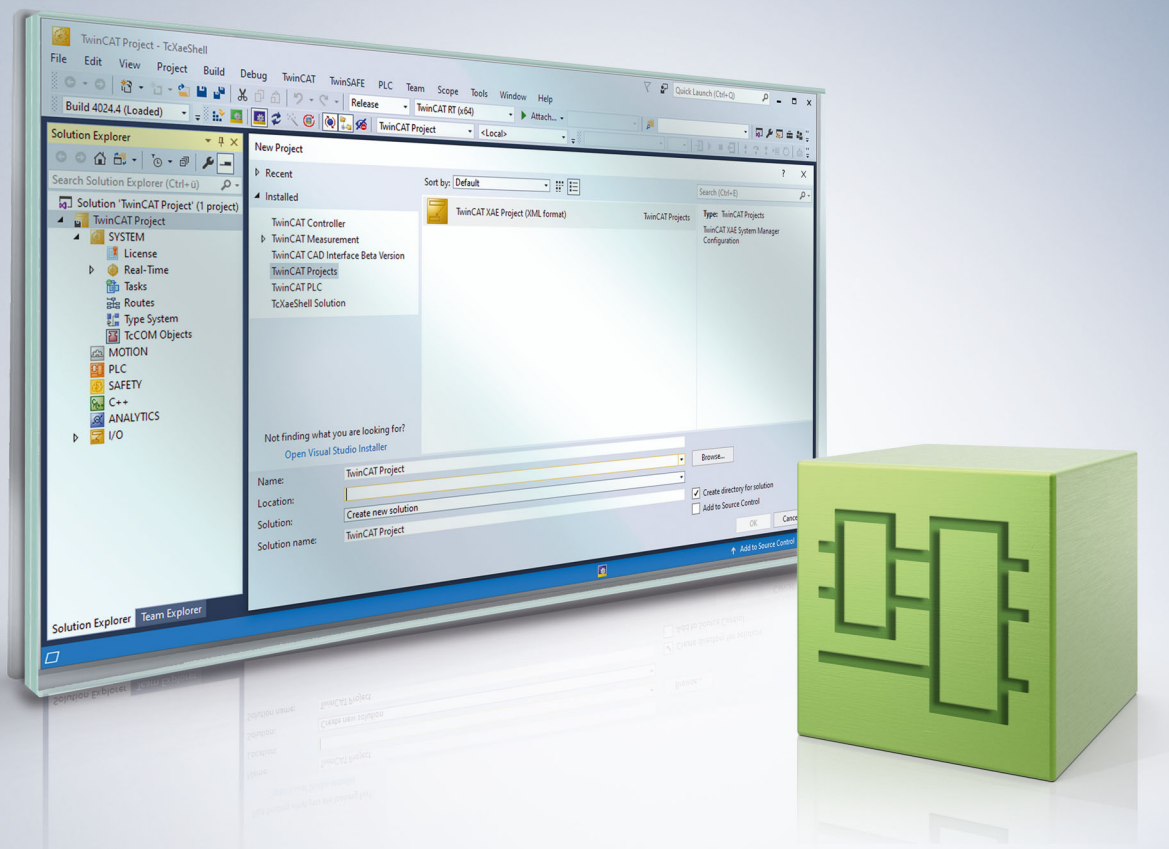
BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | PLC Library: Tc2_Utilities



2023-08-11 | Version: 2.14.1

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.
For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.
We reserve the right to revise and change the documentation at any time and without notice.
No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.
If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
and similar applications and registrations in several other countries.

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

Copyright

1.2 For your safety

Safety regulations

Read the following explanations for your safety.
Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

Exclusion of liability

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

Personnel qualification

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

Signal words

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

Personal injury warnings

⚠ DANGER

Hazard with high risk of death or serious injury.

⚠ WARNING

Hazard with medium risk of death or serious injury.

⚠ CAUTION

There is a low-risk hazard that could result in medium or minor injury.

Warning of damage to property or environment

NOTICE

The environment, equipment, or data may be damaged.

Information on handling the product

This information includes, for example:
recommendations for action, assistance or further information on the product.

1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our https://www.beckhoff.com/secguide.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at https://www.beckhoff.com/secinfo.

2 Overview

The PLC library Tc2_Utilities contains function blocks and functions for calling TwinCAT system functions and operating system functions as well as various conversion functions.

Operating system functions

Name	Description
NT_Shutdown [▶ 124]	Shutting down the operating system (shutdown).
NT_AbortShutdown [▶ 119]	Interrupting the shutdown process.
NT_Reboot [▶ 121]	Carrying out a restart of the operating system.
NT_GetTime [▶ 120]	Reading the current local Windows system time.
NT_SetLocalTime [▶ 122]	Setting the current local Windows system time.
NT_StartProcess [▶ 126]	Starts a Windows application from the PLC.
NT_SetTimeToRTCTime [▶ 123]	Synchronizing the local Windows system time with the real-time clock of the PC.
FB_RegQueryValue [▶ 99]	Reading out values from the registry.
FB_RegSetValue [▶ 102]	Writing values to the Registry.
FB_EnumFindFileEntry [▶ 51]	Searches in a directory for a file or a subdirectory whose name is similar to the specified name. Any entries found can be read individually.
FB_EnumFindFileList [▶ 52]	Searches in a directory for a file or a subdirectory whose name is similar to the specified name. The found entries can be read out in groups.
FB_GetAdaptersInfo [▶ 66]	Reads network adapter information.
FB_GetHostName [▶ 71]	Reads the host name of the local PC.
FB_GetHostAddrByName [▶ 70]	Converts the host name into the (IPv4) Internet protocol network address.
FB_GetTimeZoneInformation [▶ 77]	Reads the time zone configuration of the operating system.

Name	Description
FB_SetTimeZoneInformation [▶ 108]	Sets the time zone configuration of the operating system.
FB_LocalSystemTime [▶ 89]	Returns the local Windows system time and summer time/winter time info.

PLC functions

Name	Description
PLC_Reset [▶ 131]	Resets the PLC.
PLC_Start [▶ 131]	Starts the PLC.
PLC_Stop [▶ 132]	Stops the PLC.
PLC_ReadSymInfo [▶ 127]	Reading the symbol information of the PLC.
PLC_ReadSymInfoByName [▶ 128]	Reading the symbol information of a PLC variable with the help of the symbol name.
PLC_ReadSymInfoByNameEx [▶ 129]	Reading the symbol information of a PLC variable with the help of the symbol name. The comment will be truncated if the available buffer size is insufficient.
Profiler [▶ 133]	Determining the execution time of the PLC code.
WritePersistentData [▶ 142]	Saving the persistent data to the data storage device from the PLC program.
FB_WritePersistentData [▶ 116]	Saving the persistent data to the data storage device from the PLC program (extended version).

Checksum/CRC functions

Name	Description
F_CheckSum16 [▶ 258]	Calculates the 16 bit checksum.
F_DATA_TO_CRC16_CCITT [▶ 260]	Calculates CRC16-CCITT (cyclical redundancy check) of any data type.
F_BYTE_TO_CRC16_CCITT [▶ 257]	Calculates CRC16-CCITT (cyclical redundancy check) of an individual data byte.

System functions

Name	Description
TC_Restart [▶ 140]	Restart the TwinCAT system.
TC_Stop [▶ 141]	Stop the TwinCAT system.
TC_Config [▶ 138]	Switching TwinCAT system to CONFIG mode.
TC_CpuUsage [▶ 139]	Determining CPU usage of the TwinCAT system.
TC_SysLatency [▶ 142]	Determining the current and maximum latency time of a TwinCAT system.
GetRemotePCInfo [▶ 117]	Reading router information via the configured remote PCs.
FB_GetLocalAmsNetId [▶ 75]	Reading the AmsNetId of the local TwinCAT PC.
FB_GetRouterStatusInfo [▶ 76]	Reading TwinCAT router status information.
FB_EnumRouteEntry [▶ 54]	Reading router connection information.
FB_AddRouteEntry [▶ 41]	Adding a new router connection.
FB_RemoveRouteEntry [▶ 104]	Deleting an existing router connection.
ROUTETRANSPORT_TO_STRING [▶ 284]	Converts the AMS message router transport layer ID into a string.
FB_GetDeviceIdentification [▶ 26]	Reads the device ID.
FB_GetDeviceIdentificationEx [▶ 68]	Reads the device ID. (Allows longer strings with hardware model and serial number.)
FB_GetLicences [▶ 73]	Reads the valid and invalid TwinCAT licenses.
FB_GetSystemId [▶ 77]	Reads the SystemID as GUID.
FB_GetVolumeId [▶ 79]	Reads the SystemID and the Volume System ID.

Scope View functions

TwinCAT Scope View functions are not supported by TwinCAT 3.

Scope Server functions

Name	Description
FB_ScopeServerControl [▶ 106]	Controls (start/save..) the Scope Server for data logging.

ADS Monitor functions

Name	Description
FB_AmsLogger [▶ 43]	Start/stop AMS Logger from the PLC.

Converting functions

Name	Description
DT_TO_SYSTEMTIME [▶ 145]	Converting DATE_AND_TIME to Windows system time structure.
DT_TO_FILETIME [▶ 238]	Converting DATE_AND_TIME to Windows file time.
SYSTEMTIME_TO_DT [▶ 155]	Converting Windows system time structure to DATE_AND_TIME.
SYSTEMTIME_TO_FILETIME [▶ 240]	Converting Windows system time structure to file time.
SYSTEMTIME_TO_STRING [▶ 156]	Converting Windows system time structure to string.
STRING_TO_SYSTEMTIME [▶ 154]	Converting string to Windows system time structure.
FILETIME_TO_DT [▶ 239]	Converting Windows file time to DATE_AND_TIME.
FILETIME_TO_SYSTEMTIME [▶ 240]	Converting Windows file time to system time structure.
DEC_TO_BCD [▶ 39]	Converting decimal numbers to BCD numbers.
BCD_TO_DEC [▶ 32]	Converting BCD numbers to decimal numbers.
DEG_TO_RAD [▶ 252]	Converting degree-angle to radian.
RAD_TO_DEG [▶ 283]	Converting radian to degree-angle.
TIME_TO_OTSTRUCT [▶ 157]	Converting TIME variable to a structure with resolved milliseconds, seconds, minutes etc.
OTSTRUCT_TO_TIME [▶ 154]	Converting a structure with resolved milliseconds, seconds, minutes etc. to TIME variable.
F_SwapRealEx [▶ 265]	Replaces by the Hi and Lo Word of a REAL variable.
BYTE_TO_LREALEX [▶ 246]	Allows an explicit conversion of the type BYTE to a positive floating point number of the type LREAL.
DWORD_TO_LREALEX [▶ 256]	Allows an explicit conversion of the type DWORD to a positive floating point number of the type LREAL.
UDINT_TO_LREALEX [▶ 287]	Allows an explicit conversion of the type UDINT to a positive floating point number of the type LREAL.
UINT_TO_LREALEX [▶ 288]	Allows an explicit conversion of the type UINT to a positive floating point number of the type LREAL.
ULINT_TO_ULARGE [▶ 289]	Converts a 64 bit number of the type ULINT into a 64 bit number of the type T_ULARGE_INTEGER.
USINT_TO_LREALEX [▶ 289]	Allows an explicit conversion of the type USINT to a positive floating point number of the type LREAL.
BYTEARR_TO_MAXSTRING [▶ 247]	Converts byte array into a string.
MAXSTRING_TO_BYTEARR [▶ 277]	Converts string into a byte array.
F_TranslateFileTimeBias [▶ 238]	Converts UTC time into local time and vice versa (by bias).
FB_TzSpecificLocalTimeToFileTime [▶ 29]	Converts continuous local time (file time format) into UTC time.
FB_TzSpecificLocalTimeToSystemTime [▶ 115]	Converts continuous local time (structured system time format) into UTC time.
FB_FileTimeToTzSpecificLocalTime [▶ 27]	Converts UTC time (file time format) into local time.
FB_SystemTimeToTzSpecificLocalTime [▶ 111]	Converts UTC time (structured system time format) into local time.
HEXASCNIBBLE_TO_BYTE [▶ 269]	Converts the ASCII code of a hexadecimal half-byte character into a decimal value.
HEXCHRNIBBLE_TO_BYTE [▶ 269]	Converts a hexadecimal half-byte character into its decimal value.
GuidsEqualByVal [▶ 269]	Compares two GUID values

STRING functions

Name	Description
LREAL_TO_FMTSTR [▶ 272]	Converts a flowing point number into a string with the desired number of decimal places.

Name	Description
DWORD_TO_DECSTR [▶ 254]	Converts a decimal number into a decimal string.
DWORD_TO_HEXSTR [▶ 255]	Converts a decimal number into a hexadecimal string.
DWORD_TO_OCTSTR [▶ 257]	Converts a decimal number into an octal string.
DWORD_TO_BINSTR [▶ 253]	Converts a decimal number into a binary string.
LWORD_TO_DECSTR [▶ 275]	Converts a decimal number into a decimal string.
LWORD_TO_HEXSTR [▶ 276]	Converts a decimal number into a hexadecimal string.
LWORD_TO_OCTSTR [▶ 277]	Converts a decimal number into an octal string.
LWORD_TO_BINSTR [▶ 274]	Converts a decimal number into a binary string.
PVOID_TO_DECSTR [▶ 279]	Converts an address (pointer) into a decimal string.
PVOID_TO_HEXSTR [▶ 280]	Converts an address (pointer) into a hexadecimal string.
PVOID_TO_OCTSTR [▶ 281]	Converts an address (pointer) into an octal string.
PVOID_TO_BINSTR [▶ 278]	Converts an address (pointer) into a binary string.
PVOID_TO_STRING [▶ 282]	Converts an address (pointer) into a string.
STRING_TO_PVOID [▶ 286]	Converts a string into an address (pointer).
LINT_TO_DECSTR [▶ 271]	Converts a signed decimal number (64 bit) into a decimal string.
DINT_TO_DECSTR [▶ 252]	Converts a signed decimal number (32 bit) into a decimal string.
F_FormatArgToStr [▶ 261]	Converts and formats a decimal number or a flowing point number into a string.
BYTE_TO_BINSTR [▶ 244]	Converts a decimal number of the type byte into a binary string.
BYTE_TO_DECSTR [▶ 245]	Converts a decimal number into a decimal string.
BYTE_TO_HEXSTR [▶ 245]	Converts a decimal number into a hexadecimal string.
BYTE_TO_OCTSTR [▶ 247]	Converts a decimal number into an octal string.
WORD_TO_BINSTR [▶ 290]	Converts a decimal number of the type word into a binary string.
WORD_TO_DECSTR [▶ 291]	Converts a decimal number of the type word into a decimal string.
WORD_TO_HEXSTR [▶ 291]	Converts a decimal number of the type word into a hexadecimal string.
WORD_TO_OCTSTR [▶ 292]	Converts a decimal number of the type word into an octal string.
FB_FormatString [▶ 64]	Converts and formats up to 10 arguments (decimal or flowing point numbers).
FB_EnumStringNumbers [▶ 56]	Searches a string for numbers.
F_ToUCase [▶ 267]	Converts lowercase letters into uppercase letters in a string.
F_ToLCase [▶ 266]	Converts uppercase letters into lowercase letters in a string.
F_LTrim [▶ 264]	Removes spaces at the beginning of a string.
F_RTrim [▶ 264]	Removes spaces at the end of a string.
DATA_TO_HEXSTR [▶ 250]	Converts binary data into hexadecimal string.
HEXSTR_TO_DATA [▶ 270]	Converts hexadecimal string to binary data.
GUID_TO_STRING [▶ 268]	Converts a structured GUID variable into a GUID string variable.
GUID_TO_REGSTRING [▶ 268]	Converts a structured GUID variable into a registry GUID string variable
REGSTRING_TO_GUID [▶ 283]	Converts a registry GUID string variable into a structured GUID variable.

Extended STRING functions

Name	Description
CHAR_TO_WCHAR [▶ 158]	Converts a character of the data type STRING into a character of the data type WSTRING (with null termination).
CONCAT2 [▶ 158]	Concatenates two strings of the data type STRING.
DELETE2 [▶ 159]	Removes nLen characters from a string, starting at position nPos.
F_StringIsASCII [▶ 160]	Checks whether a string contains only ASCII characters (0x000 to 0x7F) and returns the number of ASCII characters.
FIND2 [▶ 161]	Finds a string, which may occur more than once, in another string.
FindAndDelete [▶ 161]	Finds a string, which may occur more than once, in another string and removes it.
FindAndDeleteChar [▶ 162]	Finds a character, which may occur more than once, in a string and removes it.

Name	Description
FindAndReplace [▶ 163]	Finds a string, which may occur more than once, in another string and replaces it with another string.
FindAndReplaceChar [▶ 163]	Finds a character, which may occur more than once, in a string and replaces it with another character.
INSERT2 [▶ 168]	Inserts a string into another string after position nPos.
LEN2 [▶ 169]	Returns the number of characters in a string.
REPLACE2 [▶ 169]	Replaces nLen characters of a string with another string, starting at position nPos.
sLiteral_TO_UTF8 [▶ 170]	Converts any character string of the data type STRING into a character string in UTF-8 format.
STRING_TO_UTF8 [▶ 171]	Converts any string of a variable of the data type STRING to a string in UTF-8 format.
STRING_TO_WSTRING2 [▶ 172]	Converts a variable of the data type STRING into a variable of the data type WSTRING.
STRNCPY [▶ 172]	Copies the character string of a variable of the data type STRING and checks whether the character string was completely copied
UTF8_TO_STRING [▶ 173]	Converts a string in UTF-8 format into a string of the data type STRING.
UTF8_TO_WSTRING [▶ 174]	Converts a string in UTF-8 format into a string of the data type WSTRING.
UTF8Len [▶ 175]	Returns the number of characters in a UTF-8 string.
WCHAR_TO_CHAR [▶ 176]	Converts a variable of the data type WSTRING into a variable of the data type STRING (with null termination)
WCONCAT2 [▶ 176]	Concatenates two strings of the data type WSTRING of any length.
WLEN2 [▶ 177]	Returns the number of characters in a Unicode string of the data type WSTRING.
wsLiteral_TO_UTF8 [▶ 177]	Converts any character string of the data type WSTRING into a character string in UTF-8 format.
WSTRING_TO_STRING2 [▶ 178]	Converts a variable of the data type WSTRING into a variable of the data type STRING.
WSTRING_TO_UTF8 [▶ 178]	Converts a string of a variable of the data type WSTRING into a string in UTF-8 format.
WSTRNCPY [▶ 179]	Copies a character string of a variable of the data type WSTRING and checks whether the character string was completely copied.

64 bit functions (unsigned)

Name	Description
ULARGE_INTEGER [▶ 224]	Initializes/sets a 64 bit number.
UInt64Add64 [▶ 215]	Adds two 64 bit numbers.
UInt64Add64Ex [▶ 215]	Adds two 64 bit numbers (with Overflow check).
UInt64Sub64 [▶ 223]	Subtracts two 64 bit numbers.
UInt64Cmp64 [▶ 216]	Compares two 64 bit numbers.
UInt32x32To64 [▶ 214]	Multiplies two 32 bit numbers. The result is a 64 bit number.
UInt64Mul64 [▶ 220]	Multiplies two 64 bit numbers. The result is a 64 bit number.
UInt64Mul64Ex [▶ 221]	Multiplies two 64 bit numbers. The result is a 64 bit number (with overflow check).
UInt64Div64 [▶ 217]	Division of two 64 bit numbers.
UInt64Div64Ex [▶ 218]	Division of two 64 bit numbers (with remainder).
Uint64Div16Ex [▶ 217]	Division of one 64 bit number by a 16 bit number. The result is a 64 bit number.
UInt64Mod64 [▶ 220]	Modulo division of two 64 bit numbers.
UInt64And [▶ 216]	Bit by bit AND of two 64 bit numbers.
UInt64Or [▶ 221]	Bit by bit OR of two 64 bit numbers.
UInt64Not [▶ 221]	Bit by bit NOT of one 64 bit number.
UInt64Xor [▶ 224]	Bit by bit XOR of two 64 bit numbers.
UInt64Rol [▶ 222]	Bit by bit left rotation of one 64 bit number.
UInt64Ror [▶ 222]	Bit by bit right rotation of one 64 bit number.
UInt64Shl [▶ 223]	Bit by bit left shift of one 64 bit number.
UInt64Shr [▶ 223]	Bit by bit right shift of one 64 bit number.

