されていません。

エラー			説明
10進数	16進数	名前	
8510	0x0000213E	ERROR_DS_CANT_CREATE_UNDER_SCHEMA	このクラスのオブジェクトは、スキーマコンテナの下で作成できません。スキーマコンテナの下では属性スキーマおよびクラススキーマオブジェクトのみを作成できます。
8511	0x0000213F	ERROR_DS_INSTALL_NO_SRC_SCH_VERSION	複製/子のインストールが、ソースDCのスキーマコンテナのobjectVersion属性の取得に失敗しました。スキーマコンテナにこの属性が見つからないか、または提供された資格情報がこの属性を読み出すための権限を持っていないかのいずれかです。
8512	0x00002140	ERROR_DS_INSTALL_NO_SCH_VERSION_IN_INIFILE	複製/子のインストールが、system32ディレクトリのschema.iniファイルのSCHEMAセクションでobjectVersion属性の読み出しに失敗しました。
8513	0x00002141	ERROR_DS_INVALID_GROUP_TYPE	指定されたグループタイプが無効です。
8514	0x00002142	ERROR_DS_NO_NEST_GLOBALGROUP_IN_MIXEDDOMAIN	グループのセキュリティが有効になっている場合、混合ドメインでグローバルグループをネストできません。
8515	0x00002143	ERROR_DS_NO_NEST_LOCALGROUP_IN_MIXEDDOMAIN	グループのセキュリティが有効になっている場合、混合ドメインでローカルグループをネストできません。
8516	0x00002144	ERROR_DS_GLOBAL_CANT_HAVE_LOCAL_MEMBER	グローバルグループは、ローカルグループをメンバにすることはできません。
8517	0x00002145	ERROR_DS_GLOBAL_CANT_HAVE_UNIVERSAL_MEMBER	グローバルグループは、ユニバーサルグループをメンバにすることはできません。
8518	0x00002146	ERROR_DS_UNIVERSAL_CANT_HAVE_LOCAL_MEMBER	ユニバーサルグループは、ローカルグループをメンバにすることはできません。
8519	0x00002147	ERROR_DS_GLOBAL_CANT_HAVE_CROSSDOMAIN_MEMBER	グローバルグループは、クロスドメインメンバをメンバにすることはできません。
8520	0x00002148	ERROR_DS_LOCAL_CANT_HAVE_CROSSDOMAIN_LOCAL_MEMBER	ローカルグループは、別のクロスドメインのローカルグループをメンバにすることはできません。
8521	0x00002149	ERROR_DS_HAVE_PRIMARY_MEMBERS	プライメンバを持つグループは、セキュリティが無効になったグループに変更できません。
8522	0x0000214A	ERROR_DS_STRING_SD_CONVERSION_FAILED	スキーマキャッシュロードが、クロススキーマオブジェクトで文字列デフォルトSDを変換できませんでした。
8523	0x0000214B	ERROR_DS_NAMING_MASTER_GC	Global Catalogサーバとして構成されたDSAのみが、Domain Naming Master FSMOの役割を許可されます。(Windows 2000サーバにのみ適用されます)
8524	0x0000214C	ERROR_DS_LOOKUP_FAILURE	DNSルックアップエラーのため、DSA操作を開始できませんでした。
8525	0x0000214D	ERROR_DS_COULDNT_UPDATE_SPNS	オブジェクトのDNS Host Nameの変更を処理する時に、Service Principal Nameの値が同期を継続することができませんでした。
8526	0x0000214E	ERROR_DS_CANT_RETRIEVE_SD	Security Descriptor属性を読み出すことができませんでした。
8527	0x0000214F	ERROR_DS_KEY_NOT_UNIQUE.	要求されたオブジェクトは見つかりませんでしたが、このキーを持つオブジェクトは見つかりました。
8528	0x00002150	ERROR_DS_WRONG_LINKED_ATT_SYNTAX	追加されるリンクされた属性の構文が間違っています。フォワードリンクは構文2.5.5.1、2.5.5.7、および2.5.5.14のみを持つことができ、バックリンクは構文2.5.5.1のみを持つことができます。
8529	0x00002151	ERROR_DS_SAM_NEED_BOOTKEY_PASSWORD	Security Account Managerがブートパスワードを取得する必要があります。
8530	0x00002152	ERROR_DS_SAM_NEED_BOOTKEY_FLOPPY	Security Account Managerがフロッピーディスクからブートキーを取得する必要があります。
8531	0x00002153	ERROR_DS_CANT_START	ディレクトリサービスを開始できません。
8532	0x00002154	ERROR_DS_INIT_FAILURE	ディレクトリサービスを開始できませんでした。
8533	0x00002155	ERROR_DS_NO_PKT_PRIVACY_ON_CONNECTION	クライアントとサーバ間の接続にはパケットプライバシー以上が必要です。
8534	0x00002156	ERROR_DS_SOURCE_DOMAIN_IN_FOREST	ソースドメインがドメインネーションと同じフォレスト内にない可能性があります。

エラー			説明
10進数	16進数	名前	
8535	0x00002157	ERROR_DS_DESTINATION_DOMAIN_NOT_IN_FOREST	デスティネーションドメインはフォレスト内にいる必要があります。
8536	0x00002158	ERROR_DS_DESTINATION_AUDITING_NOT_ENABLED	この操作にはデスティネーションドメインの監査機能が有効になっている必要があります。
8537	0x00002159	ERROR_DS_CANT_FIND_DC_FOR_SRC_DOMAIN	操作がソースドメインのDCを特定できませんでした。
8538	0x0000215A	ERROR_DS_SRC_OBJ_NOT_GROUP_OR_USER	ソースオブジェクトはグループまたはユーザである必要があります。
8539	0x0000215B	ERROR_DS_SRC_SID_EXISTS_IN_FOREST	ソースオブジェクトのSIDは既にデスティネーションフォレストに存在します。
8540	0x0000215C	ERROR_DS_SRC_AND_DST_OBJECT_CLASS_MISMATCH	ソースおよびデスティネーションオブジェクトは同じタイプである必要があります。
8541	0x0000215D	ERROR_SAM_INIT_FAILURE	以下のエラーのために、Security Accounts Managerの初期化に失敗しました: %1。エラーステータス: 0x%2。OKをクリックしてシステムをシャットダウンし、Safeモードで再起動してください。詳細情報をイベントログでチェックしてください。
8542	0x0000215E	ERROR_DS_DRA_SCHEMA_INFO_SHIP	スキーマ情報を複製リクエストに含めることができませんでした。
8543	0x0000215F	ERROR_DS_DRA_SCHEMA_CONFLICT	スキーマに互換性がないため、複製操作を完了できませんでした。
8544	0x00002160	ERROR_DS_DRA_EARLIER_SCHEMA_CONFLICT	以前のスキーマに互換性がないため、複製操作を完了できませんでした。
8545	0x00002161	ERROR_DS_DRA_OBJ_NC_MISMATCH	ソースまたはデスティネーションが最新のクロスドメイン移動操作に関する情報をまだ受信していないため、複製の更新が適用できませんでした。
8546	0x00002162	ERROR_DS_NC_STILL_HAS_DSAS	このドメインのホストを続けているドメインコントローラが存在するため、要求されたドメインが削除できませんでした。
8547	0x00002163	ERROR_DS_GC_REQUIRED	要求された操作は、グローバルカタログサーバでのみ実行できます。
8548	0x00002164	ERROR_DS_LOCAL_MEMBER_OF_LOCAL_ONLY	ローカルグループは、同じドメインの他のローカルグループのメンバにのみなることができます。
8549	0x00002165	ERROR_DS_NO_FPO_IN_UNIVERSAL_GROUPS	その他のグループのセキュリティプリンシパルは、ユニバーサルグループのメンバになることができません。
8550	0x00002166	ERROR_DS_CANT_ADD_TO_GC	セキュリティ上の理由から、この属性はGCに複製を作成することは許可されていません。
8551	0x00002167	ERROR_DS_NO_CHECKPOINT_WITH_PDC	現在処理中に変更が多すぎるため、PDCを持つチェックポイントを取得することができませんでした。
8552	0x00002168	ERROR_DS_SOURCE_AUDITING_NOT_ENABLED	この操作にはソースドメインの監査機能が有効になっている必要があります。
8553	0x00002169	ERROR_DS_CANT_CREATE_IN_NONDOMAIN_NC	セキュリティプリンシパルオブジェクトドメインの名前付けコンテキスト内でのみ作成することができます。
8554	0x0000216A	ERROR_DS_INVALID_NAME_FOR_SPN	提供されたホスト名が必要な形式でないため、Service Principal Name (SPN) を構築できませんでした。
8555	0x0000216B	ERROR_DS_FILTER_USES_CONSTRUCTED_ATTRS	構築された属性を使用するフィルタが渡されました。
8556	0x0000216C	ERROR_DS_UNICODEPWD_NOT_IN_QUOTES	unicodePwd属性値は、二重引用符で囲む必要があります。
8557	0x0000216D	ERROR_DS_MACHINE_ACCOUNT_QUOTA_EXCEEDED	コンピュータがドメインに結合できませんでした。このドメインで作成できるコンピュータアカウントの最大数を超過しています。システム管理者に問い合わせ、この制限をリセットまたは最大値を大きくしてもらってください。
8558	0x0000216E	ERROR_DS_MUST_BE_RUN_ON_DST_DC	セキュリティ上の理由から、この操作はデスティネーションDC上で実行する必要があります。
8559	0x0000216F	ERROR_DS_SRC_DC_MUST_BE_SP4_OR_GREATER	セキュリティ上の理由から、ソースDCはNT4SP4以上である必要があります。