Name	Description
UInt64Min [▶ 219]	Minimum function
UInt64Max [▶ 219]	Maximum function
UInt64Limit [▶ 219]	Limitation
UInt64isZero [▶ 218]	Checks whether the value of the 64 bit number is zero.
UINT64_TO_STRING [▶ 215]	Converts 64 bit number into a string.
UINT64_TO_LREAL [▶ 214]	Converts 64 bit number into a LREAL.
STRING_TO_UINT64 [▶ 213]	Converts a STRING into a 64 bit number.
LREAL_TO_UINT64 [▶ 213]	Converts LREAL into a 64 bit number.
LWORD_TO_ULARGE [▶ 213]	Converts a 64 bit number of the type LWORD into a 64 bit number of the type T_ULARGE_INTEGER.
ULARGE_TO_LWORD [▶ 225]	Converts a 64 bit number of the type T_ULARGE_INTEGER into a 64 bit number of the type LWORD.
ULARGE_TO_ULINT [▶ 225]	Converts a 64 bit number of the type T_ULARGE_INTEGER into a 64 bit number ULINT.

64 bit functions (signed)

Name	Description
LARGE_INTEGER [▶ 210]	Initializes/sets a 64 bit number.
Int64Add64 [▶ 207]	Adds two 64 bit numbers.
Int64Add64Ex [▶ 207]	Adds two 64 bit numbers (with Overflow check).
Int64Sub64 [▶ 210]	Subtracts two 64 bit numbers.
Int64Cmp64 [▶ 208]	Compares two 64 bit numbers.
Int64Div64Ex [▶ 208]	Division of two 64 bit numbers (with remainder).
Int64Not [▶ 209]	Bit by bit NOT of one 64 bit number.
Int64isZero [▶ 209]	Checks whether the value of the 64 bit number is zero.
Int64Negate [▶ 209]	Negates a 64 bit number.
INT64_TO_LREAL [▶ 206]	Converts 64 bit number into a LREAL.
LREAL_TO_INT64 [▶ 212]	Converts LREAL into a 64 bit number.
LARGE_TO_ULARGE [▶ 211]	Converts a signed 64 bit number into an unsigned 64 bit number.
ULARGE_TO_LARGE [▶ 212]	Converts an unsigned 64 bit number into a signed 64 bit number.
LARGE_TO_LINT [▶ 211]	Converts a signed 64 bit number of the type LINT into a signed 64 bit number of the type T_LARGE_INTEGER.
LINT_TO_LARGE [▶ 211]	Converts a signed 64 bit number of the type T_LARGE_INTEGER into a signed 64 bit number of the type LINT.

16 bit fixed point number functions (signed)

Name	Description
FIX16Add [▶ 202]	Adds two fixed-point numbers.
FIX16Align [▶ 202]	Changes the resolution of a fixed-point number.
FIX16Sub [▶ 204]	Subtracts two fixed-point numbers.
FIX16Div [▶ 203]	Divides two fixed-point numbers.
FIX16Mul [▶ 204]	Multiplies two fixed-point numbers.
LREAL_TO_FIX16 [▶ 205]	Converts LREAL into a fixed-point number.
WORD_TO_FIX16 [▶ 206]	Converts WORD into a fixed-point number.
FIX16_TO_LREAL [▶ 201]	Converts a fixed-point number into LREAL.
FIX16_TO_WORD [▶ 201]	Converts a fixed-point number into WORD.

Byte order converting functions

Name	Description
HOST_TO_BE16 [▶ 180]	Host-to-network converting (16 bit number)
HOST_TO_BE32 [▶ 181]	Host-to-network converting (32 bit number)
HOST_TO_BE64 [▶ 181]	Host-to-network converting (64 bit number, „legacy“-type: T_ULARGE_INTEGER)

Name	Description
HOST_TO_BE64EX [▶ 182]	Host-to-network converting (64 bit number, „native“-type: LWORD)
HOST_TO_BE128 [▶ 182]	Host-to-network converting (128 bit number, „legacy“-type: T_UHUGE_INTEGER)
BE16_TO_HOST [▶ 183]	Host-to-network converting (16 bit number)
BE32_TO_HOST [▶ 183]	Network-to-host converting (32 number)
BE64_TO_HOST [▶ 183]	Network-to-host converting (64 bit number, „legacy“-type: T_ULARGE_INTEGER)
BE64_TO_HOSTEX [▶ 184]	Network-to-host converting (64 bit number, „native“-type: LWORD)
BE128_TO_HOST [▶ 184]	Network-to-host converting (128 bit number, „legacy“-type: T_UHUGE_INTEGER)

FLOAT functions

Name	Description
BOOL_TO_FLOAT [▶ 185]	Converts a variable of the type BOOL to a variable of the type LREAL.
DINT_TO_FLOAT [▶ 185]	Converts a variable of the type DINT to a variable of the type FLOAT.
FLOAT_TO_BOOL [▶ 185]	Converts a variable of the type FLOAT to a variable of the type BOOL.
FLOAT_TO_DINT [▶ 186]	Converts a variable of the type FLOAT to a variable of the type DINT.
FLOAT_TO_INT [▶ 186]	Converts a variable of the type FLOAT to a variable of the type INT.
FLOAT_TO_SINT [▶ 186]	Converts a variable of the type FLOAT to a variable of the type SINT.
FLOAT_TO_STRING [▶ 186]	Converts a variable of the type FLOAT to a variable of the type STRING.
FLOAT_TO_TIME [▶ 187]	Converts a variable of the type FLOAT to a variable of the type TIME.
FLOAT_TO_UDINT [▶ 187]	Converts a variable of the type FLOAT to a variable of the type UDINT.
FLOAT_TO_UINT [▶ 187]	Converts a variable of the type FLOAT to a variable of the type UINT.
INT_TO_FLOAT [▶ 188]	Converts a variable of the type INT to a variable of the type FLOAT.
SINT_TO_FLOAT [▶ 188]	Converts a variable of the type SINT to a variable of the type FLOAT.
TIME_TO_FLOAT [▶ 188]	Converts a variable of the type TIME to a variable of the type FLOAT.
UDINT_TO_FLOAT [▶ 188]	Converts a variable of the type UDINT to a variable of the type FLOAT.
UINT_TO_FLOAT [▶ 189]	Converts a variable of the type UINT to a variable of the type FLOAT.
LrealIsFinite [▶ 189]	Returns TRUE, when the argument of the type LREAL has a finite value.
LrealIsNaN [▶ 189]	Returns TRUE, when the argument of the type LREAL has an undefined value (NaN).

LCOMPLEX functions

LcomplexIsNaN [▶ 191]	Returns TRUE, when the argument of the type LCOMPLEX has an undefined value (NaN).
LcomplexAbs [▶ 191]	Returns the absolute value of the complex number transferred.

P[TYPE]_TO_[TYPE] converting functions

Name	Description
PBOOL_TO_BOOL [▶ 192]	Returns the content of a BOOL pointer variable.
PBYTE_TO_BYTE [▶ 192]	Returns the content of a BYTE pointer variable.
PDATE_TO_DATE [▶ 192]	Returns the content of a DATE pointer variable.
PDINT_TO_DINT [▶ 193]	Returns the content of a DINT pointer variable.
PDT_TO_TO_DT [▶ 193]	Returns the content of a DT pointer variable.

Name	Description
PDWORD_TO_DWORD [▶ 193]	Returns the content of a DWORD pointer variable.
PHUGE_TO_HUGE [▶ 194]	Returns the content of a T_HUGE_INTEGER pointer variable.
PINT_TO_INT [▶ 194]	Returns the content of an INT pointer variable.
PLARGE_TO_LARGE [▶ 194]	Returns the content of a T_LARGE_INTEGER pointer variable.
PLINT_TO_LINT [▶ 195]	Returns the content of a LINT pointer variable.
PLREAL_TO_TO_LREAL [▶ 195]	Returns the content of a LREAL pointer variable.
PLWORD_TO_LWORD [▶ 195]	Returns the content of a LWORD pointer variable.
PMAXSTRING_TO_MAXSTRING [▶ 196]	Returns the content of a T_MaxString pointer variable.
PREAL_TO_REAL [▶ 196]	Returns the content of a REAL pointer variable.
PSINT_TO_SINT [▶ 197]	Returns the content of a SINT pointer variable.
PSTRING_TO_STRING [▶ 197]	Returns the content of a STRING pointer variable.
PTIME_TO_TIME [▶ 197]	Returns the content of a TIME pointer variable.
PTOD_TO_TOD [▶ 198]	Returns the content of a TOD pointer variable.
PUDINT_TO_UDINT [▶ 198]	Returns the content of an UDINT pointer variable.
PUHUGE_TO_UHUGE [▶ 198]	Returns the content of a T_UHUGE_INTEGER pointer variable.
PUINT_TO_UINT [▶ 199]	Returns the content of an UINT pointer variable.
PULARGE_TO_ULARGE [▶ 199]	Returns the content of a T_ULARGE_INTEGER pointer variable.
PULINT_TO_ULINT [▶ 199]	Returns the content of an ULINT pointer variable.
PUSINT_TO_USINT [▶ 200]	Returns the content of an USINT pointer variable.
PWORD_TO_WORD [▶ 200]	Returns the content of a WORD pointer variable.
PUINT64_TO_UINT64 [▶ 200]	Returns the content of a T_ULARGE_INTEGER pointer variable.

T_Arg help functions

Name	Description
F_ARGCMP [▶ 226]	Compares two variables of the type T_Arg
F_ARGCPY [▶ 226]	Copies the value of a variable of the type T_Arg to another variable and returns the number of successfully copied data bytes.
F_ARGISZERO [▶ 227]	Returns TRUE if one of the T_Arg member variables has the value zero or was not initialised.
F_BIGTYPE [▶ 227]	Returns information on a struct or array variable in a structure of the type T_Arg.
F_BOOL [▶ 228]	Returns information on a BOOL variable in a structure of the type T_Arg.
F_BYTE [▶ 228]	Returns information on a BYTE variable in a structure of the type T_Arg.
F_DINT [▶ 229]	Returns information on a DINT variable in a structure of the type T_Arg.
F_DWORD [▶ 229]	Returns information on a DWORD variable in a structure of the type T_Arg.
F_HUGE [▶ 229]	Returns information on a T_HUGE_INTEGER variable in a structure of the type T_Arg.
F_INT [▶ 230]	Returns information on an INT variable in a structure of the type T_Arg.
F_LARGE [▶ 230]	Returns information on a T_LARGE_INTEGER variable in a structure of the type T_Arg.
F_LINT [▶ 231]	Returns information on a LINT variable in a structure of the type T_Arg.
F_LREAL [▶ 231]	Returns information on a LREAL variable in a structure of the type T_Arg.
F_LWORD [▶ 231]	Returns information on a LWORD variable in a structure of the type T_Arg.
F_REAL [▶ 232]	Returns information on a REAL variable in a structure of the type T_Arg.
F_SINT [▶ 232]	Returns information on a SINT variable in a structure of the type T_Arg.
F_STRING [▶ 232]	Returns information on a T_MaxString variable in a structure of the type T_Arg.
F_UDINT [▶ 233]	Returns information on an UDINT variable in a structure of the type T_Arg.

Name	Description
F_UHUGE [▶ 233]	Returns information on a T_UHUGE_INTEGER variable in a structure of the type T_Arg.
F_UINT [▶ 234]	Returns information on an UINT variable in a structure of the type T_Arg.
F_ULARGE [▶ 234]	Returns information on a T_ULARGE_INTEGER variable in a structure of the type T_Arg.
F_ULINT [▶ 235]	Returns information on an ULINT variable in a structure of the type T_Arg.
F_USINT [▶ 235]	Returns information on an USINT variable in a structure of the type T_Arg.
F_WORD [▶ 235]	Returns information on a WORD variable in a structure of the type T_Arg.
F_PVOID [▶ 236]	Returns information on a PVOID variable in a structure of the type T_Arg.

CSV format functions

Name	Description
CSVFIELD_TO_STRING [▶ 249]	Converts the value of a string with a data field in CSV format into an PLC string variable.
STRING_TO_CSVFIELD [▶ 284]	Converts the value of a PLC string variable into a string with a data field in CSV format.
CSVFIELD_TO_ARG [▶ 248]	Converts a byte buffer with a data field in CSV format into a value of a random PLC variable.
ARG_TO_CSVFIELD [▶ 242]	Converts the value of a random PLC variable into a byte buffer with a data field in CSV format.
FB_CSVMemBufferReader [▶ 48]	Splits data sets in CSV format which are in a byte buffer into individual data fields.
FB_CSVMemBufferWriter [▶ 50]	Generates individual or several data sets in a byte buffer from individual data fields

Licensing functions

Name	Description
FB_LicFileGetStorageInfo [▶ 85]	Reads the StorageInfo of the license dongle and the file directory.
FB_LicFileCreate [▶ 83]	Creates a file on the license dongle.
FB_LicFileDelete [▶ 84]	Deletes a file from the license dongle.
FB_LicFileRead [▶ 86]	Reads a file from the license dongle to a buffer.
FB_LicFileCopyToDongle [▶ 82]	Copies a file from the hard disk to the license dongle.
FB_LicFileCopyFromDongle [▶ 81]	Copies a file from the license dongle to the hard disk.
FB_CheckLicense [▶ 48]	Determines the TwinCAT 3 license status for a given license ID.
FB_GetDongleSystemId [▶ 69]	Reads the system ID and the volume ID of the TwinCAT 3 license dongle as GUID.
FB_GetLicenseDongle [▶ 72]	Determines the number of connected license dongles and returns address and status.
FB_GetLicenses [▶ 73]	Reads the valid and invalid TwinCAT licenses.
FB_GetLicensesEx [▶ 74]	Determines the status of all TwinCAT 3 licenses and OEM licenses.

Other functions

Name	Description
FB_BasicPID [▶ 45]	Simple PID controller
F_GetVersionTcUtilities [▶ 241]	Reading version information of the library.
IsFinite [▶ 236]	Verifies the formatting of a floating point number in accordance with the IEEE.
F_YearIsLeapYear [▶ 151]	Determines whether a year is a leap year.
F_GetMaxMonthDays [▶ 148]	Determines the maximum number of days in a month.
F_GetDOYOfYearMonthDay [▶ 147]	Determines the number of the day in the year.
F_GetMonthOfDOY [▶ 149]	Determines the month based on the day number in the year.
F_GetDayOfWeek [▶ 147]	Determines the number of the weekday.

Name	Description
F_GetWeekOfTheYear [▶ 149]	Determines the calendar week.
F_GetDayOfMonthEx [▶ 146]	Determines the date of the first, second etc. weekday in a specified month and year.
F_GetWeekOfTheYear [▶ 149]	Returns the number of the calendar week for a predefined date.
RTC [▶ 135]	"Software"-RTC (Real Time Clock)
RTC_EX [▶ 136]	"Software"-RTC (Real Time Clock)
RTC_EX2 [▶ 137]	"Software"-RTC (Real Time Clock)
FB_FileRingBuffer [▶ 59]	Writes/reads data sets to or from the file (FIFO).
FB_MemRingBuffer [▶ 94]	Writes/reads data sets to or from a buffer variable (FIFO).
FB_MemRingBufferEx [▶ 96]	Writes/reads data sets to or from a buffer variable (FIFO).
FB_StringRingBuffer [▶ 110]	Writes/reads strings to or from a buffer variable (FIFO).
FB_MemStackBuffer [▶ 97]	Writes/reads data sets to or from a buffer variable (LIFO).
FB_MemBufferMerge [▶ 92]	Merges individual small data segments to one larger data segment.
FB_MemBufferSplit [▶ 93]	Splits a memory area (data buffer) into several smaller segments.
FB_HashTableCtrl [▶ 79], F_CreateHashTableHnd [▶ 258]	Simple hash table.
FB_LinkedListCtrl [▶ 87], F_CreateLinkedListHnd [▶ 259]	Simple linked list (double linked).
DCF77_TIME [▶ 32]	A simple DCF77 decoder.
DCF77_TIME_EX [▶ 35]	DCF77 decoder with plausibility check of two consecutive telegrams and time zone information.

3 Function blocks

3.1 [obsolete]

3.1.1 FB_GetDeviceIdentification

The block reads the device ID.

Obsolete functionality

For longer hardware model and hardware serial number strings the block FB_GetDeviceIdentificationEx [▶ 68] has to be used.

VAR_INPUT

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

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

sNetId: This parameter can be used to specify the AmsNetID of the TwinCAT computer whose device ID is to be read (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: If an error should occur during the transfer of the command, then this output is set once the bBusy output was reset.

nErrorId: Supplies the ADS error number [▶ 353] when the bError output is set.

stDevIdent: Provides the device ID (type: ST_DeviceIdentification [▶ 306])

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.1.2 FB_FileTimeToTzSpecificLocalTime

Outdated function

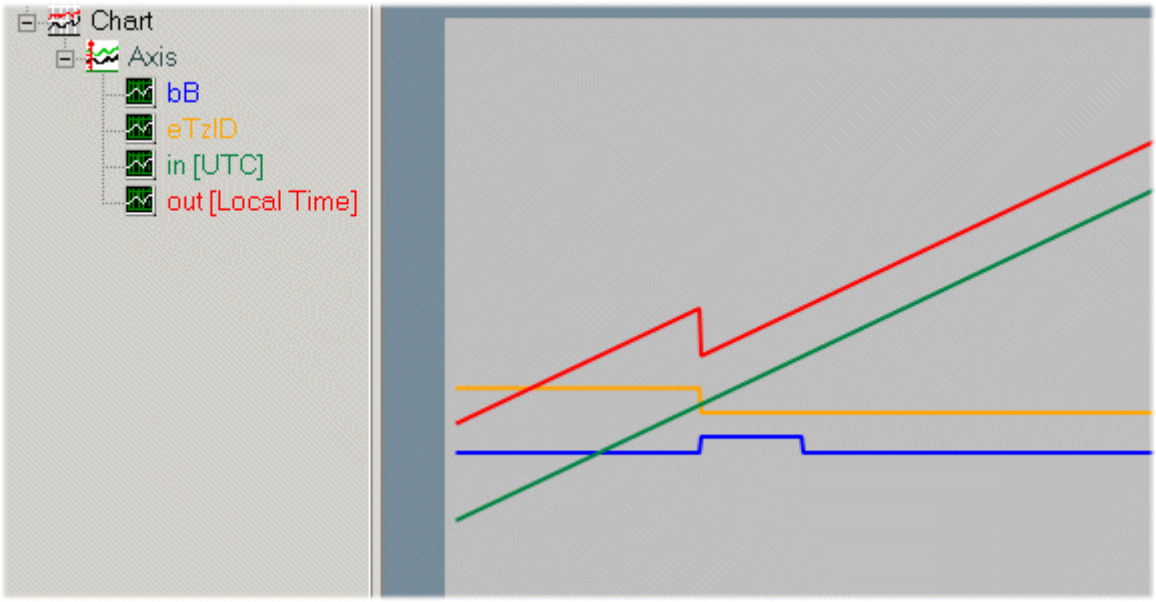
This function block is outdated. Use the function block FB_FileTime64ToTzSpecificLocalTime [▶ 61] instead.

The function block converts the UTC time (file time format) to local time (file time format), taking into account the specified time zone information. The function block: FB_SystemTimeToTzSpecificLocalTime [▶ 111] has a similar function, the difference being that it converts to a different time format (structured system time format).

The function block is only suitable for conversion of **continuous** UTC timestamp information. The function block uses the time zone information to calculate the required time steps (summer/winter time changeover) in local time. Time steps in UTC input time are not permitted and lead to incorrect conversion. The reason: the function block stores the last converted time internally, so that it can detect the B times (see below) from the UTC input time and the stored value when the local time is changed.

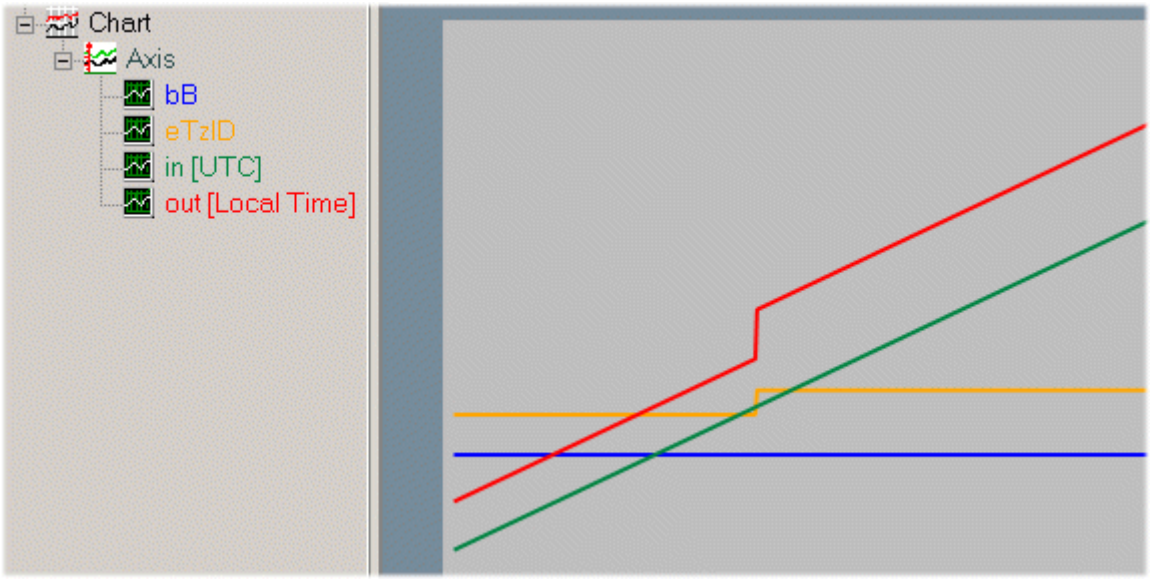
The function block is associated with an action: A_Reset(). If this action is called the function block outputs and the locally stored (last converted) time are reset to zero.

1. Graphic representation of the changeover from summer time to winter time (tzInfo = WEST_EUROPE_TZI):



The UTC input time (green) is continuous. The local time (red) jumps back. The local time: **02h:59m:59s:999ms..** is directly followed by: **02h:00m:00s:000ms..** The times between 2h and 3h occur twice. The duplicate time before the changeover is referred to as **02:05:00 CEST A**, for example, the time after the changeover as **02:05:00 CET B.** The output variable *bB* indicates whether it is the first or the second *pass*. During the second *pass* the *bB* output variable (blue) is set to TRUE. The *bB* output variable is automatically reset once the duplicate time has passed. The time zone ID (orange) changes from *eTimeZoneID_Daylight* (summer time) to *eTimeZoneID_Standard* (winter time).

2. Graphic representation of the change-over from winter time to summer time (tzInfo = WEST_EUROPE_TZI):



The UTC input time (green) is continuous. The local time (green) jumps forward. The local time: **2h:59m:59s:999ms..** is directly followed by: **3h:00m:00s:000ms..** The time zone ID (orange) changes from *eTimeZoneID_Standard* (winter time) to *eTimeZoneID_Daylight* (summer time).

VAR_INPUT

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

in: UTC time (file time format) to be converted (type: T_FILETIME [▶ 317]).

tzInfo: Structure variable with the current time zone information of the operating system (type: ST_TimeZoneInformation [▶ 315]).

VAR_OUTPUT

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

out: Converted local time (file time format, type: T_FILETIME [▶ 317]).

eTzID: Additional summer/winter time information (Type: E_TimeZoneID [▶ 302]).

bB: TRUE => B time (e.g.:**02:05:00 CET B**), FALSE => other time (e.g.: **02:05:00 CEST A**). This output is set if the local time jumps back and is reset once the duplicate local time has passed.

Example:

The UTC time: DT#2011-09-02-09:01:31 is converted to local time. The result is: 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 );
```

Further functions and function blocks for time and time zone:

- FB_TzSpecificLocalTimeToSystemTime [▶ 115]
- FB_TzSpecificLocalTimeToFileTime [▶ 29]

- FB_SystemTimeToTzSpecificLocalTime [▶ 111]
- FB_GetTimeZoneInformation [▶ 77]
- FB_SetTimeZoneInformation [▶ 108]
- NT_SetLocalTime [▶ 122]
- NT_GetTime [▶ 120]
- NT_SetTimeToRTCTime [▶ 123]
- F_TranslateFileTimeBias [▶ 238]
- FB_LocalSystemTime [▶ 89]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.1.3 FB_TzSpecificLocalTimeToFileTime

Outdated function

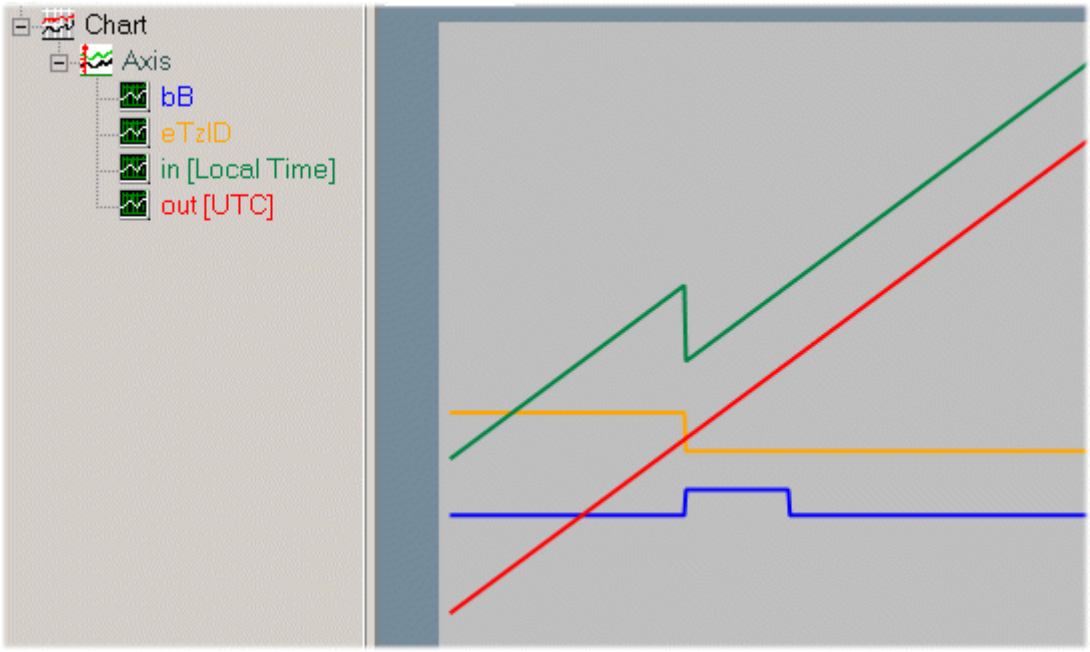
This function block is outdated. Use the function block FB_TzSpecificLocalTimeToFileTime64 [▶ 112] instead.

The function block converts the local time (file time format) to UTC time (file time format), taking into account the specified time zone information. The function block: FB_TzSpecificLocalTimeToSystemTime [▶ 115] has a similar function, the difference being that it converts to a different time format (structured system time format).

The function block is only suitable for conversion of **continuous** local timestamp information. Step changes in local time caused by summer/winter time changeover are permitted and are correctly detected by the function block. Arbitrary changes in local time result in incorrect conversion. The reason: the last converted time is stored internally in function block in order to be able to identify the summer time/winter time information and the B times (see below) when the local time is reset. The function block is associated with an action: A_Reset(). If this action is called the function block outputs and the locally stored (last converted) time are reset to zero.

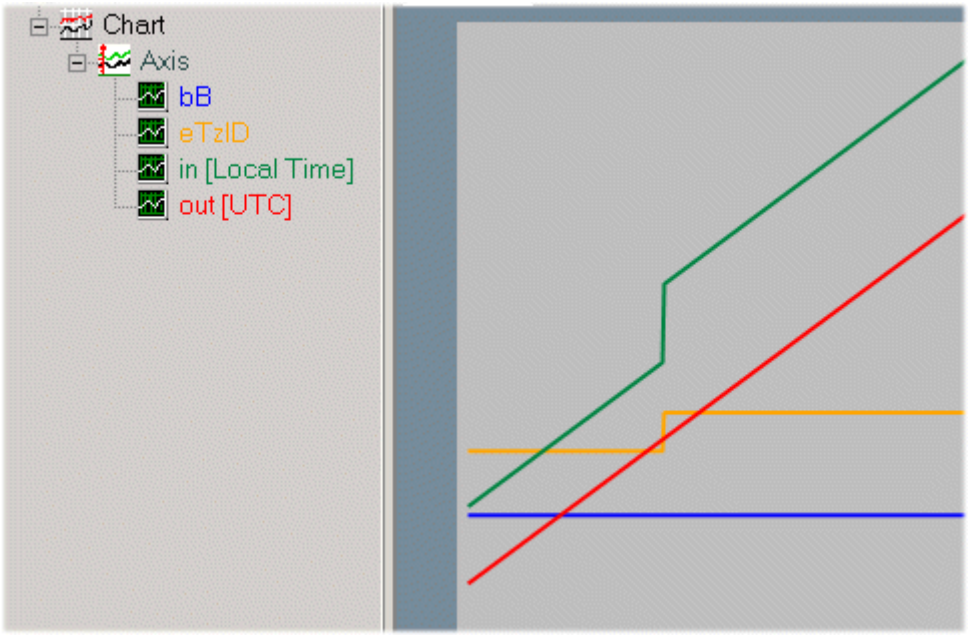
The step changes in the local time are problematic, since they have to be converted to a linear UTC time. It is therefore advisable to use the (continuous) UTC time for time stamping tasks and to convert the time to respective local time only for display purposes (e.g. in a visualization).

1. Graphic representation of the changeover from summer time to winter time (tzInfo = WEST_EUROPE_TZI):



The local time (green) jumps back. The UTC output time (red) continuous. The local time: **02h:59m:59s:999ms..** is directly followed by: **02h:00m:00s:000ms..** The times between 2h and 3h occur twice. The duplicate time before the changeover is referred to as **02:05:00 CEST A**, for example, the time after the changeover as **02:05:00 CET B.** The output variable *bB* indicates whether it is the first or the second *pass*. During the second *pass* the *bB* output variable (blue) is set to TRUE. The *bB* output variable is automatically reset once the duplicate time has passed. The time zone ID (orange) changes from *eTimeZoneID_Daylight* (summer time) to *eTimeZoneID_Standard* (winter time).

2. Graphic representation of the change-over from winter time to summer time (tzInfo = WEST_EUROPE_TZI):



The local time (green) jumps forward. The UTC output time (red) continuous. The local time: **2h:59m:59s:999ms..** is directly followed by: **3h:00m:00s:000ms..** The time zone ID (orange) changes from *eTimeZoneID_Standard* (winter time) to *eTimeZoneID_Daylight* (summer time).

VAR_INPUT

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

in: Local time (file time format) to be converted (type: T_FILETIME [▶ 317]).

tzInfo: Structure variable with the current time zone information of the operating system (type: ST_TimeZoneInformation [▶ 315]).

VAR_OUTPUT

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

out: Converted UTC time (file time format) (type: T_FILETIME [▶ 317]).

eTzID: Additional summer/winter time information (type: E_TimeZoneID [▶ 302]).

bB: TRUE => B time (e.g.:**02:05:00 CET B**), FALSE => other time (e.g.: **02:05:00 CEST A**). This output is set if the local time jumps back and is reset once the duplicate local time has passed.

Example:

The local time: DT#2011-09-02-11:01:31 is converted UTC time: DT#2011-09-02-09:01:31.

```
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 );
```

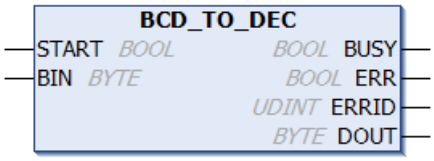
Further functions and function blocks for time and time zone:

- FB_TzSpecificLocalTimeToSystemTime [▶ 115]
- FB_SystemTimeToTzSpecificLocalTime [▶ 111]
- FB_FileTimeToTzSpecificLocalTime [▶ 27]
- FB_GetTimeZoneInformation [▶ 77]
- FB_SetTimeZoneInformation [▶ 108]
- NT_SetLocalTime [▶ 122]
- NT_GetTime [▶ 120]
- NT_SetTimeToRTCTime [▶ 123]
- F_TranslateFileTimeBias [▶ 238]
- FB_LocalSystemTime [▶ 89]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.2 BCD_TO_DEC



The "BCD_TO_DEC" function block allows numbers in BCD to be converted to decimal format. The BCD number to be converted is checked for the reliability of the values.

VAR_INPUT

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

START: The function block is activated by a positive edge at this input.

BIN: The BCD number to be converted.

VAR_OUTPUT

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

BUSY: This input is set at the start of the conversion procedure, and remains set until the conversion has been completed. Once the BUSY output has been reset, the decimal value is available at the DOUT output.

ERR: This variable is set to TRUE if an error occurs.

ERRID: Error code.

DOUT: The converted variable in decimal format is available at this output if the process is successful.

Error Codes:

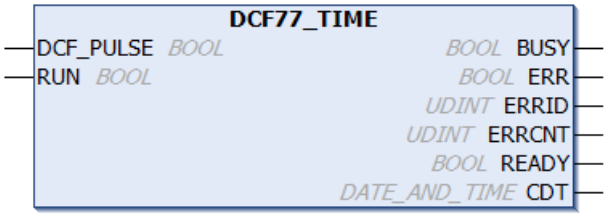
Error code	Error descriptions
0	no error
0x000F	Unreliable value in the low nibble of the BCD number
0x00F0	Unreliable value in the high nibble of the BCD number

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.3 DCF77_TIME

This function block is superseded by the DCF77_TIME_EX [▶ 35] function block.



The "DCF77_TIME" function block can be used to decode the DCF-77 radio clock signal. A rising edge at the RUN input starts the decoding process, which continues as long as the RUN input remains set. In the worst case, the function block requires a maximum of 1 minute to synchronize itself, plus a further minute to decode the data for the following minute. It is waiting, during that time, for the missing 59th second marker. Internally the function block is sampling the DCF-77 signal. In order to be able to sample the edges without error the function block should be called once in each PLC cycle. Satisfactory results can be obtained with a cycle time of <= 25 ms. If the DCF-77 signal is absent or faulty, the ERR output is set TRUE, and a corresponding error code is set at the ERRID output. The ERR and ERRID outputs are reset the next time a correct signal is received. Some receivers provide an inverted DCF-77 signal. In such cases the signal must first be inverted before being passed to the DCF_PULSE input. When operating without errors, the current time is updated at the CDT output every minute. When this is happening, the READY output is set to TRUE for one PLC cycle in the first second (second zero). At this time the DCF-77 time at the CDT output is valid, and can be evaluated by the PLC program. The READY output is only set if no error was detected in the data for the following minute. The transferred parity bits are used for error detection. When reception is poor 100% error-free identification cannot be guaranteed, in the event of two defective (inverse) bits the function block cannot detect an error and will set the READY output to TRUE. In order to obtain reliable time information additional safeguards have to be implemented, e.g. redundancy analysis of the time information in consecutive minutes.

A simple plausibility check of two consecutive telegrams is implemented in the DCF77_TIME function block. This functionality can be activated via a global Boolean variable for all instances of the DCF77_TIME block. When the plausibility check is activated the first synchronization is extended by a further minute to a maximum of 3 minutes.

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

Errors that occur during reception are registered by the function block. The ERRCNT output is an error counter. This counter indicates the number of errors that have occurred since the last correctly received signal. The counter is reset the next time a signal is correctly received.

Time code

During each minute, the numbers that encode the year, month, day, day of the week, hour and minute are transmitted in BCD format through pulse modulation of the second marks. The transmitted information always describes the subsequent minute. A second marker is transmitted each second. A second marker with a duration of 0.1 s represents a binary zero, while a duration of 0.2 s represents binary one. The information is extended with 3 check bits. No second marker is transmitted for the 59th second, and a receiver can use this "gap" to synchronize itself.

The length of the short and long pulse signals can be configured via a global variable. If the signal is poor the pulse widths are smaller. The receiver specification usually contains information about the minimum and maximum pulse for the two logic signals, with the higher value expected for higher field strengths and the lower value for low field strengths or in the event of interference. Problems may also occur near the sender (where the field strength is very large) if the pulse width of the logic zero becomes excessive. For this reason a fixed limit is set for differentiating between zero and one, depending on the receiver specification. **Check the specification of the receiver used and configure the impulse length accordingly.**

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

E.g.: the Atmel T4227 specification (time code receive) contains the following pulse length specification:
100 ms pulses (zero): Min: 70 ms, typical: 95 ms, max: **130 ms**
200 ms pulses (one): min **170 ms**, typical 195 ms, max 235 ms
For this IC a limit value of 150 ms would be ideal (130 + ((170 ms - 130 ms) / 2)).

Tip:

If the configured limit value for the impulse length is too small, short impulses are detected as long. Conversely, if the configured limit value is too small, long impulses are detected as short. If the checksum is correct, the receiver cannot detect these errors. In the first case the receiver may supply times that are in future range, in the second case the times may be in the past.

VAR_INPUT

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

DCF_PULSE: The DCF-77 signal.

RUN: A rising edge at this input initializes the function block and starts decoding the DCF-77 signal. If this input is reset, the decoding process is stopped.

VAR_OUTPUT

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

BUSY: This output is set when the function block is activated.

ERR: This output is set when an error occurs in the decoding.

ERRID: Supplies the error number when the ERR output is set.

ERRCNT: The number of errors that have occurred since the last correctly received signal.

READY: If this output is set, the data at the CDT output is valid.

CDT: The DCF-77 time in DATE_AND_TIME format.

Error Codes	Error description
0	No error
0x100	Timeout error. Possibly because no DCF-77 signal is detected.
0x200	Parity error. Incorrect bits were detected in the received data.
0x300	Faulty data received. Since the parity check can only detect one incorrect bit, the received data is checked again for validity (this error code will be generated, for instance, if month = 13).
0x400	The last decoding cycle was too long. This error can occur when reception is poor (not enough second markers were received).
0x500	The last decoding cycle was too short. This error can occur when reception is poor (additional edges were received).

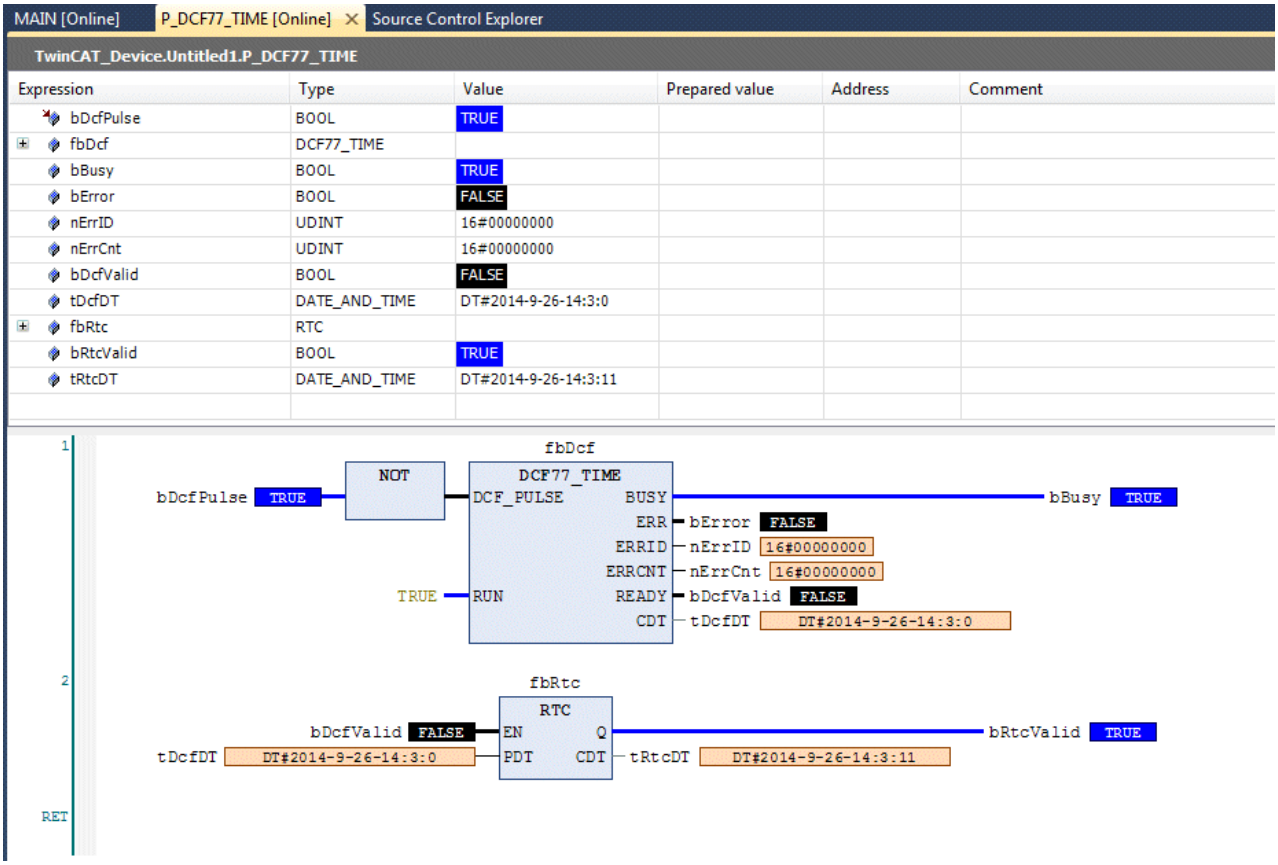
Example:

If data are received error-free, a TwinCAT software clock (RTC) is synchronized with the radio time in the sample application.