エラー			説明
10進数	16進数	名前	
8560	0x00002170	ERROR_DS_CANT_TREE_DELETE_CRITICAL_OBJ	Critical Directory Service Systemオブジェクトは、ツリー削除操作中に削除できません。ツリー削除が部分的に実行される場合があります。
8561	0x00002171	ERROR_DS_INIT_FAILURE_CONSOLE	以下のエラーのために、ディレクトリサービスを開始できませんでした: %1. エラーステータス: 0x%2. OKをクリックしてシステムをシャットダウンしてください。さらにシステムの診断を続けるには、復元コンソールを使用できます。
8562	0x00002172	ERROR_DS_SAM_INIT_FAILURE_CONSOLE	以下のエラーのために、Security Accounts Managerの初期化に失敗しました: %1. エラーステータス: 0x%2. OKをクリックしてシステムをシャットダウンしてください。さらにシステムの診断を続けるには、復元コンソールを使用できます。
8563	0x00002173	ERROR_DS_FOREST_VERSION_TOO_HIGH	このWindowsバージョンは古すぎて、最新のディレクトリフォレストの動作をサポートしていません。このサーバをフォレストのドメインコントローラにする前に、このサーバのOSをアップグレードする必要があります。
8564	0x00002174	ERROR_DS_DOMAIN_VERSION_TOO_HIGH	このWindowsバージョンは古すぎて、最新のドメインの動作をサポートしていません。このサーバをドメインのドメインコントローラにする前に、このサーバのOSをアップグレードする必要があります。
8565	0x00002175	ERROR_DS_FOREST_VERSION_TOO_LOW	このWindowsバージョンは、このディレクトリフォレストで使用される動作バージョンをサポートしていません。このサーバをフォレストのドメインコントローラにする前に、フォレスト動作バージョンをアップグレードする必要があります。
8566	0x00002176	ERROR_DS_DOMAIN_VERSION_TOO_LOW	このWindowsバージョンは、このドメインで使用される動作バージョンをサポートしていません。このサーバをドメインのドメインコントローラにする前に、フォレスト動作バージョンをアップグレードする必要があります。
8567	0x00002177	ERROR_DS_INCOMPATIBLE_VERSION	Windowsのバージョンに、ドメインまたはフォレストの動作バージョンとの互換性がありません。
8568	0x00002178	ERROR_DS_LOW_DSA_VERSION	要求された値より低いバージョンのドメインコントローラが依然として存在するため、この動作バージョンをアップグレードすることができません。
8569	0x00002179	ERROR_DS_NO_BEHAVIOR_VERSION_IN_MIXEDDOMAIN	ドメインが混合ドメインモードのままでは、動作バージョンの値を大きくすることはできません。動作バージョンを大きくする前に、最初にドメインをネイティブモードに変更する必要があります。
8570	0x0000217A	ERROR_DS_NOT_SUPPORTED_SORT_ORDER	要求されたソート順がサポートされていません。
8571	0x0000217B	ERROR_DS_NAME_NOT_UNIQUE	重複する名前を持つオブジェクトが見つかりました。
8572	0x0000217C	ERROR_DS_MACHINE_ACCOUNT_CREATED_PRENT4	マシンアカウントが、NT4以前に作成されています。アカウントを再作成する必要があります。
8573	0x0000217D	ERROR_DS_OUT_OF_VERSION_STORE	データベースのバージョンがサポートされていません。
8574	0x0000217E	ERROR_DS_INCOMPATIBLE_CONTROLS_USED	複数の競合する制御が使用されているため継続できません。
8575	0x0000217F	ERROR_DS_NO_REF_DOMAIN	このパーティションの有効なセキュリティ記述子参照ドメインを見つけることができません。
8576	0x00002180	ERROR_DS_RESERVED_LINK_ID	スキーマ更新の失敗: リンクIDが予約済みです。
8577	0x00002181	ERROR_DS_LINK_ID_NOT_AVAILABLE	スキーマ更新の失敗: 使用できるリンクIDがありません。
8578	0x00002182	ERROR_DS_AG_CANT_HAVE_UNIVERSAL_MEMBER	アカウントグループは、ユニバーサルグループをメンバにすることはできません。
8579	0x00002183	ERROR_DS_MODIFYDN_DISALLOWED_BY_INSTANCE_TYPE	名前付けコンテキストヘッドまたは読み取り専用オブジェクトでの名前の変更または移動操作は、許可されていません。

エラー			説明
10進数	16進数	名前	
8580	0x00002184	ERROR_DS_NO_OBJECT_MOVE_IN_SCHEMA_NC	スキーマ名前付けコンテキストでのオブジェクトの移動操作は、許可されていません。
8581	0x00002185	ERROR_DS_MODIFYDN_DISALLOWED_BY_FLAG	システムフラグがオブジェクトに設定されているため、オブジェクトを移動または名前変更することができません。
8582	0x00002186	ERROR_DS_MODIFYDN_WRONG_GRANDPARENT	このオブジェクトは、祖父母関係のコンテナを変更することは許可されていません。このオブジェクトの移動は禁止されていませんが、兄弟関係のコンテナへの移動に制限されています。
8583	0x00002187	ERROR_DS_NAME_ERROR_TRUST_REFERRAL	他のフォレストへの照会が生成され、完全に解決することができません。
8584	0x00002188	ERROR_NOT_SUPPORTED_ON_STANDARD_SERVER	要求されたアクションは、標準サーバではサポートされていません。
8585	0x00002189	ERROR_DS_CANT_ACCESS_REMOTE_PART_OF_AD	リモートサーバに配置されたアクティブディレクトリのパーティションにアクセスできませんでした。少なくとも1つのサーバが該当するパーティションで動作していることを確認してください。
8586	0x0000218A	ERROR_DS_CR_IMPOSSIBLE_TO_VALIDATE_V2	ディレクトリが提示された名前付けコンテキストの上に名前付けコンテキストの複製を保持していないため、または提示された名前付けコンテキストの上の名前付けコンテキストの複製に接続できないため、提示された名前付けコンテキスト(またはパーティション)を検証できません。親の名前付けコンテキストがDNSに適切に登録されていることを確認し、この名前付けコンテキストの少なくとも1つの複製がDomain Namingマスタによって到達可能であることを確認してください。
8587	0x0000218B	ERROR_DS_THREAD_LIMIT_EXCEEDED	このリクエストに対するスレッドの制限を超えました。
8588	0x0000218C	ERROR_DS_NOT_CLOSEST	グローバルカタログサーバが最も近い場所にありません。

エラー			説明
10進数	16進数	名前	
9001	0x00002329	DNS_ERROR_RCODE_FORMAT_ERROR	DNSサーバがフォーマットを解釈できません。
9002	0x0000232A	DNS_ERROR_RCODE_SERVER_FAILURE	DNSサーバの障害。
9003	0x0000232B	DNS_ERROR_RCODE_NAME_ERROR	DNS名が存在しません。
9004	0x0000232C	DNS_ERROR_RCODE_NOT_IMPLEMENTED	DNSリクエストが、ネームサーバではサポートされていません。
9005	0x0000232D	DNS_ERROR_RCODE_REFUSED	DNS操作が拒否されました。
9006	0x0000232E	DNS_ERROR_RCODE_YXDOMAIN	存在するはずのないDNS名が存在します。
9007	0x0000232F	DNS_ERROR_RCODE_YXRRSET	存在するはずのないDNS RRセットが存在します。
9008	0x00002330	DNS_ERROR_RCODE_NXRRSET	存在するはずのDNS RRセットが存在しません。
9009	0x00002331	DNS_ERROR_RCODE_NOTAUTH	DNSサーバがゾーンのコンテンツ情報を持っていません。
9010	0x00002332	DNS_ERROR_RCODE_NOTZONE	更新または必要条件のDNS名がゾーンにありません。
9016	0x00002338	DNS_ERROR_RCODE_BADSIG	DNSシグネチャが検証に失敗しました。
9017	0x00002339	DNS_ERROR_RCODE_BADKEY	DNSキーが正しくありません。
9018	0x0000233A	DNS_ERROR_RCODE_BADTIME	DNSシグネチャの有効期限が過ぎました。
9501	0x0000251D	DNS_INFO_NO_RECORDS	受け取ったDNSクエリーに対するレコードが見つかりませんでした。
9502	0x0000251E	DNS_ERROR_BAD_PACKET	DNSパケットが正しくありません。
9503	0x0000251F	DNS_ERROR_NO_PACKET	DNSパケットがありません。
9504	0x00002520	DNS_ERROR_RCODE	DNSエラー、rcodeをチェックしてください。
9505	0x00002521	DNS_ERROR_UNSECURE_PACKET	非セキュアなDNSパケット。
9551	0x0000254F	DNS_ERROR_INVALID_TYPE	無効なDNSタイプ。
9552	0x00002550	DNS_ERROR_INVALID_IP_ADDRESS	無効なIPアドレス
9553	0x00002551	DNS_ERROR_INVALID_PROPERTY	無効なプロパティ
9554	0x00002552	DNS_ERROR_TRY_AGAIN_LATER	後でもう一度、DNS操作を試みてください。
9555	0x00002553	DNS_ERROR_NOT_UNIQUE	受け取った名前とタイプのレコードが一意ではありません。

		エラー	説明
10進数	16進数	名前	
9556	0x00002554	DNS_ERROR_NON_RFC_NAME	DNS名がRFC仕様に準拠していません。
9557	0x00002555	DNS_STATUS_FQDN	DNS名は完全に仕様に準拠しています。
9558	0x00002556	DNS_STATUS_DOTTED_NAME	DNS名にピリオドが含まれています(マルチラベル)。
9559	0x00002557	DNS_STATUS_SINGLE_PART_NAME	DNS名がシングルパート名です。
9560	0x00002558	DNS_ERROR_INVALID_NAME_CHAR	DSN名に無効な文字が含まれています。
9561	0x00002559	DNS_ERROR_NUMERIC_NAME	DNS名に数字だけが使用されています。
9601	0x00002581	DNS_ERROR_ZONE_DOES_NOT_EXIST	DNSゾーンが存在しません。
9602	0x00002582	DNS_ERROR_NO_ZONE_INFO	DNSゾーン情報が使用できません。
9603	0x00002583	DNS_ERROR_INVALID_ZONE_OPERATION	DNSゾーンに対する無効な操作。
9604	0x00002584	DNS_ERROR_ZONE_CONFIGURATION_ERROR	無効なDNSゾーン設定。
9605	0x00002585	DNS_ERROR_ZONE_HAS_NO_SOA_RECORD	DNSゾーンにStart Of Authority (SOA) レコードがありません。
9606	0x00002586	DNS_ERROR_ZONE_HAS_NO_NS_RECORDS	DNSゾーンにネームサーバ(NS) レコードがありません。
9607	0x00002587	DNS_ERROR_ZONE_LOCKED	DNSゾーンがロックされています。
9608	0x00002588	DNS_ERROR_ZONE_CREATION_FAILED	DNSゾーン作成に失敗しました。
9609	0x00002589	DNS_ERROR_ZONE_ALREADY_EXISTS	DNSゾーンは既に存在しています。
9610	0x0000258A	DNS_ERROR_AUTOZONE_ALREADY_EXISTS	DNS自動ゾーンは既に存在しています。
9611	0x0000258B	DNS_ERROR_INVALID_ZONE_TYPE	無効なDNSゾーンタイプ。
9612	0x0000258C	DNS_ERROR_SECONDARY_REQUIRES_MASTER_IP	セカンダリDNSゾーンにはマスタIPアドレスが必要です。
9613	0x0000258D	DNS_ERROR_ZONE_NOT_SECONDARY	DNSゾーンがセカンダリではありません。
9614	0x0000258E	DNS_ERROR_NEED_SECONDARY_ADDRESSES	セカンダリIPアドレスが必要です。
9615	0x0000258F	DNS_ERROR_WINS_INIT_FAILED	WINSの初期化に失敗しました。
9616	0x00002590	DNS_ERROR_NEED_WINS_SERVERS	WINSサーバが必要です。
9617	0x00002591	DNS_ERROR_NBSTAT_INIT_FAILED	NBSTAT初期化呼び出しに失敗しました。
9618	0x00002592	DNS_ERROR_SOA_DELETE_INVALID	Start Of Authority (SOA) の無効な削除
9619	0x00002593	DNS_ERROR_FORWARDER_ALREADY_EXISTS	条件付きフォワードゾーンはその名前です既に存在しています。
9651	0x000025B3	DNS_ERROR_PRIMARY_REQUIRES_DATAFILE	プライマリDNSゾーンにはデータファイルが必要です。
9652	0x000025B4	DNS_ERROR_INVALID_DATAFILE_NAME	DNSゾーンに対する無効なデータファイル名。
9653	0x000025B5	DNS_ERROR_DATAFILE_OPEN_FAILURE	DNSゾーンのデータファイルを開くことができませんでした。
9654	0x000025B6	DNS_ERROR_FILE_WRITEBACK_FAILED	DNSゾーンのデータファイルを書き込むことができませんでした。
9655	0x000025B7	DNS_ERROR_DATAFILE_PARSING	DNSゾーンのデータファイルの読み出し中の障害。
9701	0x000025E5	DNS_ERROR_RECORD_DOES_NOT_EXIST	DNSレコードが存在しません。
9702	0x000025E6	DNS_ERROR_RECORD_FORMAT	DNSレコードのフォーマットエラー。
9703	0x000025E7	DNS_ERROR_NODE_CREATION_FAILED	DNSでのノード作成に失敗。
9704	0x000025E8	DNS_ERROR_UNKNOWN_RECORD_TYPE	不明なDNSレコードタイプ。
9705	0x000025E9	DNS_ERROR_RECORD_TIMED_OUT	DNSレコードがタイムアウトしました。
9706	0x000025EA	DNS_ERROR_NAME_NOT_IN_ZONE	DNSゾーンに名前がありません。
9707	0x000025EB	DNS_ERROR_CNAME_LOOP	CNAMEループが検出されました。
9708	0x000025EC	DNS_ERROR_NODE_IS_CNAME	ノードはCNAME DNSレコードです。
9709	0x000025ED	DNS_ERROR_CNAME_COLLISION	CNAMEレコードは所定の名前で既に存在しています。
9710	0x000025EE	DNS_ERROR_RECORD_ONLY_AT_ZONE_ROOT	レコードがDNSゾーンのルートのみにあります。
9711	0x000025EF	DNS_ERROR_RECORD_ALREADY_EXISTS	DNSレコードは既に存在しています。
9712	0x000025F0	DNS_ERROR_SECONDARY_DATA	セカンダリDNSゾーンデータのエラー。
9713	0x000025F1	DNS_ERROR_NO_CREATE_CACHE_DATA	DNSキャッシュデータを作成できませんでした。
9714	0x000025F2	DNS_ERROR_NAME_DOES_NOT_EXIST	DNS名が存在しません。
9715	0x000025F3	DNS_WARNING_PTR_CREATE_FAILED	ポインタ (PTR) レコードを作成できません。
9716	0x000025F4	DNS_WARNING_DOMAIN_UNDELETED	DNSドメインの削除が取り消されました。
9717	0x000025F5	DNS_ERROR_DS_UNAVAILABLE	ディレクトリサービスが使用できません。