```
PROGRAM P_DCF77_TIME
VAR
    bDcfPulse    : BOOL;
    fbDcf        : DCF77_TIME;
    bBusy        : BOOL;
    bError       : BOOL;
    nErrID       : UDINT;
```

```
    nErrCnt      : UDINT;
    bDcfValid    : BOOL;
    tDcfDT       : DT;
    fbRtc        : RTC;
    bRtcValid    : BOOL;
    tRtcDT       : DT;
END_VAR
```

Online View:

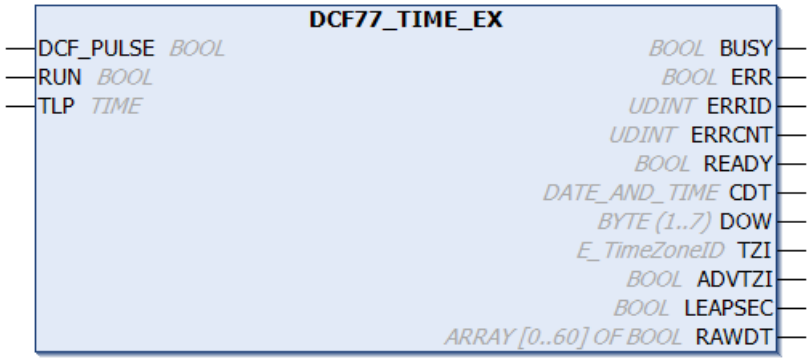


See also description of the DCF77_TIME_EX [▶ 35] function block.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.4 DCF77_TIME_EX



The "DCF77_TIME_EX" function block can be used to decode the DCF-77 radio clock signal. In contrast to the DCF77_TIME [▶ 32] function block, this block checks two consecutive telegrams for plausibility as standard.

A rising edge at the RUN input starts the decoding process, which continues as long as the RUN input remains set. In the worst case synchronization of the function block takes up to one minute and two further minutes for decoding data for the next minute. It is waiting, during that time, for the missing 59th second marker. Internally the function block is sampling the DCF-77 signal. In order to be able to sample the edges without error the function block should be called once in each PLC cycle. Satisfactory results can be obtained with a cycle time of <= 25 ms. If the DCF-77 signal is absent or faulty, the ERR output is set TRUE, and a corresponding error code is set at the ERRID output. The ERR and ERRID outputs are reset the next time a correct signal is received. Some receivers provide an inverted DCF-77 signal. In such cases the signal must first be inverted before being passed to the DCF_PULSE input. When operating without errors, the current time is updated at the CDT output every minute. When this is happening, the READY output is set to TRUE for one PLC cycle in the first second (second zero). At this time the DCF-77 time at the CDT output is valid, and can be evaluated by the PLC program. The READY output is only set if it was possible to receive the data for the coming minute without errors. The transferred parity bits are used for fault detection, and the last two telegrams are checked for plausibility. When reception is poor 100% error-free identification cannot be guaranteed, in the event of two defective (inverse) bits in two subsequent telegrams the function block cannot detect an error and will set the READY output to TRUE. Due to the plausibility check the probability of the appropriate bits becoming distorted, thereby preventing detection of such an error, is quite small.

Errors that occur during reception are registered by the function block. The ERRCNT output is an error counter. This counter indicates the number of errors that have occurred since the last correctly received signal. The counter is reset the next time a signal is correctly received.

Time code

During each minute, the numbers that encode the year, month, day, day of the week, hour and minute are transmitted in BCD format through pulse modulation of the second marks. The transmitted information always describes the subsequent minute. A second marker is transmitted each second. A second marker with a duration of 0.1 s represents a binary zero, while a duration of 0.2 s represents binary one. The information is extended with 3 check bits. No second marker is transmitted for the 59th second, and a receiver can use this "gap" to synchronize itself.

VAR_INPUT

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

DCF_PULSE: The DCF-77 signal.

RUN: A rising edge at this input initializes the function block and starts decoding the DCF-77 signal. If this input is reset, the decoding process is stopped.

TLP: Via this input a fixed limit is set for differentiating between zero and one, depending on the recipient specification. If the signal is poor the pulse widths are smaller. The receiver specifications usually contain information about the minimum and maximum pulse for the two logic signals, with the higher value expected for higher field strengths and the lower value for low field strengths or in the event of interference. Problems may also occur near the sender (where the field strength is very large) if the pulse width of the logic zero becomes excessive. **Check the specification of the receiver used and configure the impulse length accordingly.**

E.g.: the Atmel T4227 specification (time code receive) contains the following pulse length specification:
100 ms pulses (zero): Min: 70 ms, typical: 95 ms, max: **130 ms**
200 ms pulses (one): min **170 ms**, typical 195 ms, max 235 ms
For this IC a limit value of 150 ms would be ideal (130 + ((170 ms - 130 ms) / 2)).

If the configured limit value for the impulse length is too small, short impulses are detected as long. Conversely, if the configured limit value is too small, long impulses are detected as short. If the checksum is correct, the receiver cannot detect these errors. In the first case the receiver may supply times that are in future range, in the second case the times may be in the past.

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: This output is set when the function block is activated.

ERR: This output is set when an error occurs in the decoding.

ERRID: Supplies the error number when the ERR output is set.

ERRCNT: The number of errors that have occurred since the last correctly received signal.

READY: If this output is set, the data at the CDT output is valid.

CDT: The DCF-77 time in DATE_AND_TIME format.

DOW: day of the week according to ISO 8601: 1 = Monday... 7 = Sunday.

TZI: time zone information (summer/winter time).

ADVTZI: summer/winter time changeover notification, e.g. CET -> CEST or CEST -> CET. The changeover between CEST/CET occurs at the end of this hour (see telegram examples).

LEAPSEC: leap second notification. A leap second is added at the end of this hour (see telegram examples).

RAWDT: last decoded (raw) bit information. Please ensure that only the parity bits of the time information are checked. The parity bits of the weather data are not analyzed.

Error Codes	Error description
0	No error
0x100	Timeout error. Possibly because no DCF-77 signal is detected.
0x200	Parity error. Incorrect bits were detected in the received data.
0x300	Incorrect data was received. Since the parity check can only detect one incorrect bit, the received data is checked again for validity (this error code will be generated, for instance, if month = 13).
0x400	The last decoding cycle was too long. This error can occur when reception is poor (not enough second markers were received).
0x500	The last decoding cycle was too short. This error can occur when reception is poor (additional edges were received).

Telegram examples:

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

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

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

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

Leap second

'0 10110000100001 000111 10011010 0000000 100000 001 10000 100100001': Thursday, 01.01.09 00:59:00, TZI = eTimeZoneID_Standard, LEAPSEC = TRUE
'0 11010010111000 000111 00000000 1000001 100000 001 10000 1001000010': Thursday, 01.01.09 01:00:00, TZI = eTimeZoneID_Standard, LEAPSEC = TRUE
'0 01000110011101 000101 10000001 1000001 100000 001 10000 100100001': Thursday, 01.01.09 01:01:00, TZI = eTimeZoneID_Standard, LEAPSEC = FALSE

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

Example:

If data are received error-free, a TwinCAT software clock (RTC) is synchronized with the radio time in the sample application.

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

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

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

Online View:

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

TwinCAT_Device.Untitled1.MAIN

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

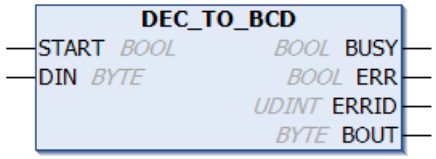
```
fbDcf( DCF_PULSE:= NOT bDcfPulse, RUN:= TRUE, TLP:= T#140MS,
    BUSY=>bBusy, ERR=>bError, ERRID=>nErrID, ERRCNT=>nErrCnt,
    READY=>bDcfValid, CDT=>tDcfDt, DOW=>nDow, TZI=>eTzi,
    ADVTZI=>bAdvTzi, LEAPSEC=>bLeapSec, RAWDT=>arRawDt );

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

See also description of the DCF77_TIME [▶ 32] function block.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.5 DEC_TO_BCD



The "DEC_TO_BCD" function block allows decimal numbers to be converted to BCD format. The number to be converted is checked for the reliability of the values.

VAR_INPUT

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

START: The function block is activated by a positive edge at this input.

DIN: The decimal number requiring conversion.

VAR_OUTPUT

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

BUSY: This input is set at the start of the conversion procedure, and remains set until the conversion has been completed. Once the BUSY output has been reset, the BCD value is available at the BOUT output.

ERR: This variable is set to TRUE if an error occurs.

ERRID: Error code.

BOUT: The converted variable in BCD format is available at this output if the process is successful.

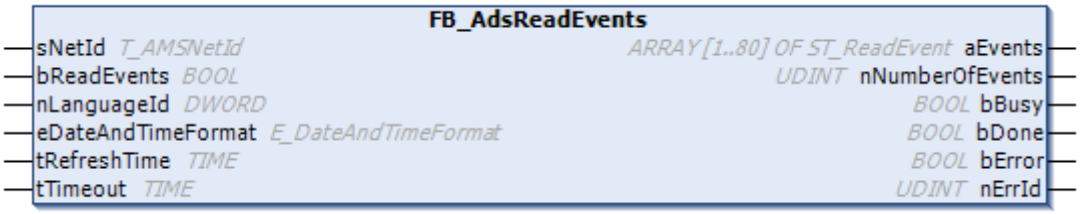
Error Codes:

Error Code	Error Description
0	no error
0x00FF	the variable that is to be converted has a forbidden decimal value

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.6 FB_AdsReadEvents



The function block queries the active messages of the EventLogger via ADS and makes them available in the form of an array `aEvents`. To display the messages in the visualization element Event table, the array `aEvents` has to be entered in its property Message data array.

Messages with a text length less than or equal to 255 characters can be output in full at the output. Messages with a text length greater than 255 characters and less than or equal to 1023 characters are output with truncated text. Messages with a text length greater than 1023 characters cannot be output and the function block returns an error.

VAR_INPUT

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

sNetId: AmsNetID of the device, from which the messages of the EventLogger are to be queried. If the messages are to be read locally, an empty string can be specified.

bReadEvents: The input can be used to enable reading of the messages. When the enable is reset, the error outputs (bError and nErrId) are also reset.

nLanguageId: (Language Id) Defines the language in which message texts are displayed.

eDateAndTimeFormat: Defines the timestamp format. The available options are:

- de_De – German spelling: dd.MM.yyyy hh:mm:ss (24 h)
- en_GB – British spelling: dd/MM/yyyy hh:mm:ss (12 h)
- en_US – American spelling: MM/dd/yyyy hh:mm:ss (12 h)

tRefreshTime: Defines the time interval, after which the message query is repeated.

tTimeout: Defines the time interval, after which a timeout error is triggered.

VAR_OUTPUT

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

aEvents: The function block uses this array to make the read messages available. The array can store a maximum of 80 messages. (see ST_ReadEvent [▶ 312])

nNumberOfEvents: Indicates how many messages are currently stored in the array `aEvents`.

bBusy: Indicates whether the function block is currently busy.

bDone: TRUE, if the function block is not busy at present, but has carried out at least one operation.

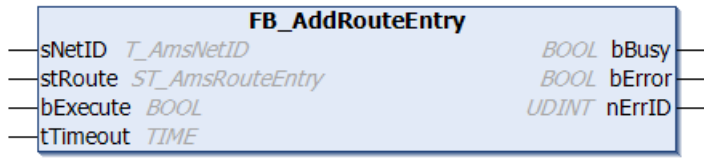
bError: Indicates whether an error has occurred.

nErrorId: Indicates the error ID.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.7 FB_AddRouteEntry



The function block can be used to add a new AMS router connection (remote route) to a TwinCAT system.

> **AMS Router connection list**
>
> Both communication partners of an AMS Router connection have an AMS Router connection list. These lists contain AMS router connections. The functionality of an AMS router connection is given if both communication partners have entered each other in their connection list.
> When using the function block, the list of only one communication partner is extended.

VAR_INPUT

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

sNetID: Here you can specify a string with the network address of the TwinCAT computer on which a new connection is to be added to the AMS router connection list (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

stRoute: Structure element with parameters for the new connection (type: ST_AmsRouteEntry [▶ 305]).

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

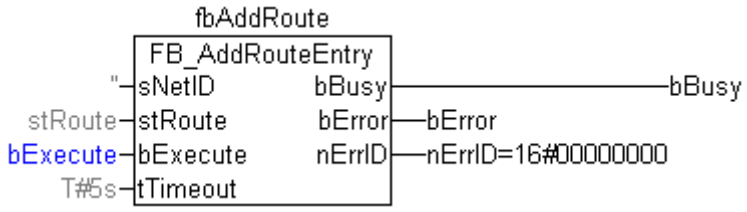
nErrId: Supplies the ADS error number [▶ 353] when the *bError* output is set.

Example:

On the local TwinCAT system a new AMS router connection is to be added with the connection name: "TEST", TwinCAT network address: "172.16.6.111.1.1", IP address: "172.16.6.111" and transport route: "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
```

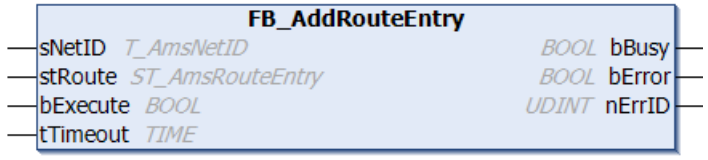
The required connection parameters are already initialized in the declaration part. The new connection is added if a rising edge is detected at the bExecute variable.



Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.8 FB_AddRouteEntryEx



The function block can be used to add a new AMS router connection (remote route) to a TwinCAT system.

The function block supports the AmsNAT functionality with the virtual AmsNetId. This enables creation of routes to two or more controllers that have the same AmsNetId.

AMS router connection list

Both communication partners of an AMS router connection have an AMS router connection list. These lists contain AMS router connections. An AMS router connection is functional when both communication partners have entered each other in their respective connection list.
When the function block is used, the list of only one communication partner is extended.

VAR_INPUT

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

sNetID: Here you can specify a string with the network address of the TwinCAT computer on which a new connection is to be added to the AMS router connection list (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

stRoute: Structural element with parameters for the new connection (type: ST_AmsRouteEntryEx [▶ 305]).

bExecute: The function block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrId: Supplies the ADS error number [▶ 353] when the *bError* output is set.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (system) >= v3.3.41.0

3.9 FB_AmsLogger

The "TwinCAT AMS Logger" is a component of the "TwinCAT ADS Monitor" (..\TwinCAT\AdsMonitor\Logger**TcAmsLog.exe**). The logger records the AMS/ADS commands on the data storage device. The recording can subsequently be displayed and analyzed with the "TwinCAT AMS ADS Viewer", e.g. for troubleshooting.
The FB_AmsLogger function block can be used to start or stop the recording from a PLC program. The FB_AmsLogger function block can only communicate with an existing/running instance of TcAmsLog.exe. In other words, TcAmsLog.exe must already have been started, e.g. manually via the Start menu or via the NT_StartProcess [▶ 126] block.

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: Here you can specify the AmsNetID of the TwinCAT computer on which the status of the "TwinCAT AMS Logger" (type: T_AmsNetID). An empty string can be specified for the logger on the local computer.

eMode: The new status to which the "TwinCAT AMS Logger" is to be set (type: E_AmsLoggerMode [▶ 294], start/stop recording).

sCfgFilePath: (Optional) path for a "TwinCAT AMS Logger" configuration file (type: T_MaxString). Currently not yet implemented and reserved for future applications (Please enter an empty string).

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: If an error should occur during the transfer of the command, then this output is set once the bBusy output was reset.

nErrId: Supplies the ADS error number [▶ 353] when the bError output is set.

Example:

When the PLC program starts an instance of TcAmsLog.exe is started on a local system. When the bRecord variable is set to TRUE, the recording of the AMS/ADS command is started. It is stopped when the variable is reset to FALSE.

Declaration part:

```
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
```

Implementation:

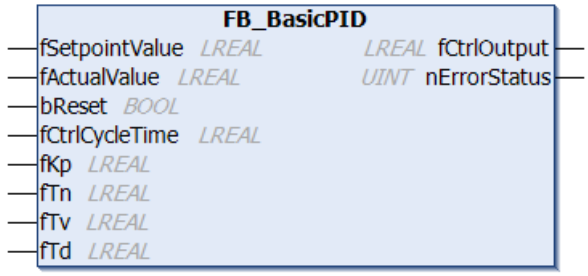
```
CASE state OF
0:(* Start instance of TcAmsLogger.exe *)
    fbStartProcess( START := FALSE );
    fbStartProcess( START:= TRUE );
    state := 1;

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
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.10 FB_BasicPID

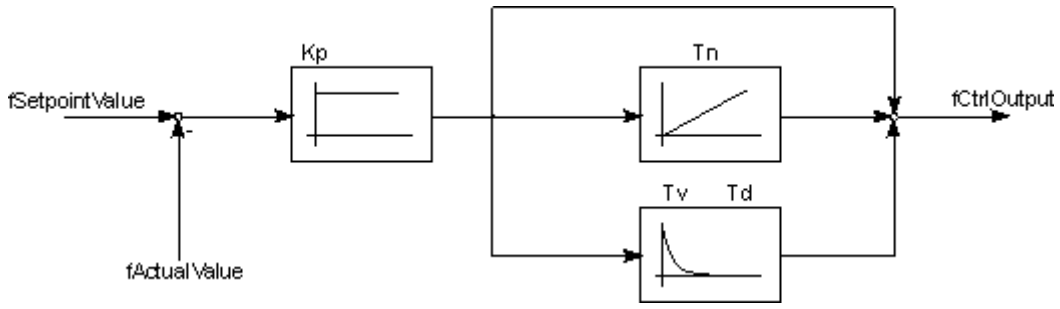


The function block is a simple discretized PID element.

Transfer function:

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

Action diagram:



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 : Set value of the controlled variable.

fActualValue : Actual value of the controlled variable.

bReset: TRUE at this input resets the internal state variables and the controller output.

fCtrlCycleTime: Cycle time with which the function block is called and with which the control loop is processed [s].