エラー			説明
10進数	16進数	名前	
9718	0x000025F6	DNS_ERROR_DS_ZONE_ALREADY_EXISTS	DNSゾーンがディレクトリサービスに既に存在しています。
9719	0x000025F7	DNS_ERROR_NO_BOOTFILE_IF_DS_ZONE	ディレクトリサービスのブートファイルの作成または読み取りをしないDNSサーバが、DNSゾーンを統合しました。
9751	0x00002617	DNS_INFO_AXFR_COMPLETE	DNS AXFR (ゾーン転送)が完了しました。
9752	0x00002618	DNS_ERROR_AXFR	DNSゾーン転送に失敗しました。
9753	0x00002619	DNS_INFO_ADDED_LOCAL_WINS	追加されたローカルWINSサーバ。
9801	0x00002649	DNS_STATUS_CONTINUE_NEEDED	セキュアな更新呼び出しが、更新リクエストを継続する必要があります。
9851	0x0000267B	DNS_ERROR_NO_TCPIP	TCP/IPネットワークプロトコルがインストールされていません。
9852	0x0000267C	DNS_ERROR_NO_DNS_SERVERS	DNSサーバがローカルシステムに設定されていません。
9901	0x000026AD	DNS_ERROR_DP_DOES_NOT_EXIST	指定されたディレクトリパーティションは存在しません。
9902	0x000026AE	DNS_ERROR_DP_ALREADY_EXISTS	指定されたディレクトリパーティションは既に存在しています。
9903	0x000026AF	DNS_ERROR_DP_NOT_ENLISTED	DSは指定されたディレクトリパーティションに組み込まれていません。
9904	0x000026B0	DNS_ERROR_DP_ALREADY_ENLISTED	DSは既に指定されたディレクトリパーティションに組み込まれています。

エラー			説明
10進数	16進数	名前	
10004	0x00002714	WSAEINTR	ブロック操作がWSACancelBlockingCallの呼び出しによって中断されました。
10009	0x00002719	WSAEBADF	渡されたファイルハンドルが有効ではありません。
10013	0x0000271D	WSAEACCES	アクセス権限によって禁止されている方法でソケットにアクセスしようとしました。
10014	0x0000271E	WSAEFAULT	呼び出しのポインタ引数を使用しようとして、システムが無効なポインタアドレスを検出しました。
10022	0x00002726	WSAEINVAL	無効な引数が渡されました。
10024	0x00002728	WSAEMFILE	開いているソケットが多すぎます。
10035	0x00002733	WSAEWOULDBLOCK	ノンブロッキングソケット操作が直ちに完了できませんでした。
10036	0x00002734	WSAEINPROGRESS	ブロック操作が現在実行中です。
10037	0x00002735	WSAEALREADY	既に操作が進行中のノンブロッキングソケットで操作を行おうとしました。
10038	0x00002736	WSAENOTSOCK	ソケット以外のもので操作を行おうとしました。
10039	0x00002737	WSAEDESTADDRREQ	ソケットの操作から必要なアドレスが省略されています。
10040	0x00002738	WSAEMSGSIZE	データグラムソケットに送信されたメッセージが内部メッセージバッファまたはその他のネットワーク制限より大きいか、またはデータグラムの受信に使用されたバッファがデータグラム自体より小さい。
10041	0x00002739	WSAEPROTOTYPE	ソケット関数呼び出しで、要求されたソケットタイプのセマンティクスをサポートしていないプロトコルが指定されました。
10042	0x0000273A	WSAENOPROTOPT	不明な、無効な、またはサポートされていないオプションまたはレベルが、getsockopt または setsockopt の呼び出しで指定されました。
10043	0x0000273B	WSAEPROTONOSUPPORT	要求されたプロトコルがシステムに設定されていないか、またはそのための実装が存在しません。
10044	0x0000273C	WSAESOCKTNOSUPPORT	指定されたソケットタイプのサポートが、このアドレスファミリに存在しません。
10045	0x0000273D	WSAEPNOTSUPP	この操作は、参照されたオブジェクトのタイプではサポートされていません。
10046	0x0000273E	WSAEPFNOSUPPORT	プロトコルファミリがシステムに設定されていないか、またはそのための実装が存在しません。

エラー			説明
10進数	16進数	名前	
10047	0x0000273F	WSAEAFNOSUPPORT	要求されたプロトコルと互換性がないアドレスが使用されました。
10048	0x00002740	WSAEADDRINUSE	各ソケットアドレス(プロトコル/ネットワークアドレス/ポート)は、通常、1回のみの使用が許可されています。
10049	0x00002741	WSAEADDRNOTAVAIL	要求されたアドレスは、このコンテキストでは有効ではありません。
10050	0x00002742	WSAENETDOWN	ソケット操作でデッドネットワーク(ネットワーク接続の喪失)が発生しました。
10051	0x00002743	WSAENETUNREACH	到達できないネットワークに対してソケット操作を行おうとしました。
10052	0x00002744	WSAENETRESET	操作が進行中にエラーを検出するkeep-alive動作のために、接続が切断しました。
10053	0x00002745	WSAECONNABORTED	確立した接続がホストマシンのソフトウェアによって中止されました。
10054	0x00002746	WSAECONNRESET	既存の接続がリモートホストによって強制的に終了しました。
10055	0x00002747	WSAENOBUFS	システムに十分なバッファスペースがないため、またはキューの空き領域がないため、ソケットでの操作が実行できませんでした。
10056	0x00002748	WSAEISCONN	既に接続されているソケットに対して接続リクエストが発行されました。
10057	0x00002749	WSAENOTCONN	ソケットが接続されておらず、(sendto呼び出しを使用してデータグラムソケットで送信時に)アドレスが提供されなかったため、データを送受信するリクエストは許可されませんでした。
10058	0x0000274A	WSAESHUTDOWN	ソケットが前のシャットダウン呼び出しに従って既にシャットダウンされたため、データを送受信するリクエストは許可されませんでした。
10059	0x0000274B	WSAETOOMANYREFS	カーネルオブジェクトへの参照が多すぎます。
10060	0x0000274C	WSAETIMEDOUT	接続された側が一定の時間後に適切に応答しなかったために接続に失敗したか、または接続されたホストが応答に失敗したために確立された接続でエラーになりました。
10061	0x0000274D	WSAECONNREFUSED	ターゲットマシンが明確に拒否しているため、接続できませんでした。
10062	0x0000274E	WSAELOOP	名前を変換できません。
10063	0x0000274F	WSAENAMETOOLONG	名前の構成要素または名前が長すぎます。
10064	0x00002750	WSAEHOSTDOWN	デスティネーションホストが停止したため、ソケット操作に失敗しました。
10065	0x00002751	WSAEHOSTUNREACH	到達できないホストに対してソケット操作を行おうとしました。
10066	0x00002752	WSAENOTEMPTY	空でないディレクトリは削除することができません。
10067	0x00002753	WSAEPROCLIM	Windows Socketsの実装には、同時にこれを使用するアプリケーションの数に制限がある場合があります。
10068	0x00002754	WSAEUSERS	割り当てが不足しています。
10069	0x00002755	WSAEDQUOT	ディスク割り当てが不足しています。
10070	0x00002756	WSAESTALE	ファイルハンドル参照は使用できなくなりました。
10071	0x00002757	WSAEREMOTE	アイテムがローカルで使用できません。
10091	0x0000276B	WSASYSNOTREADY	ネットワークサービスを提供するためにこの機能を使用する基本システムが現在使用できないため、WSAStartupは現在使用できません。
10092	0x0000276C	WSAVERNOTSUPPORTED	要求されたWindows Socketsのバージョンがサポートされていません。
10093	0x0000276D	WSANOTINITIALISED	アプリケーションがWSAStartupを呼び出さなかったか、またはWSAStartupが失敗したかのいずれかです。

		エラー		説明
10進数	16進数	名前		
10101	0x00002775	WSAEDISCON		リモート側がグレースフル (Graceful) シャットダウンシーケンスを開始したことを示すために、WSARecv または WSARecvFrom によって返されました。
10102	0x00002776	WSAENOMORE		WSALookupServiceNextによって返される結果はこれ以上ありません。
10103	0x00002777	WSAECANCELLED		この呼び出しの処理中に、WSALookupServiceEndの呼び出しが行われました。呼び出しはキャンセルされました。
10104	0x00002778	WSAEINVALIDPROCTABLE		プロシージャ呼び出しテーブルが無効です。
10105	0x00002779	WSAEINVALIDPROVIDER		要求されたサービスプロバイダが無効です。
10106	0x0000277A	WSAEPROVIDERFAILEDINIT		要求されたサービスプロバイダがロードまたは初期化できませんでした。
10107	0x0000277B	WSASYSALLFAILURE		失敗してはならないシステム呼び出しが失敗しました。
10108	0x0000277C	WSASERVICE_NOT_FOUND		そのようなサービスは不明です。指定された名前空間でこのサービスを見つけることはできません。
10109	0x0000277D	WSATYPE_NOT_FOUND		指定されたクラスが見つかりませんでした。
10110	0x0000277E	WSA_E_NO_MORE		WSALookupServiceNextによって返される結果はこれ以上ありません。
10111	0x0000277F	WSA_E_CANCELLED		この呼び出しの処理中に、WSALookupServiceEndの呼び出しが行われました。呼び出しはキャンセルされました。
10112	0x00002780	WSAEREFUSED		データベースのクエリーは、明確に拒否されたため失敗しました。
11001	0x00002AF9	WSAHOST_NOT_FOUND		そのようなホストは不明です。
11002	0x00002AFA	WSATRY_AGAIN		これは通常、ホスト名解決中の一時的なエラーで、ローカルサーバが権威サーバから応答を受信しなかったことを意味します。
11003	0x00002AFB	WSANO_RECOVERY		データベースルックアップ中に回復不能なエラーが発生しました。
11004	0x00002AFC	WSANO_DATA		要求された名前は有効でありデータベース内に見つかりましたが、解決を行うための正しく関連付けられたデータがありません。
11005	0x00002AFD	WSA_QOS_RECEIVERS		少なくとも1つの予備が到着しました。
11006	0x00002AFE	WSA_QOS_SENDERS		少なくとも1つのパスが到着しました。
11007	0x00002AFF	WSA_QOS_NO_SENDERS		セNDERはありません。
11008	0x00002B00	WSA_QOS_NO_RECEIVERS		レシーバはありません。
11009	0x00002B01	WSA_QOS_REQUEST_CONFIRMED		予備が確認されました。
11010	0x00002B02	WSA_QOS_ADMISSION_FAILURE		リソース不足によるエラー。
11011	0x00002B03	WSA_QOS_POLICY_FAILURE		管理上の理由から拒否されました - 資格情報が無効です。
11012	0x00002B04	WSA_QOS_BAD_STYLE		不明または競合するスタイル。
11013	0x00002B05	WSA_QOS_BAD_OBJECT		一般的に、filterspec または providerspecificバッファの一部で問題が発生しています。
11014	0x00002B06	WSA_QOS_TRAFFIC_CTRL_ERROR		flowspecの一部で問題が発生しています。
11015	0x00002B07	WSA_QOS_GENERIC_ERROR		一般的なQOSエラー。
11016	0x00002B08	WSA_QOS_ESERVICETYPE		無効または認識されないサービスタイプがflowspecに見つかりました。
11017	0x00002B09	WSA_QOS_EFLOWSPEC		無効または整合性のないflowspecが、QOS構造体に見つかりました。
11018	0x00002B0A	WSA_QOS_EPROVSPECBUF		無効なQOSプロバイダ固有のバッファ。
11019	0x00002B0B	WSA_QOS_EFILTERSTYLE		無効なQOSフィルタスタイルが使用されました。
11020	0x00002B0C	WSA_QOS_EFILTERTYPE		無効なQOSフィルタタイプが使用されました。
11021	0x00002B0D	WSA_QOS_EFILTERCOUNT		FLOWDESCRIPTORに指定されたQOS FILTERSPECの数が正しくありません。
11022	0x00002B0E	WSA_QOS_EOBJLENGTH		無効なObjectLengthフィールドを持つオブジェクトが、QOSプロバイダ固有のバッファに指定されました。

エラー			説明
10進数	16進数	名前	
11023	0x00002B0F	WSA_QOS_EFLOWCOUNT	QOS構造体に指定されたフロー記述子の数が正しくありません。
11024	0x00002B10	WSA_QOS_EUNKNOWNPSSOBJ	認識されないオブジェクトが、QOSプロバイダ固有のバッファで見つかりました。
11025	0x00002B11	WSA_QOS_EPOLICYOBJ	無効なポリシーオブジェクトが、QOSプロバイダ固有のバッファで見つかりました。
11026	0x00002B12	WSA_QOS_EFLOWDESC	無効なQOSフロー記述子が、フロー記述子リストで見つかりました。
11027	0x00002B13	WSA_QOS_EPSFLOWSPEC	無効または整合性のないflowspecが、QOSプロバイダ固有のバッファで見つかりました。
11028	0x00002B14	WSA_QOS_EPSFILTERSPEC	無効なFILTERSPECが、QOSプロバイダ固有のバッファで見つかりました。
11029	0x00002B15	WSA_QOS_ESDMODEOBJ	無効な形状破棄モードオブジェクトが、QOSプロバイダ固有のバッファで見つかりました。
11030	0x00002B16	WSA_QOS_ESHAPERATEOBJ	無効なシェイピングレートオブジェクトが、QOSプロバイダ固有のバッファで見つかりました。
11031	0x00002B17	WSA_QOS_RESERVED_PETYPE	予約されたポリシーエレメントが、QOSプロバイダ固有のバッファで見つかりました。

エラー			説明
10進数	16進数	名前	
12000	0x00002EE0	ERROR_SXS_SECTION_NOT_FOUND	要求されたセクションは、アクティベーションコンテキストには存在しませんでした。
12001	0x00002EE1	ERROR_SXS_CANT_GEN_ACTCTX	このアプリケーションは、アプリケーションの設定が正しくないため、開始できませんでした。アプリケーションを再インストールすることで問題が解決することもあります。
12002	0x00002EE2	ERROR_SXS_INVALID_ACTCTXDATA_FORMAT	アプリケーションのバインディングデータ形式が無効です。
12003	0x00002EE3	ERROR_SXS_ASSEMBLY_NOT_FOUND	参照されたアセンブリはシステムにインストールされていません。
12004	0x00002EE4	ERROR_SXS_MANIFEST_FORMAT_ERROR	マニフェストファイルは、必要なタグおよびフォーマット情報から開始していません。
12005	0x00002EE5	ERROR_SXS_MANIFEST_PARSE_ERROR	マニフェストファイルに1つ以上の構文エラーが含まれています。
12006	0x00002EE6	ERROR_SXS_ACTIVATION_CONTEXT_DISABLED	アプリケーションが、無効になったアクティベーションコンテキストをアクティベートしようとしてしました。
12007	0x00002EE7	ERROR_SXS_KEY_NOT_FOUND	要求されたロックアップキーは、有効なアクティベーションコンテキストで見つかりませんでした。
12008	0x00002EE8	ERROR_SXS_VERSION_CONFLICT	アプリケーションによって要求されるコンポーネントバージョンが、既にアクティブな別のコンポーネントのバージョンと競合しています。
12009	0x00002EE9	ERROR_SXS_WRONG_SECTION_TYPE	要求されたアクティベーションコンテキストセクションのタイプが、使用されているクエリ-APIと一致しません。
12010	0x00002EEA	ERROR_SXS_THREAD_QUERIES_DISABLED	システムリソースが不足しているため、現在の実行スレッドで分離アクティベーションを無効にする必要があります。
12011	0x00002EEB	ERROR_SXS_PROCESS_DEFAULT_ALREADY_SET	プロセスのデフォルトアクティベーションコンテキストが既に設定されているため、プロセスのデフォルトアクティベーションコンテキストを設定しようとして失敗しました。
12012	0x00002EEC	ERROR_SXS_UNKNOWN_ENCODING_GROUP	指定されたエンコーディンググループIDが認識されません。
12013	0x00002EED	ERROR_SXS_UNKNOWN_ENCODING	要求されたエンコーディングが認識されません。
12014	0x00002EEE	ERROR_SXS_INVALID_XML_NAMESPACE_URI	マニフェストに無効なURIへの参照が含まれています。
12015	0x00002EEF	ERROR_SXS_ROOT_MANIFEST_DEPENDENCY_NOT_INSTALLED	アプリケーションマニフェストに、インストールされていない依存アセンブリへの参照が含まれています。