Here you **must** specify the cycle time of the PLC task, if the block is to be called in each PLC cycle,

otherwise the required multiple of the PLC task cycle time.

fKp : Controller amplification / controller coefficient

fTn : Integral action time [s]

fTv : Rate time [s]

fTd : Damping time [s]

VAR_OUTPUT

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

fCtrlOutput : Output of the PID-element.

nErrorStatus: Indicates the error number in the event of an error (nErrorStatus <> 0).

Error Codes:

Value	Constant	Error description
0	**nERR_NOERROR**	No error
1	**nERR_INVALIDPARAM**	Invalid parameter
2	**nERR_INVALIDCYCLETIME**	Invalid cycle time.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.11 FB_CalcHashValue

This function block calculates a hash value.

For this purpose the methods start(), update() and finish() are used.

The methods decouple the appending of input data from the actual calculation of the hash value and also allow input data to be appended piece by piece in multiple steps. If it is not necessary to add input data multiple times, it is recommended to use the function F_GenerateHashValue() [▶ 263] instead of the function block.

Start() method

This method initializes the hash calculation with the specified hash mode.

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

hashMode: a hash mode, such as SHA 512, is specified here. See E_HashMode [▶ 297].

Method update()

This method can be called once or multiple times. Input data for the hash calculation is added with each call.

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

pData: the address of the input data is specified here.

nData: the size of the input data in bytes is specified here.

Method finish()

This method performs the hash calculation and outputs the calculated hash value.

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

pHash: here the address of the buffer is specified where the hash value is to be stored.

nHash: the size of the buffer for the hash value is specified here. The size depends on the hash mode, see also E_HashMode [▶ 297].

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024.29	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.51.0

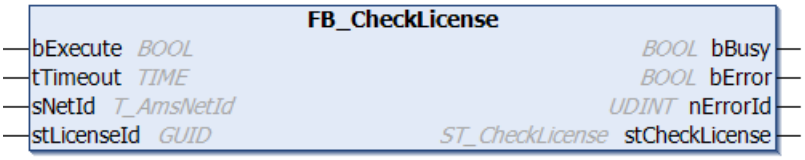
Also see about this

- ST_FindFileEntry [▶ 310]
- ADS Return Codes [▶ 353]
- F_GenerateHashValue [▶ 263]

3.12 FB_CheckLicense

If you use OEM licenses make sure you encrypt your boot project!

Remember that the license ID queried via FB_CheckLicense in the binary code can easily be found and (with a little effort) manipulated with a hex editor. Therefore, be sure to encrypt your boot project (safest), or at least disguise the queried license ID in the source code as best as possible.



The function block FB_CheckLicense determines the TwinCAT 3 license status for a given license ID.

VAR_INPUT

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

bExecute: The function block is activated by a positive edge at this input.

tTimeout: Timeout time that must not be exceeded when the command is executed.

sNetId: AmsNetId (AMS network identifier) of the TwinCAT computer whose license status is to be read (type: T_AmsNetId). If it is to be run on the local computer, an empty string can be entered.

stLicenseId: License ID (type: GUID)

VAR_OUTPUT

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

bBusy: TRUE, as long as the function block is active.

bError: TRUE if an error occurs during command execution.

nErrorId: Supplies the ADS error number when the bError output is set.

stCheckLicense: Structure with license data (type: ST_CheckLicense)

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4022	PC or CX (x64, x86)	Tc2_Utilties >= 3.3.24.0

3.13 FB_CSVMemBufferReader

This function block can be used to decompose/interpret data sets stored in an external buffer into individual data fields. The buffer data could first be read from a file with the aid of the function blocks for file access, for example. The function block reads the first or the next data field and returns its value either as a string at the *getValue* output or as an address/byte value at the *pValue*/*cbValue* output.

The data in the buffer must have a certain format to ensure that the function block can interpreted them correctly. CRLF (CR=Carriage Return, LF=Line Feed) must be used as a data set separator. The last data set must end with a CRLF. Individual data fields must be separated with the data field separator. The default data field separator is a semicolon. The separator can be configured from semicolon to comma via the global PLC variable **DEFAULT_CSV_FIELD_SEP**.

VAR_INPUT

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

eCmd: Control parameter for the buffer component (type: E_EnumCmdType [▶ 297]). eEnumCmd_First reads the first data field, eEnumCmd_Next reads the next data field. No other parameter values are used.

pBuffer: Address (pointer) for the source buffer variable. The address can be determined with the ADR operator. This buffer contains the data set/data field data to be read.

cbBuffer: The byte size of the data to be interpreted in the source buffer (data set/data field data). The buffer size may be much larger than the amount of data to be interpreted. Please enter the actual length of the data to be interpreted.

VAR_OUTPUT

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

bOk: TRUE = Success, FALSE = Faulty data/faulty input parameters, or the end of the data was reached and no further data field could be read.

getValue: The last read data field as string (type: T_MaxString). For data fields without control characters and binary data, this output returns the complete data field as a null-terminated string. However, data fields with control characters or binary data can return an incomplete string at this output. In this case the outputs *pValue/cbValue* are used to access the last read data field.

pValue: Address (pointer) to the first data byte in the data field. Please note that empty void data fields are not terminated with null (as is usual for a PLC string) and therefore have no data. In this case the address is null.

cbValue: Data field length in bytes. Please note that empty void data fields are not terminated with null (as is usual for a PLC string) and therefore have no data. The length is also null in this case. The maximum size can be specified with the global parameter `cMaxCSVFieldValueSize`.

bCRLF: This output is set if the end of the data set was reached during the last read command. The last read data field belongs to the previous data set. The next data field belongs to a new data set.

cbRead: Number of successfully read/interpreted data bytes. This number may be greater than the data field length at the *cbValue* output. The length at the *cbRead* output includes the interpreted data field/data set separators.

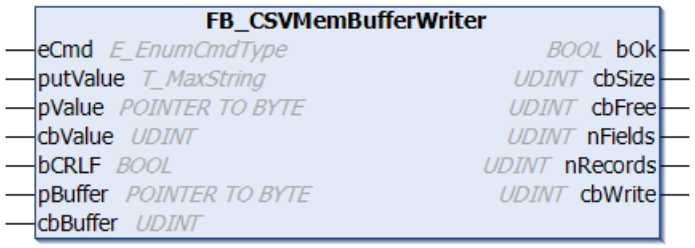
Example:

See: Example: Writing/reading of a CSV file. [▶ 344]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.14 FB_CSVMemBufferWriter



This function block can be used to generate data sets in an external buffer in CSV format from individual data fields. The content of the buffer can then be written into a file, e.g. with the aid of the function blocks for file access. The new data field can be transferred to the block either via the *putValue* variable (string) or via the optional *pValue* and *cbValue* variables, depending on whether data fields without control characters (string), data fields with control character or binary data are to be written into the data set. The function block can generate several data sets in the buffer until the maximum available buffer size is reached. The end of the data set (last data field in the current data set) is automatically appended to the data field if the bCRLF variable was set to TRUE during writing of the data field. The block automatically adds the data field separators. The default data field separator is a semicolon. The separator can be configured from semicolon to comma via the global PLC variable **DEFAULT_CSV_FIELD_SEP**.

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: Control parameter for the buffer block (type: E_EnumCmdType [▶ 297]). eEnumCmd_First adds the first data field to the buffer, eEnumCmd_Next adds the next data field. No other parameter values are used.

putValue: A new data field as string (type: T_MaxString). This input must be an empty string if the optional parameters *pValue* and *cbValue* are used instead of this input.

pValue: Optional: Address of an external byte buffer containing the new data field. Together with the cbValue parameter this input can then be used to write a data field with control characters or binary data into the data set, for example. Control characters or binary data in the data field could truncate the *putValue* string at an undesired position and are therefore transferred as a byte buffer. This input must be null if it is not used.

cbValue: Optional: Length of the data field data in the external byte buffer. This input must be null if it is not used. The maximum size can be specified with the global parameter cMaxCSVFieldValueSize.

bCRLF: If this input is set, then the new data field is terminated with a CRLF data set separator. Subsequent data fields belong to the new data set.

pBuffer: Address (pointer) to the destination buffer variable. The address can be determined with the ADR operator. In this buffer the function block generates the data sets in CSV format.

cbBuffer: Maximum available size (in bytes) of the destination buffer variable. The size can be determined with the SIZEOF operator.

VAR_OUTPUT

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

bOk: TRUE = success, FALSE = buffer overflow or faulty input parameters.

cbSize: Current buffer fill status (number of data bytes created in the buffer).

cbFree: Number of free data bytes in the buffer.

nFields: Number of written data fields.

nRecords: Number of written data sets.

cbWrite: Number of last written data bytes (length of the last data field + any data set or data field separators).

Example:

See: Example: Writing/reading of a CSV file. [▶ 344]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.15 FB_EnumFindFileEntry

This function block searches a directory for a file or a subdirectory whose name is similar to the specified name. Any entries found can be read individually. See also description of the FB_EnumFindFileList function block. The input parameter *eCmd* is used for navigating through the list of entries. The *eCmd* input determines whether the first or the next input is read, for example.

Important notes:

A new search may be started only if the previous search has been fully completed. The function block may need to be activated several times (by a rising edge at the bExecute input) for a complete search. The search is only fully complete if bEOE =TRUE was reached or if the search was terminated prematurely with ECMD = eEnumCmd_Abort.

For the TwinCAT system, the search may not yet be completed if the PLC application has already found the file or directory that was sought.

If not all entries are to be read (i.e. *bEOE=TRUE* is not reached), the function block subsequently has to be called with the input parameter *eCmd = eEnumCmd_Abort*. This is necessary in order to complete the search and release all internal resources (file handles). If *bEOE=TRUE* was reached or if an error occurs, *eEnumCmd_Abort* is automatically executed internally.

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: String containing the network address of the TwinCAT computer on which a directory search is to be executed (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

sPathName: A valid directory name or directory with file name as string (type: T_MaxString). The string can contain (* and ?) as wildcards. If the path ends with a wildcard, dot or the directory name, the user must have access rights to this path and its subdirectories.

eCmd: Command parameter for the enumeration block (type: E_EnumCmdType [▶ 297]).

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrId: Supplies the ADS error number [▶ 353] when the *bError* output is set.

bEOE: End of enumeration was reached. During the first attempt to read a non-existing entry this output is set to TRUE. This means that read entries are valid as long as *bEOE* = FALSE and *bError* = FALSE.

stFindFile: If successful this structure variable returns information about the file that was found (type: ST_FindFileEntry [▶ 310]).

Example:

See: Example: File search (FB_EnumFindFileEntry, FB_EnumFindFileList). [▶ 328]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Utilities (System)

3.16 FB_EnumFindFileList

This function block searches a directory for a file or a subdirectory whose name is similar to the specified name. Any entries found can be read individually. See also description of the FB_EnumFindFileEntry function block. The input parameter *eCmd* is used for navigating through the list of entries. The *eCmd* input determines whether the first or the next input is read, for example.

Important notice:

A new search may be started only if the previous search has been fully completed. The function block may need to be activated several times (by a rising edge at the bExecute input) for a complete search. The search is only fully complete if bEOE =TRUE was reached or if the search was terminated prematurely with ECMD = eEnumCmd_Abort.

For the TwinCAT system, the search may not yet be completed if the PLC application has already found the file or directory that was sought.

If not all entries are to be read (i.e. *bEOE=TRUE* is not reached), the function block subsequently has to be called with the input parameter *eCmd = eEnumCmd_Abort*. This is necessary in order to complete the search and release all internal resources (file handles). If *bEOE=TRUE* was reached or if an error occurs, *eEnumCmd_Abort* is automatically executed internally.

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: String containing the network address of the TwinCAT computer on which a directory search is to be executed (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

sPathName: A valid directory name or directory with file name as string (type: T_MaxString). The string can contain (* and ?) as wildcards. If the path ends with a wildcard, dot or the directory name, the user must have access rights to this path and its subdirectories.

eCmd: Control command for the enumeration block (type: E_EnumCmdType [▶ 297]).

pFindList: Address (pointer variable) of an array variable of type: ST_FindFileEntry [▶ 310].

cbFindList: Byte size of array variable of type: ST_FindFileEntry [▶ 310].

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrId: Supplies the ADS error number [▶ 353] when the *bError* output is set.

bEOE: End of enumeration was reached. During the first attempt to read a non-existing entry this output is set to TRUE. This means that read entries are valid as long as *bEOE* = FALSE and *bError* = FALSE.

nFindFiles: Number of valid entries in the buffer.

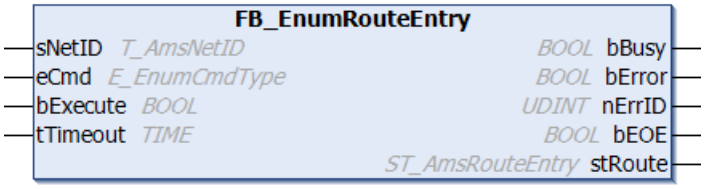
Example:

See: Example: File search (FB_EnumFindFileEntry, FB_EnumFindFileList). [▶ 328]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Utilities (System)

3.17 FB_EnumRouteEntry



This function block is used to transfer information to other TwinCAT systems via the AMS router connections (remote routes). If several connections are used the function block has to be called up repeatedly. Only one entry can be handled for each call. The input parameter *eCmd* is used for navigating through the list of entries. The *eCmd* input determines whether the first or the next input is read.

VAR_INPUT

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

sNetID: Here you can enter a string with the network address of the TwinCAT computer whose AMS router connections are to be read (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

eCmd: Control command for the enumeration block (type: E_EnumCmdType [▶ 297]).

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrId: Supplies the ADS error number [▶ 353] when the *bError* output is set.

bEOE: End of enumeration was reached. During the first attempt to read a non-existing entry this output is set to TRUE. This means that read entries are valid as long as *bEOE* = FALSE and *bError* = FALSE.

stRoute: Structural element containing the last read connection parameters (type: ST_AmsRouteEntry [▶ 305]).

Example:

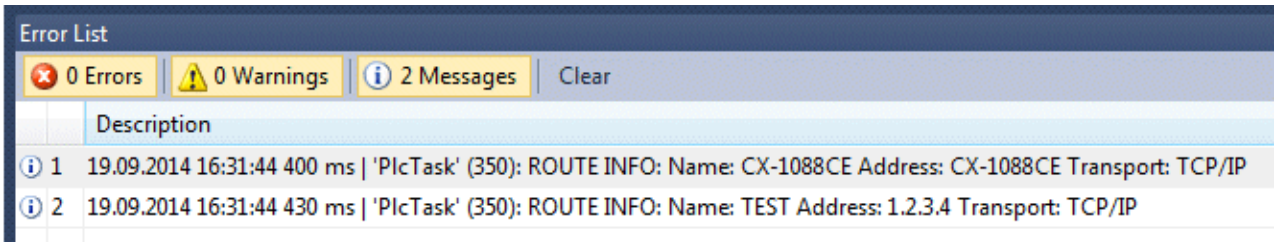
The configured AMS router connections are to be read on the local TwinCAT system and written into the TwinCAT System Manager logger output as messages.

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

A rising edge at the bEnum variable triggers reading of the connection information.

```
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, ROUTETRANSPORT_TO_STRING( 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
```

Log messages in the TwinCAT System Manager logger output:



Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.18 FB_EnumStringNumbers

This function block can be used to search a string in a REPEAT or WHILE loop for numbers. The string may contain several numbers. Any numbers that are found are output as sub-strings at the block output. The function searches from the current position for the first character that can be interpreted as a numeral. The search is aborted if a character is found that cannot be interpreted as a number. The *eCmd* parameter determines whether the search is for the first number or the next number. The *eType* parameter determines the format of the numbers in the search string.

VAR_INPUT

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

sSearch: Search string to be searched for numbers (type: T_MaxString).

eCmd: Control command for the enumeration block (type: E_EnumCmdType [▶ 297]).

eType: Number format of the searched number (type: E_NumGroupTypes [▶ 299]). This parameter determines which characters are to be ignored and which are to be interpreted as numerals:

Value	Meaning
eNumGroup_Float	Numbers '0' to '9', '+', '-' (signs) and 'e' or 'E' (exponent character) are interpreted as valid numerals.
eNumGroup_Unsigned	Numbers '0' to '9' are interpreted as valid numerals. '+', '-', 'e', 'E' characters are ignored.
eNumGroup_Signed	Numbers '0' to '9', '+', '-' (signs) are interpreted as valid numerals. 'e', 'E' characters are ignored.

If the string contains numbers in the exponential notation, eType = *eNumGroup_Float* must be set (default).

VAR_OUTPUT

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

sNumber: The last found number as string (type: T_MaxString).

nPos: This variable always returns the position after the last found and correctly formatted numeral, i.e. at this position the block will start searching for new numerals when it is called next time. *nPos* is zero when the final zero of the *sSearch* string has been reached. The first character in the string has position number = 1 (non-zero based position).

bEOS: This variable is FALSE if a new number was found and the end of the string has not yet been reached. In this case *sNumber* returns a valid number as a string. This variable is TRUE if no further number was found. In this case any further search must be aborted (*sNumber* returns no valid value).

Example:

In the following example the *sNumber* variable is searched for valid numbers. Any sub-strings that are found are stored in the array variable *arrNums*.

```
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_U
nsigned *) );
    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
```

Found strings:

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

eType parameter *eNumGroup_Signed* returns the following results:

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

eType parameter *eNumGroup_Unsigned* returns the following results:

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

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.19 FB_FileProperties

This function block outputs the file properties of a selected file.

VAR_INPUT

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

sNetID: Here you can enter a string with the network address of the TwinCAT computer whose directory is to be searched (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

sPathName: Here the complete file path including file name and file extension is specified as a string (type: T_MaxString).

bExecute: The function block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

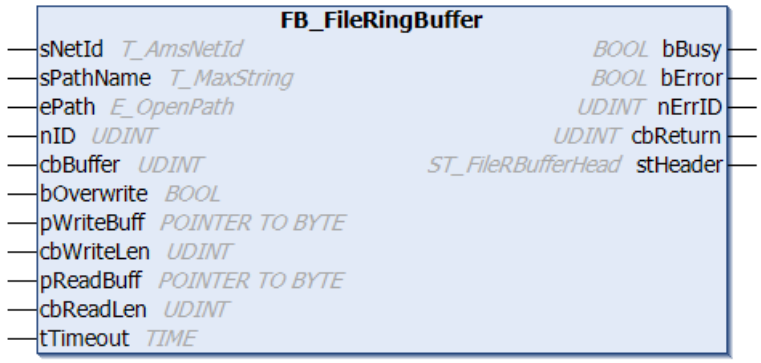
nErrId: Supplies the ADS error number [▶ 353] when the *bError* output is set.

stProperties: If the search was successful, this structure variable returns information on the file that was found (type: ST_FindFileEntry [▶ 310]).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Utilities (System)

3.20 FB_FileRingBuffer



The FB_FileRingBuffer function block allows data records of varying lengths to be written into a ring buffer file, or for data records that have previously been written there to be removed from the ring buffer file. The written data sets are read in the same order as they were previously written to the ring buffer file, based on the FIFO principle, i.e. the oldest entries are read first. Opening, closing, writing and reading the data records is controlled by action calls. The function block features the following tasks:

- **A_Open** (opens an existing ring buffer file for appending or generating new data sets. An error is not returned if the file is already open.)
- **A_Close** (Closes an open ring buffer file. An error is not returned if the file is already closed.)
- **A_Create** (Opens a new ring buffer file. If the file already exists, it is overwritten. An error is not returned if the file is already open.)
- **A_AddTail** (Write a new data record into the ring buffer file.)
- **A_GetHead** (Reads the oldest data record from the ring buffer file, but does not remove it – the data pointer is not moved to the next data record.)
- **A_RemoveHead** (Reads and removes the oldest data record from the ring buffer file – the data pointer is moved on to the next data record.)
- **A_Reset** (Deletes all the data records from an open ring buffer file. Only the data pointer and the number of data records are reset; the existing physical file size is not changed, although the oldest data records will be overwritten by new ones.)