エラー			説明
10進数	16進数	名前	
12016	0x00002EF0	ERROR_SXS_LEAF_MANIFEST_DEPENDENCY_NOT_INSTALLED	アプリケーションによって使用されるアセンブリのマニフェストに、インストールされていない依存アセンブリへの参照が含まれています。
12017	0x00002EF1	ERROR_SXS_INVALID_ASSEMBLY_IDENTITY_ATTRIBUTE	マニフェストに有効でないアセンブリIDの属性が含まれています。
12018	0x00002EF2	ERROR_SXS_MANIFEST_MISSING_REQUIRED_DEFAULT_NAMESPACE	マニフェストに、アセンブリエレメントに関する必要なデフォルト名前空間仕様が見つかりません。
12019	0x00002EF3	ERROR_SXS_MANIFEST_INVALID_REQUIRED_DEFAULT_NAMESPACE	マニフェストにはアセンブリエレメントに指定されたデフォルト名前空間が含まれていますが、その値が「urn:schemas-microsoft-com:asm.v1」ではありません。
12020	0x00002EF4	ERROR_SXS_PRIVATE_MANIFEST_CROSS_PATH_WITH_REPARSE_POINT	プライベートマニフェストプローブが、リパースポイント関連のパスを通過しました。
12021	0x00002EF5	ERROR_SXS_DUPLICATE_DLL_NAME	アプリケーションマニフェストによって直接または間接的に参照される2つ以上のコンポーネントが、同じ名前のファイルを持っています。
12022	0x00002EF6	ERROR_SXS_DUPLICATE_WINDOWCLASS_NAME	アプリケーションマニフェストによって直接または間接的に参照される2つ以上のコンポーネントが、同じ名前のウィンドウクラスを持っています。
12023	0x00002EF7	ERROR_SXS_DUPLICATE_CLSID	アプリケーションマニフェストによって直接または間接的に参照される2つ以上のコンポーネントが、同じ名前のCOMサーバのCLSIDを持っています。
12024	0x00002EF8	ERROR_SXS_DUPLICATE_IID	アプリケーションマニフェストによって直接または間接的に参照される2つ以上のコンポーネントが、同じCOMインターフェイスIIDのプロキシを持っています。
12025	0x00002EF9	ERROR_SXS_DUPLICATE_TLBID	アプリケーションマニフェストによって直接または間接的に参照される2つ以上のコンポーネントが、同じCOMタイプライブラリのTLBIDを持っています。
12026	0x00002EFA	ERROR_SXS_DUPLICATE_PROGID	アプリケーションマニフェストによって直接または間接的に参照される2つ以上のコンポーネントが、同じCOM ProgIDを持っています。
12027	0x00002EFB	ERROR_SXS_DUPLICATE_ASSEMBLY_NAME	アプリケーションマニフェストによって直接または間接的に参照される2つ以上のコンポーネントが、同じコンポーネントの異なるバージョンです (これは許可されていません)。
12028	0x00002EFC	ERROR_SXS_FILE_HASH_MISMATCH	コンポーネントのファイルが、コンポーネントマニフェストに存在する検証情報と一致しません。
12029	0x00002EFD	ERROR_SXS_POLICY_PARSE_ERROR	ポリシーマニフェストに1つ以上の構文エラーが含まれています。
12030	0x00002EFE	ERROR_SXS_XML_E_MISSINGQUOTE	マニフェストパースエラー: 文字列リテラルが必要でしたが、開始の引用符文字が見つかりませんでした。
12031	0x00002EFF	ERROR_SXS_XML_E_COMMENTSYNTAX	マニフェストパースエラー: コメントで使用された構文が正しくありません。
12032	0x00002F00	ERROR_SXS_XML_E_BADSTARTNAMECHAR	マニフェストパースエラー: 名前が無効な文字で開始されました。
12033	0x00002F01	ERROR_SXS_XML_E_BADNAMECHAR	マニフェストパースエラー: 名前に無効な文字が含まれています。
12034	0x00002F02	ERROR_SXS_XML_E_BADCHARINSTRING	マニフェストパースエラー: 文字列リテラルに無効な文字が含まれています。
12035	0x00002F03	ERROR_SXS_XML_E_XMLDECLSYNTAX	マニフェストパースエラー: XML宣言の無効な構文。
12036	0x00002F04	ERROR_SXS_XML_E_BADCHARDATA	マニフェストパースエラー: 無効な文字がテキストコンテキストに見つかりました。
12037	0x00002F05	ERROR_SXS_XML_E_MISSINGWHITESPACE	マニフェストパースエラー: 必要な空白がありません。
12038	0x00002F06	ERROR_SXS_XML_E_EXPECTINGTAGEND	マニフェストパースエラー: 文字 ' > ' が必要です。
12039	0x00002F07	ERROR_SXS_XML_E_MISSINGSEMICOLON	マニフェストパースエラー: セミコロンが必要です。

エラー			説明
10進数	16進数	名前	
12040	0x00002F08	ERROR_SXS_XML_E_UNBALANCEDPAREN	マニフェストパースエラー: 括弧が釣り合っていない。
12041	0x00002F09	ERROR_SXS_XML_E_INTERNALERROR	マニフェストパースエラー: 内部エラー。
12042	0x00002F0A	ERROR_SXS_XML_E_UNEXPECTED_WHITESPACE	マニフェストパースエラー: この位置には空白を入れてはいけません。
12043	0x00002F0B	ERROR_SXS_XML_E_INCOMPLETE_ENCODING	マニフェストパースエラー: 現在のエンコーディングでは無効な状態でファイル末尾に達しました。
12044	0x00002F0C	ERROR_SXS_XML_E_MISSING_PAREN	マニフェストパースエラー: 括弧がありません。
12045	0x00002F0D	ERROR_SXS_XML_E_EXPECTINGCLOSEQUOTE	マニフェストパースエラー: 一重または二重の閉じ引用符文字(‘ or ’)がありません。
12046	0x00002F0E	ERROR_SXS_XML_E_MULTIPLE_COLONS	マニフェストパースエラー: 名前には複数のコロンは使用できません。
12047	0x00002F0F	ERROR_SXS_XML_E_INVALID_DECIMAL	マニフェストパースエラー: 10進数で無効な文字。
12048	0x00002F10	ERROR_SXS_XML_E_INVALID_HEXIDECIMAL	マニフェストパースエラー: 16進数で無効な文字。
12049	0x00002F11	ERROR_SXS_XML_E_INVALID_UNICODE	マニフェストパースエラー: このプラットフォームで無効なUnicode文字値。
12050	0x00002F12	ERROR_SXS_XML_E_WHITESPACEORQUESTIONMARK	マニフェストパースエラー: 空白または ‘?’ が必要です。
12051	0x00002F13	ERROR_SXS_XML_E_UNEXPECTEDENDTAG	マニフェストパースエラー: この位置に終了タグは必要ありません。
12052	0x00002F14	ERROR_SXS_XML_E_UNCLOSEDTAG	マニフェストパースエラー: 以下のタグが閉じられていません: %1
12053	0x00002F15	ERROR_SXS_XML_E_DUPLICATEATTRIBUTE	マニフェストパースエラー: 重複属性。
12054	0x00002F16	ERROR_SXS_XML_E_MULTIPLEROOTS	マニフェストパースエラー: XMLドキュメントではトップレベルエレメントは1つしか許可されていません。
12055	0x00002F17	ERROR_SXS_XML_E_INVALIDATROOTLEVEL	マニフェストパースエラー: ドキュメントのトップレベルが無効です。
12056	0x00002F18	ERROR_SXS_XML_E_BADXMLDECL	マニフェストパースエラー: 無効なXML宣言。
12057	0x00002F19	ERROR_SXS_XML_E_MISSINGROOT	マニフェストパースエラー: XMLドキュメントにはトップレベルエレメントが必要です。
12058	0x00002F1A	ERROR_SXS_XML_E_UNEXPECTEDEOF	マニフェストパースエラー: 予期しないファイル末尾。
12059	0x00002F1B	ERROR_SXS_XML_E_BADPEREFINSUBSET	マニフェストパースエラー: パラメータエンティティは、内部サブセットのマークアップ宣言内では使用できません。
12060	0x00002F1C	ERROR_SXS_XML_E_UNCLOSEDSTARTTAG	マニフェストパースエラー: エレメントが閉じられていません。
12061	0x00002F1D	ERROR_SXS_XML_E_UNCLOSEDENDTAG	マニフェストパースエラー: 終了エレメントに文字 ‘>’ がありません。
12062	0x00002F1E	ERROR_SXS_XML_E_UNCLOSEDSTRING	マニフェストパースエラー: 文字列リテラルが閉じられていません。
12063	0x00002F1F	ERROR_SXS_XML_E_UNCLOSEDCOMMENT	マニフェストパースエラー: コメントが閉じられていません。
12064	0x00002F20	ERROR_SXS_XML_E_UNCLOSEDDECL	マニフェストパースエラー: 宣言が閉じられていません。
12065	0x00002F21	ERROR_SXS_XML_E_UNCLOSEDCDATA	マニフェストパースエラー: CDATAセクションが閉じられていません。
12066	0x00002F22	ERROR_SXS_XML_E_RESERVEDNAMESPACE	マニフェストパースエラー: 名前空間プレフィックスは、予約済み文字列 ‘xml’ で開始してはなりません。
12067	0x00002F23	ERROR_SXS_XML_E_INVALIDENCODING	マニフェストパースエラー: システムは指定されたエンコーディングをサポートしていません。
12068	0x00002F24	ERROR_SXS_XML_E_INVALIDSWITCH	マニフェストパースエラー: 現在のエンコーディングから指定されたエンコーディングへの切り替えはサポートされていません。
12069	0x00002F25	ERROR_SXS_XML_E_BADXMLCASE	マニフェストパースエラー: 名前 ‘xml’ は予約されており、小文字を使用する必要があります。

		エラー		説明
10進数	16進数	名前		
12070	0x00002F26	ERROR_SXS_XML_E_INVALID_STANDALONE		マニフェストパースエラー：スタンドアロン属性は値 'yes' または 'no' を持つ必要があります。
12071	0x00002F27	ERROR_SXS_XML_E_UNEXPECTED_STANDALONE		マニフェストパースエラー：スタンドアロン属性は外部エンティティでは使用できません。
12072	0x00002F28	ERROR_SXS_XML_E_INVALID_VERSION		マニフェストパースエラー：無効なバージョン番号。
12073	0x00002F29	ERROR_SXS_XML_E_MISSINGEQUALS		マニフェストパースエラー：属性と属性値の間に等号がありません。
13000	0x000032C8	ERROR_IPSEC_QM_POLICY_EXISTS		指定されたクイックモードポリシーは既に存在しています。
13001	0x000032C9	ERROR_IPSEC_QM_POLICY_NOT_FOUND		指定されたクイックモードポリシーが見つかりませんでした。
13002	0x000032CA	ERROR_IPSEC_QM_POLICY_IN_USE		指定されたクイックモードポリシーは使用中です。
13003	0x000032CB	ERROR_IPSEC_MM_POLICY_EXISTS		指定されたメインモードポリシーは既に存在しています。
13004	0x000032CC	ERROR_IPSEC_MM_POLICY_NOT_FOUND		指定されたメインモードポリシーが見つかりませんでした。
13005	0x000032CD	ERROR_IPSEC_MM_POLICY_IN_USE		指定されたメインモードポリシーは使用中です。
13006	0x000032CE	ERROR_IPSEC_MM_FILTER_EXISTS		指定されたメインモードフィルタは既に存在しています。
13007	0x000032CF	ERROR_IPSEC_MM_FILTER_NOT_FOUND		指定されたメインモードフィルタが見つかりませんでした。
13008	0x000032D0	ERROR_IPSEC_TRANSPORT_FILTER_EXISTS		指定された転送モードフィルタは既に存在しています。
13009	0x000032D1	ERROR_IPSEC_TRANSPORT_FILTER_NOT_FOUND		指定された転送モードフィルタは存在しません。
13010	0x000032D2	ERROR_IPSEC_MM_AUTH_EXISTS		指定されたメインモード認証リストは存在しています。
13011	0x000032D3	ERROR_IPSEC_MM_AUTH_NOT_FOUND		指定されたメインモード認証リストが見つかりませんでした。
13012	0x000032D4	ERROR_IPSEC_MM_AUTH_IN_USE		指定されたクイックモードポリシーは使用中です。
13013	0x000032D5	ERROR_IPSEC_DEFAULT_MM_POLICY_NOT_FOUND		指定されたメインモードポリシーが見つかりませんでした。
13014	0x000032D6	ERROR_IPSEC_DEFAULT_MM_AUTH_NOT_FOUND		指定されたクイックモードポリシーが見つかりませんでした。
13015	0x000032D7	ERROR_IPSEC_DEFAULT_QM_POLICY_NOT_FOUND		マニフェストファイルに1つ以上の構文エラーが含まれています。
13016	0x000032D8	ERROR_IPSEC_TUNNEL_FILTER_EXISTS		アプリケーションが、無効になったアクティベーションコンテキストをアクティベートしようとしてしました。
13017	0x000032D9	ERROR_IPSEC_TUNNEL_FILTER_NOT_FOUND		要求されたロックアップキーは、有効なアクティベーションコンテキストで見つかりませんでした。
13018	0x000032DA	ERROR_IPSEC_MM_FILTER_PENDING_DELETION		メインモードフィルタは削除待ちです。
13019	0x000032DB	ERROR_IPSEC_TRANSPORT_FILTER_PENDING_DELETION		転送フィルタは削除待ちです。
13020	0x000032DC	ERROR_IPSEC_TUNNEL_FILTER_PENDING_DELETION		トンネルフィルタは削除待ちです。
13021	0x000032DD	ERROR_IPSEC_MM_POLICY_PENDING_DELETION		メインモードポリシーは削除待ちです。
13022	0x000032DE	ERROR_IPSEC_MM_AUTH_PENDING_DELETION		メインモード認証バンドルは削除待ちです。
13023	0x000032DF	ERROR_IPSEC_QM_POLICY_PENDING_DELETION		クイックモードポリシーは削除待ちです。
13801	0x000035E9	ERROR_IPSEC_IKE_AUTH_FAIL		IKE認証資格情報が許可できません。
13802	0x000035EA	ERROR_IPSEC_IKE_ATTRIB_FAIL		IKEセキュリティ属性が許可できません。
13803	0x000035EB	ERROR_IPSEC_IKE_NEGOTIATION_PENDING		IKEネゴシエーションが進行中です。
13804	0x000035EC	ERROR_IPSEC_IKE_GENERAL_PROCESSING_ERROR		一般的な処理エラー。
13805	0x000035ED	ERROR_IPSEC_IKE_TIMED_OUT		ネゴシエーションがタイムアウトしました。
13806	0x000035EE	ERROR_IPSEC_IKE_NO_CERT		IKEが有効なマシン証明書を見つけることができませんでした。
13807	0x000035EF	ERROR_IPSEC_IKE_SA_DELETED		接続が確立する前にピアによってIKE SAが削除されました。
13808	0x000035F0	ERROR_IPSEC_IKE_SA_REAPED		接続が確立する前にIKE SAが削除されました。