When a ring buffer file that already exists is opened, the file header is read first. In a ring buffer file that has previously been closed without error, bit 0 in the header status (Header.status.Bit 0) must be zero. If not, it is assumed that the file was not properly closed beforehand, and the corrupt file is replaced by a new, empty file, while Header.status.Bit 1 is set to 1 (file corrupted). When the file is closed, Header.status.Bit 0 is set to 0, and the complete file header is updated in the file.

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: A string containing the network address of the TwinCAT computer where the buffer file is to be written or read can be given here (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

sPathName: Contains the path and file name for the buffer file to be opened (type: T_MaxString). **Notice**: The path can only point to the local computer's file system! This means that network paths cannot be used here!

ePath: This input can be used to select a TwinCAT system path on the target device for opening the file (type: E_OpenPath).

nID: User-defined 32-bit value. When a new file is opened, this value is saved in the file and can be used, for instance, for checking the version of the buffer file.

cbBuffer: The maximum size, in bytes, of the buffer file that is to be open. This parameter is saved in the file header when the file is created, and is checked when the same file is opened again. You can only reopen files that have been created using the same maximum buffer size. You cannot, in other words, create a file with a smaller buffer size, fill it with data records, and then open it again with a larger buffer size. If the check of the maximum buffer size fails, a new file with the new buffer size is automatically created and opened. Bit 1 (file corrupted) is also set in the file header status.

bOverwrite: Write behavior when the maximum file buffer size is reached. If TRUE is asserted at this input, the oldest entries are overwritten if the maximum file buffer size has already been reached. (Entries are deleted until there is enough free buffer size to save the new entry.) If FALSE is present at this input, a buffer overflow when the maximum file buffer size is reached is reported as an error.

pWriteBuff: The address of the PLC variable or of a buffer variable that contains the value data that is to be written. The address can be determined with the ADR operator. The programmer is himself responsible for dimensioning the buffer variable in such a way that *cbWriteLen* data bytes can be taken from it.

cbWriteLen: The number of value data bytes that are to be written. (In the case of string variables this includes the final null).

pReadBuff: The address of the PLC variables or of a buffer variable into which the value data that has been read is to be copied. The address can be determined with the ADR operator. The programmer is himself responsible for dimensioning the buffer variable in such a way that it can accept cbReadLen data bytes. The size of the buffer variables in bytes must be greater than or equal to the size of the data record that is to be read.

cbReadLen: The number of value data bytes to be read. If the buffer size is too small, data is not copied. The function block reports a buffer underflow error, and the buffer size required for the next data record that is to be read is returned at the *cbReturn* output.

tTimeOut: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrId: Supplies the ADS error number [▶ 353] or the command-specific error code (table) when the *bError* output is set.

cbReturn: The number of value data bytes successfully read. If a read buffer underflow error has occurred, this output supplies the necessary read buffer size in bytes.

stHeader: Ring buffer file header/status. (Type: ST_FileRBufferHead [▶ 309])

Command-specific error codes	Error description
0x8000	Writing: File buffer is empty. Reading: File buffer overflow.
0x8001	PLC application: read buffer underflow (pReadBuff, cbReadLen) has been dimensioned too small.
0x8002	Ring buffer file is closed, and must be opened first.
0x8003	Input parameter has incorrect value.

It is not essential for the PLC application to know the binary structure of the file. The following illustration, however, shows the general structure of the ring buffer file used:

Header (48 bytes) see description of ST_FileRBufferHead	Length of data set 1 (4 bytes)	Data set 1	Length of data set 2 (4 bytes)	Data set 2	Length of data set 3 (4 bytes)	Data set 3	Length of data set n (4 bytes)	Data set n

An empty ring buffer file only contains the header. The buffer itself follows the header. The variables Header.ptrFirst and Header.ptrLast point to the position immediately behind the header. Writing causes the ptrLast data pointer to be moved onwards. The ptrFirst data pointer follows the ptrLast data pointer during reading. When the maximum buffer size is reached the pointers are returned to the start of the buffer.

Example:

See: Example: File ring FiFo (FB_FileRingBuffer). [▶ 330]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

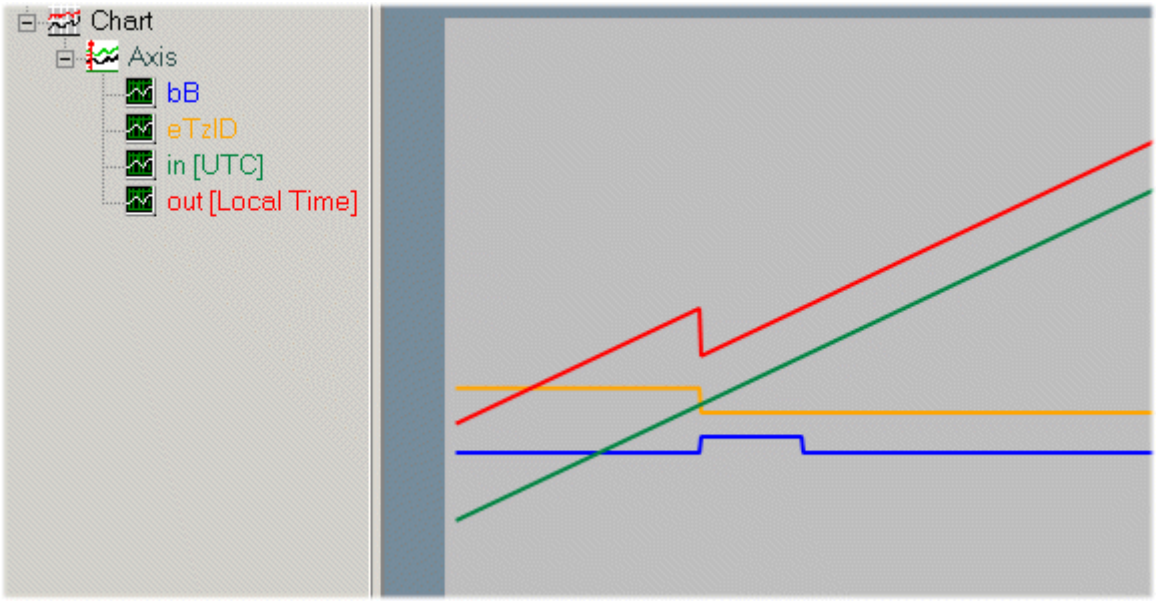
3.21 FB_FileTime64ToTzSpecificLocalTime

The function block converts the UTC time (file time format) to local time (file time format), taking into account the specified time zone information. The function block: FB_SystemTimeToTzSpecificLocalTime [▶ 111] has a similar function, the difference being that it converts to a different time format (structured system time format).

The function block is only suitable for conversion of **continuous** UTC timestamp information. The function block uses the time zone information to calculate the required time steps (summer/winter time changeover) in local time. Time steps in UTC input time are not permitted and lead to incorrect conversion. The reason: the function block stores the last converted time internally, so that it can detect the B times (see below) from the UTC input time and the stored value when the local time is changed.

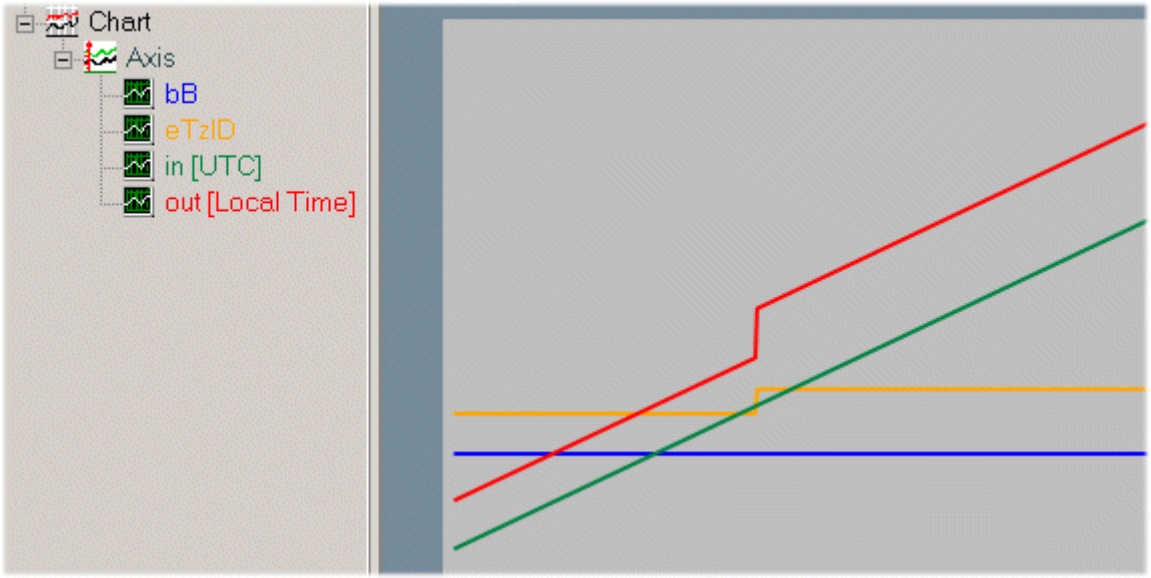
The function block is associated with an action: A_Reset(). If this action is called the function block outputs and the locally stored (last converted) time are reset to zero.

1. Graphic representation of the changeover from summer time to winter time (tzInfo = WEST_EUROPE_TZI):



The UTC input time (green) is continuous. The local time (red) jumps back. The local time: **02h:59m:59s:999ms..** is directly followed by: **02h:00m:00s:000ms..** The times between 2h and 3h occur twice. The duplicate time before the changeover is referred to as **02:05:00 CEST A**, for example, the time after the changeover as **02:05:00 CET B.** The output variable *bB* indicates whether it is the first or the second *pass*. During the second *pass* the *bB* output variable (blue) is set to TRUE. The *bB* output variable is automatically reset once the duplicate time has passed. The time zone ID (orange) changes from *eTimeZoneID_Daylight* (summer time) to *eTimeZoneID_Standard* (winter time).

2. Graphic representation of the change-over from winter time to summer time (tzInfo = WEST_EUROPE_TZI):



The UTC input time (green) is continuous. The local time (green) jumps forward. The local time: **2h:59m:59s:999ms..** is directly followed by: **3h:00m:00s:000ms..** The time zone ID (orange) changes from *eTimeZoneID_Standard* (winter time) to *eTimeZoneID_Daylight* (summer time).

VAR_INPUT

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

in: UTC time (file time format) that is to be converted (type: T_FILETIME64 [▶ 317]).

tzInfo: Structure variable with the current time zone information of the operating system (type: ST_TimeZoneInformation [▶ 315]).

VAR_OUTPUT

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

out: Converted local time (file time format, type: T_FILETIME64 [▶ 317]).

eTzID: Additional summer/winter time information. (Type: E_TimeZoneID [▶ 302])

bB: TRUE => B-time (e.g.:**02:05:00 CET B**), FALSE => remaining time (e.g.: **02:05:00 CEST A**). This output is set if the local time jumps back and is reset once the duplicate local time has passed.

Sample:

The UTC time: DT#2011-09-02-09:01:31 is converted to local time. The result is: DT#2011-09-02-11:01:31.

```
PROGRAM MAIN
VAR
    in        : DT := DT#2011-09-02-09:01:31;(* UTC time *)
    out       : DT;(* Local time *)
    fbToLocal : FB_FileTime64ToTzSpecificLocalTime;
END_VAR
```

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

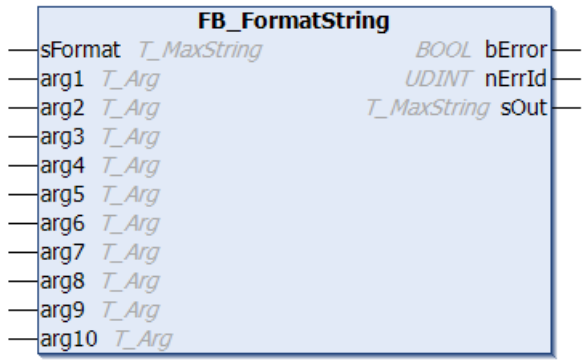
Further time and time zone functions and function blocks:

- FB_TzSpecificLocalTimeToSystemTime [▶ 115]
- FB_TzSpecificLocalTimeToFileTime64 [▶ 112]
- FB_SystemTimeToTzSpecificLocalTime [▶ 111]
- FB_GetTimeZoneInformation [▶ 77]
- FB_SetTimeZoneInformation [▶ 108]
- NT_SetLocalTime [▶ 122]
- NT_GetTime [▶ 120]
- NT_SetTimeToRTCTime [▶ 123]
- F_TranslateFileTime64Bias [▶ 150]
- FB_LocalSystemTime [▶ 89]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.44.0

3.22 FB_FormatString



This function block can be used for converting up to 10 arguments (similar to fprintf) into a string and formatting them according to the format specification [▶ 349]. Formatting occurs in the same PLC cycle, i.e. the output string is available immediately after the FB was called.

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: Format specification as string (type: T_MaxString)(e.g. '%+20.5f' or 'Measure X: %+..10d, Y: %+..10d').

arg1 to arg10: Arguments to be formatted (type: T_Arg [▶ 316]). The following auxiliary functions can be used for converting different types of PLC variables into the required data type T_Arg [▶ 316]: F_BYTE [▶ 228], F_WORD [▶ 235], F_DWORD [▶ 229], F_LWORD [▶ 231], F_SINT [▶ 232], F_INT [▶ 230], F_DINT [▶ 229], F_LINT [▶ 231], F_USINT [▶ 235], F_UINT [▶ 234], F_UDINT [▶ 233], F_ULINT [▶ 235], F_STRING [▶ 232], F_REAL [▶ 232], F_LREAL [▶ 231].

VAR_OUTPUT

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

bError: This output is set if an error occurs during formatting.

nErrId: Returns the format error code [▶ 352] if the bError output is set.

sOut: If successful, this output returns the formatted output string (type: T_MaxString)

Example:

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

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

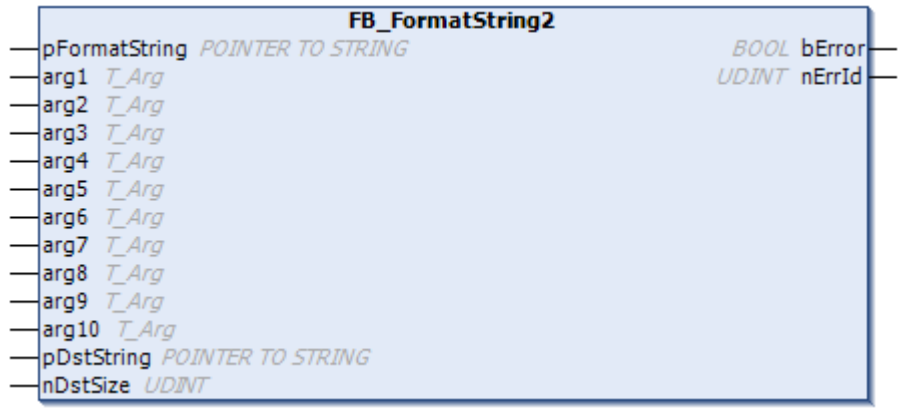
The result:

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

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.23 FB_FormatString2



This function block can be used for converting up to 10 arguments (similar to fprintf) into a string and formatting them according to the format specification [▶ 349] (e.g. '%+20.5f' or 'Measure X: %+.10d, Y: %+.10d'). The formatting takes place in the same PLC cycle. This means that the output string is available immediately after calling the function block.

As opposed to the function block FB_FormatString [▶ 64], the size of the format string and the output string is not limited to 255 characters. However, each argument is limited to a representation of 250 characters at the most. The function block outputs an error if the number of characters is exceeded or if the output string is too small for the formatted character string.

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: Pointer to the format specification as a string. The address must be assigned each time the function block is called. The operator ADR() can be used for the assignment.

arg1 to **arg10**: Arguments to be formatted (type: T_Arg [▶ 316]). The following helper functions can be used for converting different types of PLC variables into the required data type T_Arg [▶ 316]: F_BYTE [▶ 228], F_WORD [▶ 235], F_DWORD [▶ 229], F_LWORD [▶ 231], F_SINT [▶ 232], F_INT [▶ 230], F_DINT [▶ 229], F_LINT [▶ 231], F_USINT [▶ 235], F_UINT [▶ 234], F_UDINT [▶ 233], F_ULINT [▶ 235], F_STRING [▶ 232], F_REAL [▶ 232], F_LREAL [▶ 231].

pDstString: Pointer to the resulting STRING variable. If successful, the formatted character string will be written here. The address must be assigned each time the function block is called. The operator ADR() can be used for the assignment.

nDstSize: Size of the resulting STRING variable in bytes. The operator SIZEOF() can be used for the assignment.

VAR_OUTPUT

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

bError: TRUE if an error occurred during the formatting.

nErrId: Returns the format error code [▶ 352] if the bError output is set.

Sample

```
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 );
```

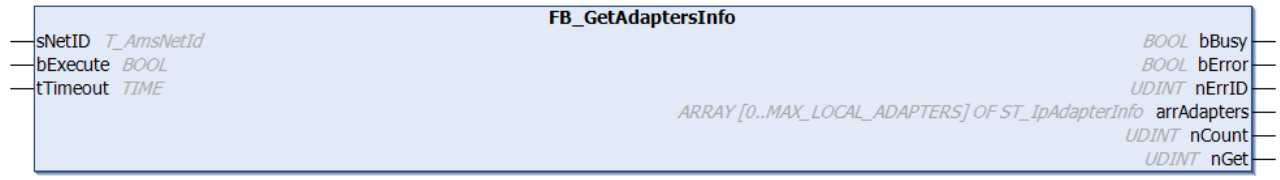
Result:

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

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.35.0

3.24 FB_GetAdaptersInfo



This function block can be used to read adapter information for a TwinCAT PC. The maximum number of adapter data that can be read is currently limited to MAX_LOCAL_ADAPTERS + 1 (default = 6).

VAR_INPUT

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

sNetID: This parameter can be used to specify the AmsNetID of the TwinCAT computer whose adapter information is to be read. For the local PC an empty string may be specified (type: T_AmsNetID).

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded during execution of the command.

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: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrID: Supplies the ADS error number [▶ 353] when the *bError* output is set.

arrAdapters: Array variable containing the last read adapter information. Each array element provides information for one adapter (type: ST_IpAdapterInfo [▶ 311]).

nCount: Maximum number of local adapters that were found.

nGet: Number of valid entries in the *arrAdapters* output variable.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64)	Tc2_Utilities (System)

3.25 FB_GetAdaptersInfoEx

This function block can be used to read adapter information for a TwinCAT PC. The maximum number of adapter information read is limited to 64.

After a successful call, the read process is finished and the read adapter information can be copied using the Get() method.

VAR_INPUT

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

sNetID: This parameter can be used to specify the AmsNetID of the TwinCAT computer whose adapter information is to be read. For the local PC an empty string may be specified (type: T_AmsNetID).

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded during execution of the command.

VAR_OUTPUT

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

bBusy: If the function block is activated, this output is set. It remains set until a feedback is received.

bError: If an error should occur during the transmission of the command, this output is set after the *bBusy* output has been reset.

nErrorID: Returns the ADS error number [▶ 353] when the *bError* output is set.

nAdapters: Number of local adapters found. Their adapter information has been read and can be copied using the Get() method.

Get() method

Once the function block has been called successfully, the read operation is complete and the adapter information read can be copied using the Get() method.

It is possible to copy the information of all adapters together. Likewise, the information per adapter can be copied individually one after the other.