		エラー	説明
10進数	16進数	名前	
13809	0x000035F1	ERROR_IPSEC_IKE_MM_ACQUIRE_DROP	ネゴシエーションリクエストがキューに待機する時間が長すぎます。
13810	0x000035F2	ERROR_IPSEC_IKE_QM_ACQUIRE_DROP	ネゴシエーションリクエストがキューに待機する時間が長すぎます。
13811	0x000035F3	ERROR_IPSEC_IKE_QUEUE_DROP_MM	ネゴシエーションリクエストがキューに待機する時間が長すぎます。
13812	0x000035F4	ERROR_IPSEC_IKE_QUEUE_DROP_NO_MM	ネゴシエーションリクエストがキューに待機する時間が長すぎます。
13813	0x000035F5	ERROR_IPSEC_IKE_DROP_NO_RESPONSE	ピアからの応答がありません。
13814	0x000035F6	ERROR_IPSEC_IKE_MM_DELAY_DROP	ネゴシエーションに長くかかり過ぎました。
13815	0x000035F7	ERROR_IPSEC_IKE_QM_DELAY_DROP	ネゴシエーションに長くかかり過ぎました。
13816	0x000035F8	ERROR_IPSEC_IKE_ERROR	不明なエラーが発生しました。
13817	0x000035F9	ERROR_IPSEC_IKE_CRL_FAILED	証明書の認定取り消しチェックに失敗しました。
13818	0x000035FA	ERROR_IPSEC_IKE_INVALID_KEY_USAGE	無効な証明書キーの使用。
13819	0x000035FB	ERROR_IPSEC_IKE_INVALID_CERT_TYPE	無効な証明書タイプ。
13820	0x000035FC	ERROR_IPSEC_IKE_NO_PRIVATE_KEY	マシン証明書に関連付けられた秘密鍵がありません。
13822	0x000035FE	ERROR_IPSEC_IKE_DH_FAIL	Diffie-Helman計算の失敗。
13824	0x00003600	ERROR_IPSEC_IKE_INVALID_HEADER	無効なヘッダ
13825	0x00003601	ERROR_IPSEC_IKE_NO_POLICY	ポリシーは設定されていません。
13826	0x00003602	ERROR_IPSEC_IKE_INVALID_SIGNATURE	シグネチャの検証に失敗しました。
13827	0x00003603	ERROR_IPSEC_IKE_KERBEROS_ERROR	Kerberosを使用する認証に失敗しました。
13828	0x00003604	ERROR_IPSEC_IKE_NO_PUBLIC_KEY	ピアの証明書には公開鍵がありませんでした。
13829	0x00003605	ERROR_IPSEC_IKE_PROCESS_ERR	エラー処理errorペイロード。
13830	0x00003606	ERROR_IPSEC_IKE_PROCESS_ERR_SA	エラー処理SAペイロード。
13831	0x00003607	ERROR_IPSEC_IKE_PROCESS_ERR_PROP	エラー処理Proposalペイロード。
13832	0x00003608	ERROR_IPSEC_IKE_PROCESS_ERR_TRANS	エラー処理Transformペイロード。
13833	0x00003609	ERROR_IPSEC_IKE_PROCESS_ERR_KEY	エラー処理KEペイロード。
13834	0x0000360A	ERROR_IPSEC_IKE_PROCESS_ERR_ID	エラー処理IDペイロード。
13835	0x0000360B	ERROR_IPSEC_IKE_PROCESS_ERR_CERT	エラー処理Certペイロード。
13836	0x0000360C	ERROR_IPSEC_IKE_PROCESS_ERR_CERT_REQ	エラー処理Certificate Requestペイロード。
13837	0x0000360D	ERROR_IPSEC_IKE_PROCESS_ERR_HASH	エラー処理Hashペイロード。
13838	0x0000360E	ERROR_IPSEC_IKE_PROCESS_ERR_SIG	エラー処理Signatureペイロード。
13839	0x0000360F	ERROR_IPSEC_IKE_PROCESS_ERR_NONCE	エラー処理Nonceペイロード。
13840	0x00003610	ERROR_IPSEC_IKE_PROCESS_ERR_NOTIFY	エラー処理Notifyペイロード。
13841	0x00003611	ERROR_IPSEC_IKE_PROCESS_ERR_DELETE	エラー処理Deleteペイロード。
13842	0x00003612	ERROR_IPSEC_IKE_PROCESS_ERR_VENDOR	エラー処理Vendor Idペイロード。
13843	0x00003613	ERROR_IPSEC_IKE_INVALID_PAYLOAD	無効なペイロードを受信しました。
13844	0x00003614	ERROR_IPSEC_IKE_LOAD_SOFT_SA	ソフトSAがロードされました。
13845	0x00003615	ERROR_IPSEC_IKE_SOFT_SA_TORN_DOWN	ソフトSAが破壊されました。
13846	0x00003616	ERROR_IPSEC_IKE_INVALID_COOKIE	無効なクッキーを受信しました。
13847	0x00003617	ERROR_IPSEC_IKE_NO_PEER_CERT	ピアが有効なマシン証明書の送信に失敗しました。
13848	0x00003618	ERROR_IPSEC_IKE_PEER_CRL_FAILED	ピアの証明書の認定取り消しチェックに失敗しました。
13849	0x00003619	ERROR_IPSEC_IKE_POLICY_CHANGE	新しいポリシーが古いポリシーで構成されたSAを無効にしました。
13850	0x0000361A	ERROR_IPSEC_IKE_NO_MM_POLICY	使用可能なメインモードIKEポリシーがありません。
13851	0x0000361B	ERROR_IPSEC_IKE_NOTCBPRIV	TCB権限の有効化に失敗しました。
13852	0x0000361C	ERROR_IPSEC_IKE_SECLDFAIL	SECURITY.DLLのロードに失敗しました。
13853	0x0000361D	ERROR_IPSEC_IKE_FAILSSPINIT	SSPIからのセキュリティ機能テーブル送付アドレスの取得に失敗しました。
13854	0x0000361E	ERROR_IPSEC_IKE_FAILQUERYSSP	最大トークンサイズを取得するKerberosパッケージの問合せに失敗しました。
13855	0x0000361F	ERROR_IPSEC_IKE_SRVACQFAIL	ISAKMP/ERROR_IPSEC_IKEサービスのためのKerberosサーバ資格情報の取得に失敗しました。Kerberos認証は機能しません。最も考えられるそ

エラー		名前	説明
10進数	16進数		
			の理由は、ドメインのメンバーシップの欠如です。コンピュータがワークグループのメンバである場合、これは正常です。
13856	0x00003620	ERROR_IPSEC_IKE_SRVQUERYCRED	ISAKMP/ERROR_IPSEC_IKEサービス (QueryCredentialsAttributes) のためのSSPIプリンシパル名が決定できませんでした。
13857	0x00003621	ERROR_IPSEC_IKE_GETSPIFAIL	Ipssecドライバから着信SA用の新しいSPIドライバの取得に失敗しました。最も一般的なこの理由は、ドライバに正しいフィルタが設定されていないことです。ポリシーをチェックしてフィルタを確認してください。
13858	0x00003622	ERROR_IPSEC_IKE_INVALID_FILTER	受け取ったフィルタが無効です。
13859	0x00003623	ERROR_IPSEC_IKE_OUT_OF_MEMORY	メモリ割り当てに失敗しました。
13860	0x00003624	ERROR_IPSEC_IKE_ADD_UPDATE_KEY_FAILED	セキュリティアソシエーションをIPSecドライバに追加できませんでした。最も一般的なこの理由は、IKEネゴシエーションの完了に時間が長くなりすぎるということです。問題が解決しない場合は、障害が発生するマシンの負荷を減らしてください。
13861	0x00003625	ERROR_IPSEC_IKE_INVALID_POLICY	無効なポリシー
13862	0x00003626	ERROR_IPSEC_IKE_UNKNOWN_DOI	無効なDOI
13863	0x00003627	ERROR_IPSEC_IKE_INVALID_SITUATION	無効なシチュエーション
13864	0x00003628	ERROR_IPSEC_IKE_DH_FAILURE	Diffie-Hellman障害
13865	0x00003629	ERROR_IPSEC_IKE_INVALID_GROUP	無効なDiffie-Hellmanグループ。
13866	0x0000362A	ERROR_IPSEC_IKE_ENCRYPT	ペイロードの暗号化エラー。
13867	0x0000362B	ERROR_IPSEC_IKE_DECRYPT	ペイロードの復号エラー。
13868	0x0000362C	ERROR_IPSEC_IKE_POLICY_MATCH	ポリシーの一致エラー。
13869	0x0000362D	ERROR_IPSEC_IKE_UNSUPPORTED_ID	サポートされていないID。
13870	0x0000362E	ERROR_IPSEC_IKE_INVALID_HASH	ハッシュ検証に失敗しました。
13871	0x0000362F	ERROR_IPSEC_IKE_INVALID_HASH_ALG	無効なハッシュアルゴリズム。
13872	0x00003630	ERROR_IPSEC_IKE_INVALID_HASH_SIZE	無効なハッシュサイズ
13873	0x00003631	ERROR_IPSEC_IKE_INVALID_ENCRYPT_ALG	無効な暗号化アルゴリズム
13874	0x00003632	ERROR_IPSEC_IKE_INVALID_AUTH_ALG	無効な認証アルゴリズム
13875	0x00003633	ERROR_IPSEC_IKE_INVALID_SIG	無効な証明書のシグネチャ。
13876	0x00003634	ERROR_IPSEC_IKE_LOAD_FAILED	ロードに失敗しました。
13877	0x00003635	ERROR_IPSEC_IKE_RPC_DELETE	RPC呼び出しによって削除されました。
13878	0x00003636	ERROR_IPSEC_IKE_BENIGN_REINIT	再初期化を実行するために一時的な状態が作成されました。これは本当の障害ではありません。
13879	0x00003637	ERROR_IPSEC_IKE_INVALID_RESPONDER_LIFETIME_NOTIFY	Responder Lifetime Notifyで受信したライフタイム値が、Windows 2000で設定されている最小値を下回っています。ピアマシン上のポリシーを修正してください。
13880	0x00003638	ERROR_IPSEC_IKE_INVALID_MAJOR_VERSION	受信者はヘッダーで指定されている IKE のバージョンに対応できません。
13881	0x00003639	ERROR_IPSEC_IKE_INVALID_CERT_KEYLEN	設定されたセキュリティ要件に対して証明書のキーの長さが短すぎます。
13882	0x0000363A	ERROR_IPSEC_IKE_MM_LIMIT	MM SAからピアへの最大接続数を超えました。
13883	0x0000363B	ERROR_IPSEC_IKE_NEGOTIATION_DISABLED	IKEがネゴシエーションを無効にするポリシーを受信しました。
13884	0x0000363C	ERROR_IPSEC_IKE_QM_LIMIT	メインモードのクイックモード上限値に達しました。新しいメインモードが開始されます。

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