When called, a local array (type: ST_IpAdapterInfo [▶ 311]) is specified. Each array element thus provides information of an adapter.

```
METHOD Get : BOOL
VAR_INPUT
    pAdapters        : POINTER TO ST_IpAdapterInfo; // pointer to array of adapter info (variable ar
ray length)
    nAdaptersSize    : UDINT; // size in bytes of array of adapter info
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.48.0

Also see about this

- ST_IPAdapterInfo [▶ 311]

3.26 FB_GetDeviceIdentificationEx

The block reads the device ID. Allows longer strings for hardware model and hardware serial number than FB_GetDeviceIdentification [▶ 26].

VAR_INPUT

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

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

sNetId: This parameter can be used to specify the AmsNetID of the TwinCAT computer whose device ID is to be read (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: If an error should occur during the transfer of the command, then this output is set once the bBusy output was reset.

nErrorId: Supplies the ADS error number [▶ 353] when the bError output is set.

stDevIdent: Provides the device ID (type: ST_DeviceIdentificationEx [▶ 307]).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.27 FB_GetDongleSystemID

The function block FB_GetDongleSystemID reads the system ID and the volume ID of the TwinCAT 3 license dongle as GUID.

VAR_INPUT

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

bExecute: The function block is activated by a positive edge at this input.

tTimeout: Timeout time that must not be exceeded when the command is executed.

sNetId: AmsNetId (AMS network identifier) of the TwinCAT computer whose license status is to be read (type: T_AmsNetId). If it is to be run on the local computer, an empty string can be entered.

stAmsAddr: Network address (AmsNetId and port) of the license dongle (type: AMSADDR)

VAR_OUTPUT

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

```
    stSystemID : GUID;
    stVolumeID : GUID;
END_VAR
```

bBusy: TRUE, as long as the function block is active.

bError: TRUE if an error occurs during command execution.

nErrorId: Supplies the ADS error number when the bError output is set.

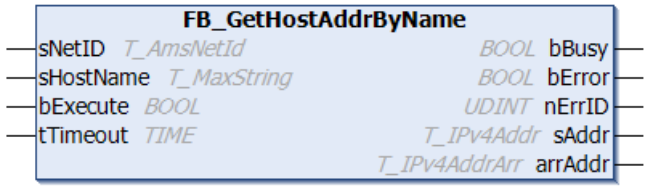
stSystemID: System ID of the license dongle (type: GUID)

stVolumeID: Volume ID of the license dongle (type: GUID)

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4022	PC or CX (x64, x86)	Tc2_Utilties >= 3.3.24.0

3.28 FB_GetHostAddrByName



This function block can be used to read the (IPv4) Internet protocol network address for the specified host name. The address is returned as string and byte array.

VAR_INPUT

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

sNetID: Here you can specify the network address of the TwinCAT computer on which the command is to be executed (type: T_AmsNetID). For the local computer (default) an empty string may be specified.

sHostName: Host name as string (type: T_MaxString). E.g.: 'DataServer1'.

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded during execution of the command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrID: Supplies the ADS error number [▶ 353] when the *bError* output is set.

sAddr: Internet protocol network address (IPv4) as string (type: T_Ipv4Addr). E.g.: '172.16.7.199'

arrAddr: Internet protocol network address as byte array (type: T_Ipv4AddrArr).

Sample:

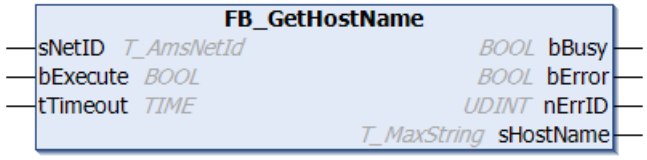
```
PROGRAM MAIN
VAR
    fbGet    : FB_GetHostAddrByName;
    bGet     : BOOL := TRUE;
    bError   : BOOL;
    nErrID   : UDINT;
    sIPv4    : T_IPv4Addr;(* Result: '87.106.8.100' *)
    arrIPv4  : T_IPv4AddrArr;
    state    : BYTE;
END_VAR
```

```
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
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Utilities (System)

3.29 FB_GetHostName



This function block can be used to read the host name of a TwinCAT PC.

VAR_INPUT

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

sNetID: This parameter can be used to specify the network address of the TwinCAT computer whose host name is to be read (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded during execution of the command.

VAR_OUTPUT

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

```
    nErrID    : UDINT;
    sHostName : T_MaxString;
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrID: Supplies the ADS error number [▶ 353] when the *bError* output is set.

sHostName: Host name as string (type: T_MaxString).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Utilities (System)

3.30 FB_GetLicenseDongles

The function block determines the number of connected license dongles and returns address and status.

VAR_INPUT

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

bExecute: The function block is activated by a positive edge at this input.

tTimeout: Timeout time that must not be exceeded when the command is executed.

sNetId: AmsNetId (AMS network identifier) of the TwinCAT computer whose license status is to be read (type: T_AmsNetId). If it is to be run on the local computer, an empty string can be entered.

VAR_OUTPUT

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

bBusy: TRUE, as long as the function block is active.

bError: TRUE if an error occurs during command execution.

nErrorId: Supplies the ADS error number when the bError output is set.

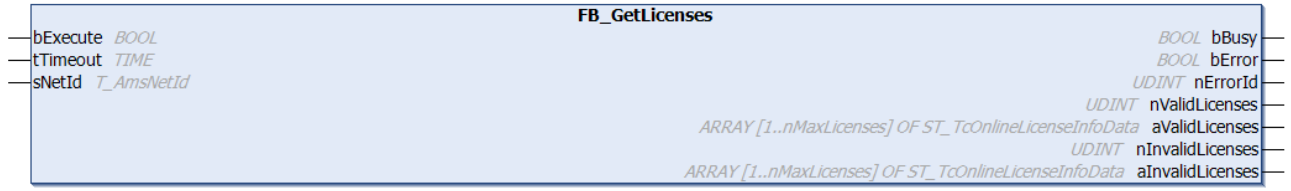
nLicenseDeviceDongles: Number of license dongles

aLicenseDeviceDongles: Identification data of the connected license dongles

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4022	PC or CX (x64, x86)	Tc2_Utilties >= 3.3.24.0

3.31 FB_GetLicenses



The function block reads the valid and invalid TwinCAT licenses.

VAR_INPUT

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

bExecute: The function block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

sNetId: This is the parameter to specify the AmsNetID (type: T_AmsNetID) of the TwinCAT computer, whose current TwinCAT licenses are to be read. If it is to be run on the local computer, an empty string can be entered.

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: When the function block is activated this output is set. It remains set until a feedback is received.

bError: If an error should occur during the transfer of the command, then this output is set once the bBusy output was reset.

nErrorId: Supplies the ADS error number [▶ 353] when the bError output is set.

nValidLicenses: Returns the number of valid TwinCAT licenses.

aValidLicenses: Returns a list of valid TwinCAT licenses of data type ST_TcOnlineLicenseInfodata [▶ 314].

nInvalidLicenses: Returns the number of invalid TwinCAT licenses.

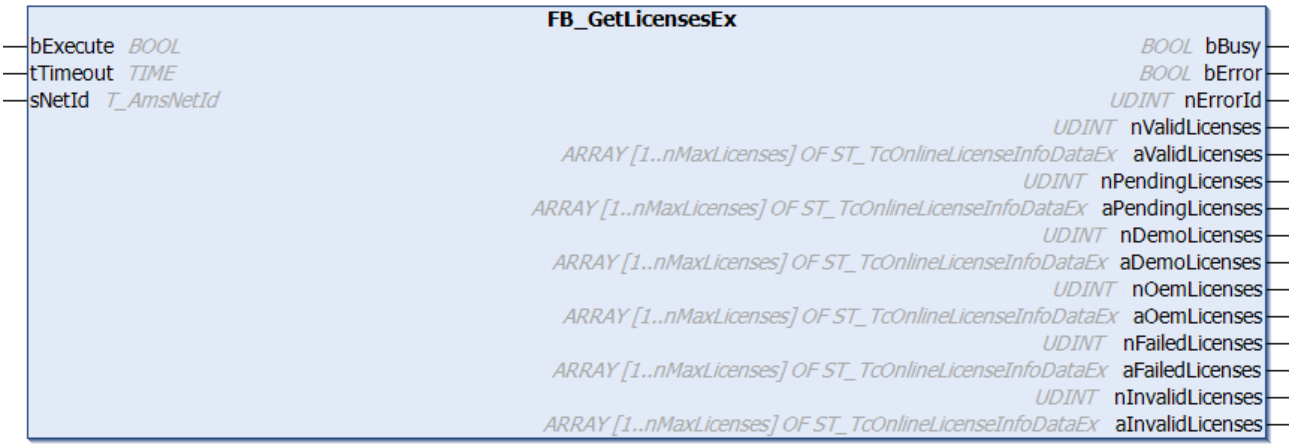
aInvalidLicenses: Returns a list of invalid TwinCAT licenses of data type ST_TcOnlineLicenseInfodata [▶ 314].

By default, the maximum number of license list entries is 50. This limit can be changed in the parameter list of the library via nMaxLicenses.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4018	PC or CX (x86, x64, ARM)	Tc2_Utilities (system) v3.3.9.0 or higher

3.32 FB_GetLicensesEx



The function block FB_GetLicensesEx determines the status of all TwinCAT 3 licenses and OEM licenses.

VAR_INPUT

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

bExecute: The function block is activated by a positive edge at this input.

tTimeout: Timeout time that must not be exceeded when the command is executed.

sNetId: AmsNetId (AMS network identifier) of the TwinCAT computer whose license status is to be read (type: T_AmsNetId). If it is to be run on the local computer, an empty string can be entered.

VAR_OUTPUT

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

bBusy: TRUE, as long as the function block is active.

bError: TRUE if an error occurs during command execution.

nErrorId: Supplies the ADS error number when the bError output is set.

nValidLicenses: Number of valid licenses

aValidLicenses: Information on valid licenses

nPendingLicenses: Number of open licenses

aPendingLicenses: Information on outstanding licenses

nDemoLicenses: Number of valid demo licenses

aDemoLicenses: Information on valid demo licenses

nOemLicenses: Number of valid OEM licenses

aOemLicenses: Information on valid OEM licenses

nFailedLicenses: Number of failed licenses

aFailedLicenses: Information on valid licenses

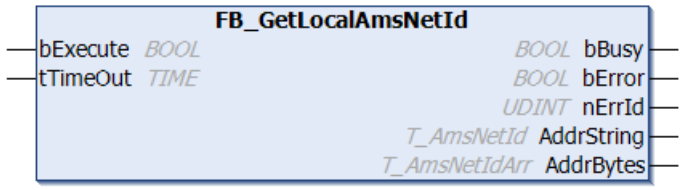
nInvalidLicenses: Number of invalid licenses

aInvalidLicenses: Information on invalid licenses

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4022	PC or CX (x64, x86)	Tc2_Utilties >= 3.3.24.0

3.33 FB_GetLocalAmsNetId



This function block can be used to read the network address (AmsNetID) of a local TwinCAT PC.

VAR_INPUT

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

bExecute: the block is activated by a positive edge at this input.

tTimeOut: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrId: Supplies the ADS error number [▶ 353] when the *bError* output is set.

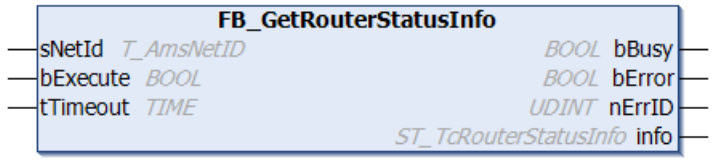
AddrString: AmsNetID of the local PC as string (type: T_AmsNetID).

AddrBytes: AmsNetID of the local PC as byte array (type: T_AmsNetIDArr).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.34 FB_GetRouterStatusInfo



The function block FB_GetRouterStatusInfo can be used to read status information for the TwinCAT router from the PLC (available memory, number of registered ports etc.).

VAR_INPUT

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

sNetId: Here you can enter a string with the network address of the TwinCAT computer whose TwinCAT router information is to be read (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

bExecute: the block is activated by a positive edge at this input.

tTimeOut: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrId: Supplies the ADS error number [▶ 353] when the *bError* output is set.

info: Structure variable with TwinCAT router status information (type: ST_TcRouterStatusInfo [▶ 315]).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.35 FB_GetSystemId

The function block reads the system ID as GUID (see "About TwinCAT..." at the TwinCAT icon in the system tray).

VAR_INPUT

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

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

sNetId: This parameter can be used to specify the AmsNetID of the TwinCAT computer whose system ID is to be read (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: If an error should occur during the transfer of the command, then this output is set once the bBusy output was reset.

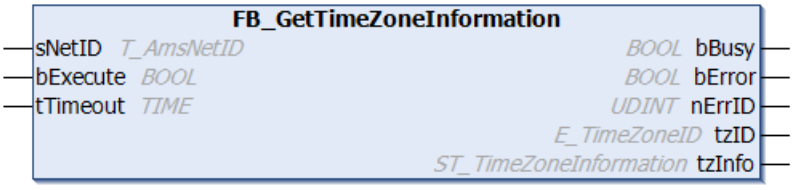
nErrorId: Supplies the ADS error number [▶ 353] when the bError output is set.

stSystemId: Supplies the system ID as GUID [▶ 303].

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.36 FB_GetTimeZoneInformation



This function block can be used to read the time zone settings of the operating system.

VAR_INPUT

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

sNetID: Here you can enter a string with the network address of the TwinCAT computer whose time zone settings are to be read (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrId: Supplies the ADS error number [▶ 353] when the *bError* output is set.

tzID: Additional summer/ winter time information (not always present) (type: E_TimeZoneID [▶ 302]).

tzInfo: If successful this structure variable supplies the current time zone information of the operating system (type: ST_TimeZoneInformation [▶ 315]).

Example:

See description of the FB_SetTimeZoneInformation [▶ 108] function block.

Further functions and function blocks for time and time zone:

- FB_TzSpecificLocalTimeToSystemTime [▶ 115]
- FB_TzSpecificLocalTimeToFileTime [▶ 29]
- FB_SystemTimeToTzSpecificLocalTime [▶ 111]
- FB_FileTimeToTzSpecificLocalTime [▶ 27]
- FB_SetTimeZoneInformation [▶ 108]
- NT_SetLocalTime [▶ 122]
- NT_GetTime [▶ 120]
- NT_SetTimeToRTCTime [▶ 123]
- F_TranslateFileTimeBias [▶ 238]
- FB_LocalSystemTime [▶ 89]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Utilities (System)

3.37 FB_GetVolumeId

The function block FB_GetVolumeId reads the System-ID and the Volume-System-ID as a GUID.

VAR_INPUT

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

bExecute: The block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

sNetId: The AmsNetId of the TwinCAT computer whose System-ID is to be read (type: T_AmsNetId) can be specified here. If it is to be run on the local computer, an empty string can be entered.

VAR_OUTPUT

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

bBusy: This output is set when the function block is activated, and remains set until an acknowledgement is received.

bError: If an error should occur during the transmission of the command, this output is set after the bBusy output has been reset.

nErrorId: Supplies the ADS error number [▶ 353] when the bError output is set.

stVolumeId: Supplies the Volume-System-ID as a GUID [▶ 303].

stSystemId: Supplies the System-ID as GUID [▶ 303].

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4018	PC or CX (x86, x64, ARM)	Tc2_Utilities

3.38 FB_HashTableCtrl

The hash table can be used as an efficient tool for finding individual data element among a larger number of data elements. The data objects must have a unique key. The key enables the data objects to be identified unambiguously and found quickly in the table.

The function block FB_HashTableCtrl can be used to realize a simple hash table in the PLC project, using the hashing with chaining (separate chaining) procedure. The hashing with chaining (separate chaining) procedure is used.

The maximum number of data elements cannot be changed at runtime and must be specified in advance. Adding/removing/finding of data elements is controlled through action calls. The function block features the following tasks:

- **A_Add** (adds a new data element to the table (key/value). If an element with the same key already exists, it is overwritten!)
- **A_GetFirst** (reads the first table data element. If successful, *getValue* supplies the associated value.)
- **A_GetNext** (reads the next table data element. The address *putPosPtr* must point to the previous data element!)
- **A_Lookup** (looks for a data element matching the key. If successful, *getValue* supplies the associated value.)
- **A_Remove** (removes a data element matching the key.)
- **A_RemoveAll** (removes all data elements)
- **A_RemoveFirst** (removes the first data element)
- **A_reset** (deletes all data elements and resets the table.)
- **A_GetIndexAtPosPtr** (supplies the array index of the data element at the address: *putPosPtr*. When successful, *getValue* returns the null-based array index. The value of *putValue* is not used. Please note that the value *getValue* returns a data element index, not the data element value!)

VAR_IN_OUT

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

hTable: Hash table handle (type: T_HHASHTABLE [▶ 319]). The handle must be initialized once with the function F_CreateHashTableHnd [▶ 258]before it can be used. A corresponding instance of the handle variable must be created and initialized for each table.

VAR_INPUT

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

key: Key (unsigned 32 bit number or pointer). This key enables a data object to be identified and found quickly in the table.

putValue: Value/data element (input parameter, 32/64 bit, unsigned number or pointer).

putPosPtr: Address for data element (input parameter, type: T_HashTableEntry [▶ 318])

VAR_OUTPUT

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

bOk: Returns TRUE if a new data element was added to/removed from or found in the table. Returns FALSE, if the searched data element was not found, the table is empty or an overflow occurred (table has no free data elements).

getValue: The value matching the key/data element (output parameter, 32/64 bit, unsigned number or pointer).

getPosPtr: The address for the data element (output parameter, type: T_HashTableEntry [▶ 318]).

Example:

See: Example: Hash table (FB_HashTableCtrl). [▶ 334]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.39 FB_LicFileCopyFromDongle

The function block copies a file from the license dongle to the hard disk. If the file is larger than the buffer (cbCopyLen), the file copying procedure is automatically split into several read and write operations until the whole file is copied. Only then does bBusy switch to FALSE.

VAR_INPUT

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

sNetIdSrc: AmsNetId (AMS network identifier) of the license dongle (type: T_AmsNetId)

- USB dongle: AmsNetId of the TwinCAT computer. If it is to be run on the local computer, an empty string can be entered.
- EL6070: AmsNetId of the EtherCAT master (see AdsAddr.netId in the InfoData of the EL6070)

nPortSrc: AMS port of the license dongle

- USB: ADS port of the ESB device (see ADS port on the ESB Device tab of the USB dongle; the default is 16#7100)
- EL6070: ADS port of the EtherCAT Terminal (see AdsAddr.port in the InfoData of the EL6070)

sNetIdDest: AmsNetId (AMS network identifier) of the TwinCAT computer. If it is to be run on the local computer, an empty string can be entered.

sFileNameSrc: Name of the file on the license dongle

sFilePathNameScr: Path name of the file on the hard disk. (Type: T_MaxString)

pCopyBuff: Buffer address for write

cbCopyLen: Count of bytes for write

bExecute: The function block is activated by a positive edge at this input.

dwPassCode: Passcode for file access

tTimeout: Timeout time that must not be exceeded when the command is executed.

VAR_OUTPUT

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

bBusy: TRUE, as long as the function block is active.

bError: TRUE if an error occurs during command execution.

nErrorId: Supplies the ADS error number when the bError output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4022	PC or CX (x64, x86)	Tc2_Utilties >= 3.3.26.0

3.40 FB_LicFileCopyToDongle

The function block copies a file from the hard disk to the license dongle. If the file is larger than the buffer (cbCopyLen), the file copying procedure is automatically split into several read and write operations until the whole file is copied. Only then does bBusy switch to FALSE.

VAR_INPUT

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

sNetIdSrc: AmsNetId (AMS network identifier) of the TwinCAT computer. If it is to be run on the local computer, an empty string can be entered. (type: T_AmsNetId)

sNetIdDest: AmsNetId (AMS network identifier) of the license dongle (type: T_AmsNetId)

- USB dongle: AmsNetId of the TwinCAT computer. If it is to be run on the local computer, an empty string can be entered.
- EL6070: AmsNetId of the EtherCAT master (see AdsAddr.netId in the InfoData of the EL6070)

nPortDest: AMS port of the license dongle

- USB: ADS port of the ESB device (see ADS port on the ESB Device tab of the USB dongle; the default is 16#7100)
- EL6070: ADS port of the EtherCAT Terminal (see AdsAddr.port in the InfoData of the EL6070)

sFilePathNameScr: Path name of the file on the hard disk (type: T_MaxString)

sFileNameDest: Name of the file on the license dongle

pCopyBuff: Buffer address for write

cbCopyLen: Count of bytes for write

bExecute: The function block is activated by a positive edge at this input.

dwPassCode: Passcode for file access

tTimeout: Timeout time that must not be exceeded when the command is executed.

VAR_OUTPUT

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

bBusy: TRUE, as long as the function block is active.

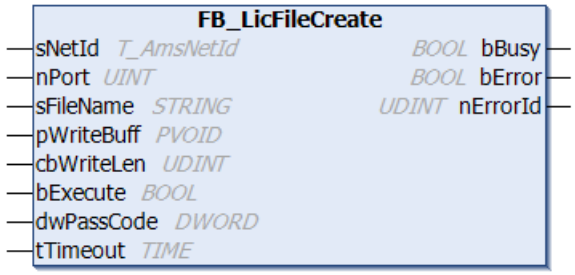
bError: TRUE if an error occurs during command execution.

nErrorId: Supplies the ADS error number when the bError output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4022	PC or CX (x64, x86)	Tc2_Utilties >= 3.3.26.0

3.41 FB_LicFileCreate



The function block creates a file on the license dongle. A positive edge at bExecute triggers writing of the data from the buffer (pWriteBuff and cbWriteLen) directly to a new file on the dongle.

VAR_INPUT

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

sNetId: AmsNetId (AMS network identifier) of the license dongle (type: T_AmsNetId)

- USB dongle: AmsNetId of the TwinCAT computer. If it is to be run on the local computer, an empty string can be entered.
- EL6070: AmsNetId of the EtherCAT master (see AdsAddr.netId in the InfoData of the EL6070)

nPort: AMS port of the license dongle

- USB: ADS port of the ESB device (see ADS port on the ESB Device tab of the USB dongle; the default is 16#7100)
- EL6070: ADS port of the EtherCAT Terminal (see AdsAddr.port in the InfoData of the EL6070)

sFileName: Name of the file to be created

pWriteBuff: Buffer address for write

cbWriteLen: Count of bytes for write

bExecute: The function block is activated by a positive edge at this input.

dwPassCode: Passcode for file access

tTimeout: Timeout time that must not be exceeded when the command is executed.

VAR_OUTPUT

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

bBusy: TRUE, as long as the function block is active.

bError: TRUE if an error occurs during command execution.

nErrorId: Supplies the ADS error number when the bError output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4022	PC or CX (x64, x86)	Tc2_Utilties >= 3.3.26.0

3.42 FB_LicFileDelete

The function block deletes a file from the license dongle. The file name and file length are zeroed, and the data bytes of the file to be deleted are released on the dongle but not overwritten.

VAR_INPUT

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

sNetId: AmsNetId (AMS network identifier) of the license dongle (type: T_AmsNetId)

- USB dongle: AmsNetId of the TwinCAT computer. If it is to be run on the local computer, an empty string can be entered.
- EL6070: AmsNetId of the EtherCAT master (see AdsAddr.netId in the InfoData of the EL6070)

nPort: AMS port of the license dongle

- USB: ADS port of the ESB device (see ADS port on the ESB Device tab of the USB dongle; the default is 16#7100)
- EL6070: ADS port of the EtherCAT Terminal (see AdsAddr.port in the InfoData of the EL6070)

sFileName: Name of the file to be deleted

pWriteBuff: Buffer address for write

cbWriteLen: Count of bytes for write

dwPassCode: Passcode for file access

tTimeout: Timeout time that must not be exceeded when the command is executed.

VAR_OUTPUT

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

bBusy: TRUE, as long as the function block is active.

bError: TRUE if an error occurs during command execution.

nErrorId: Supplies the ADS error number when the bError output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4022	PC or CX (x64, x86)	Tc2_Utilties >= 3.3.26.0

3.43 FB_LicFileGetStorageInfo

The function block reads the StorageInfo of the license dongle and the file directory.

The StorageInfo contains administrative data of the storage medium (such as capacity, number of free bytes, number of files,…) and an array of the individual file entries (name, size, attributes,… of the file).

VAR_INPUT

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

sNetId: AmsNetId (AMS network identifier) of the license dongle (type: T_AmsNetId)

- USB dongle: AmsNetId of the TwinCAT computer. If it is to be run on the local computer, an empty string can be entered.
- EL6070: AmsNetId of the EtherCAT master (see AdsAddr.netId in the InfoData of the EL6070)

nPort: AMS port of the license dongle

- USB: ADS port of the ESB device (see ADS port on the ESB Device tab of the USB dongle; the default is 16#7100)

- EL6070: ADS port of the EtherCAT Terminal (see AdsAddr.port in the InfoData of the EL6070)

bExecute: The function block is activated by a positive edge at this input.

dwPassCode: Passcode for file access (only for specially protected files)

tTimeout: Timeout time that must not be exceeded when the command is executed.

VAR_OUTPUT

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

bBusy: TRUE, as long as the function block is active.

bError: TRUE if an error occurs during command execution.

nErrorId: Supplies the ADS error number when the bError output is set.

nFileEntries: Number of files on the license dongle

VAR_IN_OUT

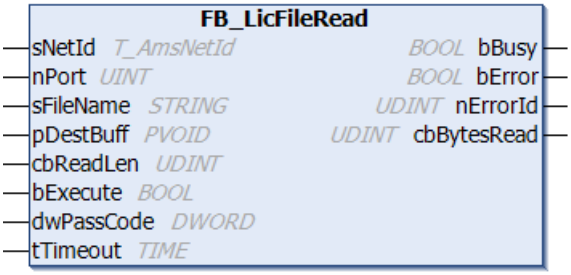
```
VAR_IN_OUT
    stStorageInfo : ST_LicStorageInfo;
END_VAR
```

stStorageInfo: StorageInfo of the license dongle (type: ST_LicStorageInfo)

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4022	PC or CX (x64, x86)	Tc2_Utilties >= 3.3.26.0

3.44 FB_LicFileRead



The function block reads a file from the license dongle to a buffer (pDestBuff and cbReadLen) via a positive edge at bExecute. The buffer must be large enough for the file, otherwise only the first part of the file is read.

VAR_INPUT

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

sNetId: AmsNetId (AMS network identifier) of the license dongle (type: T_AmsNetId)

- USB dongle: AmsNetId of the TwinCAT computer. If it is to be run on the local computer, an empty string can be entered.

- EL6070: AmsNetId of the EtherCAT master (see AdsAddr.netId in the InfoData of the EL6070)

nPort: AMS port of the license dongle

- USB: ADS port of the ESB device (see ADS port on the ESB Device tab of the USB dongle; the default is 16#7100)
- EL6070: ADS port of the EtherCAT Terminal (see AdsAddr.port in the InfoData of the EL6070)

sFileName: Name of the file to be read

pDestBuff: Buffer address for read

cbReadLen: Count of bytes for read

bExecute: The function block is activated by a positive edge at this input.

dwPassCode: Passcode for file access

tTimeout: Timeout time that must not be exceeded when the command is executed.

VAR_OUTPUT

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

bBusy: TRUE, as long as the function block is active.

bError: TRUE if an error occurs during command execution.

nErrorId: Supplies the ADS error number when the bError output is set.

Requirements

Development environment	Target platform	PLC libraries to include
TwinCAT v3.1.4022	PC or CX (x64, x86)	Tc2_Utilties >= 3.3.26.0

3.45 FB_LinkedListCtrl

The function block FB_LinkedListCtrl can be used to implement a linked list in the PLC project. A double-linked list is created. A linked list allows values (known as nodes) to be stored. It is possible to iterate the list from the back to the front or the other way. Nodes can quickly be added or deleted.

It is not possible to change the maximum number of nodes at runtime; it must be specified before compiling. An array of type *T_LinkedListEntry* is used as a "node pool". Adding/removing/finding of nodes is controlled through action calls. The function block features the following tasks:

- **A_AddHeadValue** (adds a new node with the value *putValue* to the top of the list. The same value can be added more than once. If successful, *getPosPtr* returns the address while *getValue* returns the value of the new node.)
- **A_AddTailValue** (adds a new node with the value *putValue* to the end of the list. The same value can be added more than once. If successful, *getPosPtr* returns the address while *getValue* returns the value of the new node.)
- **A_FindNext** (searches for the next node (relative to *putPosPtr*) whose value is the same as *putValue*. If successful, *getPosPtr* returns the address while *getValue* returns the value of the node.)
- **A_FindPrev** (searches for the previous node (relative to *putPosPtr*) whose value is the same as *putValue*. If successful, *getPosPtr* returns the address while *getValue* returns the value of the node.)

- **A_GetNext** (navigates to the next node (relative to *putPosPtr*). The address *putPosPtr* must point the previous node! The value of *putValue* is not used.)
- **A_GetPrev** (navigates to the previous node (relative to *putPosPtr*) in the opposite direction to *A_GetNext*. The address *putPosPtr* must point the previous node! The value of *putValue* is not used.)
- **A_GetHead** (reads the starting node. If successful, *getPosPtr* returns the address of the node while *getValue* returns the associated value. The values of *putValue* and *putPosPtr* are not used.)
- **A_GetTail** (reads the final node. If successful, *getPosPtr* returns the address of the node while *getValue* returns the associated value. The values of *putValue* and *putPosPtr* are not used.)
- **A_RemoveHeadValue** (removes a node from the top of the list. If successful, *getPosPtr* returns the address while *getValue* returns the value of the node. The values of *putValue* and *putPosPtr* are not used.)
- **A_RemoveTailValue** (removes a node from the end of the list. If successful, *getPosPtr* returns the address while *getValue* returns the value of the node. The values of *putValue* and *putPosPtr* are not used.)
- **A_RemoveValueAtPosPtr** (searches for and removes a node with address *putPosPtr*. If successful, *getPosPtr* returns the address while *getValue* returns the value of the node. The value of *putValue* is not used.)
- **A_GetIndexAtPosPtr** (returns the array index (from the "node pool") of the node at address *putPosPtr*. When successful, *getValue* returns the null-based array index. The value of *putValue* is not used. Please note that the value of getValue is a node index, not the value of the node!)
- **A_SetValueAtPosPtr** (updates/sets the value of the node with *putValue* at the address *putPosPtr*. If successful, *getPosPtr* returns the address while *getValue* returns the value of the node.)
- **A_Reset** (deletes all list elements and resets the list.)

VAR_IN_OUT

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

hList: Linked List Handle (type: T_HLINKEDLIST [▶ 319]). The handle must be initialized once with the function F_CreateLinkedListHnd [▶ 259] before being used. An associated instance of the handle variable must be created and initialized for each linked list

VAR_INPUT

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

putValue: Value/data element (input parameter, 32/64 bit, unsigned number or pointer).

putPosPtr: Address of the node element (input parameter, type: T_LinkedListEntry [▶ 320]).

VAR_OUTPUT

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

bOk: Result of the last action called. Returns TRUE if a new node element could be added, removed, or found in the list. FALSE is returned if the node element that was searched for could not be found, the list is empty, or has overflowed (no more free node elements).

getValue: Value/data element (output parameter, 32/64 bit, unsigned number or pointer).

getPosPtr: Address of the node element (output parameter, type: T_LinkedListEntry [▶ 320]).

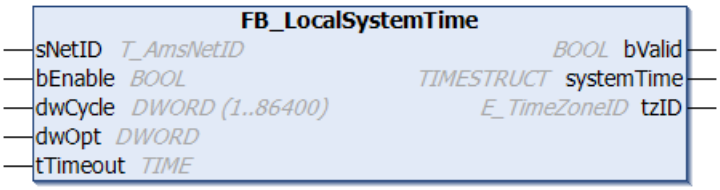
Example:

See: Example: Linked list (FB_LinkedListCtrl). [▶ 338]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.46 FB_LocalSystemTime



In some applications the local Windows system time is synchronized via the SNTP time server or a radio clock. In many cases the local Windows system time has to be used in the PLC (e.g. in the form of timestamp log messages to the HMI). The local Windows system time is displayed in the taskbar. For such applications the FB_LocalSystemTime function block can be useful.

This function block internally combines the functions of the following function blocks: RTC_EX2 [▶ 137], NT_GetTime [▶ 120], FB_GetTimeZoneInformation [▶ 77] and NT_SetTimeToRTCTime [▶ 123]. The RTC_EX2 function block can be used for generating time stamps for log outputs, for example. However, this block has the disadvantage that its time is not synchronized with the local Windows system time and has to be resynchronized cyclically via the NT_GetTime function block (see RTC block examples in the documentation). Cyclic synchronization of the internal time (*systemTime* output) is already implemented in the function block. The cycle time can be configured via the *dwCycle* input. The function block also provides time zone information (summer time/winter time).

The FB_LocalSystemTime function block must be called cyclically (e.g. every second or during each PLC cycle). This is necessary to enable the time between synchronizations to be calculated.

Jitter!

The local Windows system time is read with the aid of acyclic services (ADS function blocks). Due to the system characteristics the runtime of the ADS commands cannot be specified/estimated. Differences in command runtimes may lead to time jitter at the systemTime output, depending on the operating system, the synchronization interval and the PLC cycle time. For this reason, the time provided by the function block is only conditionally suitable for more precise measuring tasks. However, the accuracy is adequate, for example, for building automation applications.

Switching between daylight-saving time and standard time

The function block cannot be called exactly at the time when the switchover from daylight-saving time to standard time and vice versa takes place. In order to avoid complex calculations, the following implementation was chosen (explained in the example).

In our example the function block synchronises its own time with the local Windows system time (grey) every 60 seconds.
The PLC application needs and reads the time in the function block (e.g. every 30 seconds, blue). In our example the switchover between daylight-saving time and standard time is detected with a delay of 15 seconds. This behaviour should be unproblematic for most applications.

Switching from standard time to daylight-saving time

- ...
- 30-03-2008-01:58:10, tzID = standard time
- 30-03-2008-01:58:15, after internal synchronisation
- 30-03-2008-01:58:40, tzID = standard time
- 30-03-2008-01:59:10, tzID = standard time
- 30-03-2008-01:59:15, after internal synchronisation

- 30-03-2008-01:59:40, tzID = standard time
- 30-03-2008-02:00:00, the operating system changes the time from 2 am to 3 am
- 30-03-2008-02:00:10, tzID = standard time (still!)
- 30-03-2008-03:00:15, after internal synchronisation, subsequent tzID = daylight-saving time
- 30-03-2008-03:00:40, tzID = daylight-saving time
- 30-03-2008-03:01:10, tzID = daylight-saving time
- 30-03-2008-03:01:15, after internal synchronisation
- 30-03-2008-03:01:40, tzID = daylight-saving time
- ...

Switching from daylight-saving time to standard time

- ...
- 26-10-2008-02:58:10, tzID = daylight-saving time
- 26-10-2008-02:58:15, after internal synchronisation
- 26-10-2008-02:58:40, tzID = daylight-saving time
- 26-10-2008-02:59:10, tzID = daylight-saving time
- 26-10-2008-02:59:15, after internal synchronisation
- 26-10-2008-02:59:40, tzID = daylight-saving time
- 26-10-2008-03:00:00, the operating system changes the time from 3 am to 2 am
- 26-10-2008-03:00:10, tzID = daylight-saving time (still!)
- 26-10-2008-02:00:15, after internal synchronisation, subsequent tzID = standard time
- 26-10-2008-02:00:40, tzID = standard time
- 26-10-2008-02:01:10, tzID = standard time
- 26-10-2008-02:01:15, after internal synchronisation
- 26-10-2008-02:01:40, tzID = standard time
- ...

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: A string containing the network address of the TwinCAT computer whose time is to be used for the synchronization can be given here (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

bEnable: A rising edge at this input triggers immediate synchronization of the internal time with the local Windows system time. The output *bValid* remains set to FALSE until the synchronization is complete. The first rising edge activates the cyclic synchronization. The subsequent cyclic synchronizations are then executed automatically. In most cases the application only has to set this input to TRUE once.

dwCycle: Cycle time (in seconds) during which the function block resynchronizes its own time. Cyclic synchronization is activated after the first rising edge at the *bEnable* input. Default: Synchronization every 5 seconds.

dwOpt: Additional option parameters. The following parameters are currently available:

- Bit 0: If this is set, the Windows system time is additionally synchronized cyclically with the hardware clock (RTC) (corresponds to the function NT_SetTimeToRTCTime). Default: Activated.
  This option is irrelevant for a Windows CE system.

tTimeout: States the length of the timeout that may not be exceeded by execution of the internal ADS command.

VAR_OUTPUT

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

bValid: The time at the *systemTime* output is invalid if this output is FALSE. The time is valid if TRUE (i.e. it was synchronized with the local Windows time at least once).

systemTime: Local Windows system time (type: TIMESTRUCT [▶ 321]).

tzID: Time zone information (summer time, winter time) (type: E_TimeZoneID [▶ 302]).

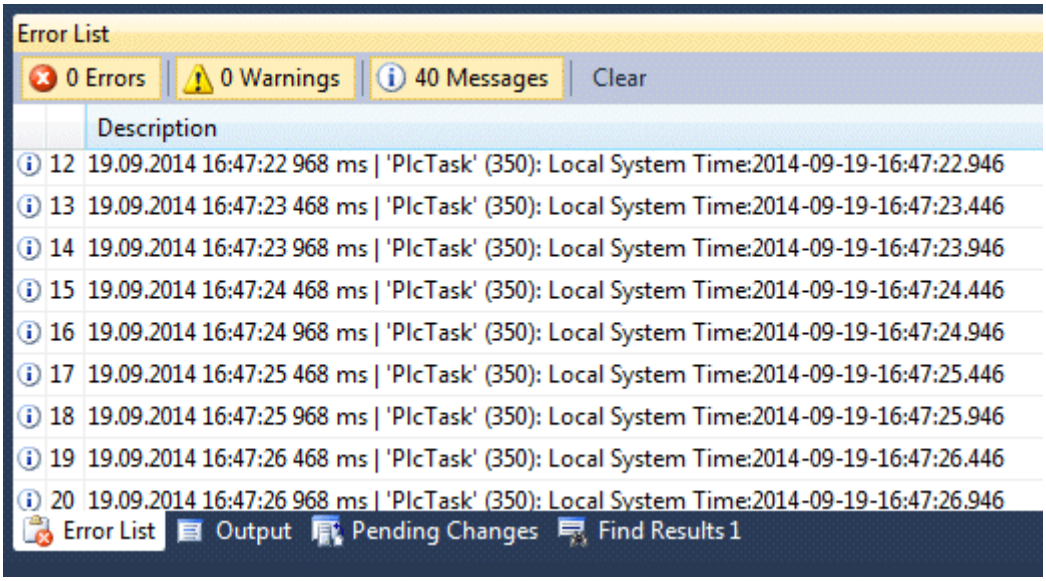
Example:

In the example the FB_LocalSystemTime function block is activated on program startup (rising edge at *bEnable* input). Once the time has been synchronized (*bValid* = TRUE), the PLC writes a message to the TwinCAT System Logview every 500 ms. The internal synchronization is carried out every second.

```
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
```

The written messages can be viewed in the TwinCAT Logview.



Further functions and function blocks for time and time zone:

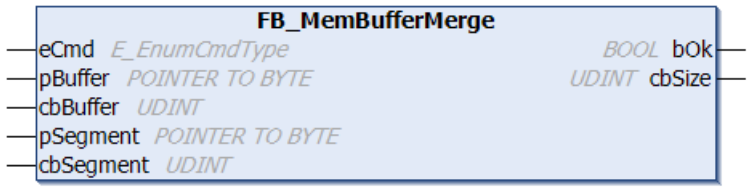
- FB_TzSpecificLocalTimeToSystemTime [▶ 115]
- FB_TzSpecificLocalTimeToFileTime [▶ 29]
- FB_SystemTimeToTzSpecificLocalTime [▶ 111]
- FB_FileTimeToTzSpecificLocalTime [▶ 27]
- FB_GetTimeZoneInformation [▶ 77]
- FB_SetTimeZoneInformation [▶ 108]
- NT_SetLocalTime [▶ 122]

- NT_GetTime [▶ 120]
- NT_SetTimeToRTCTime [▶ 123]
- F_TranslateFileTimeBias [▶ 238]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Utilities (System)

3.47 FB_MemBufferMerge



This function block consolidates individual smaller data segments to form a larger data segment. The target buffer must be transferred as input parameter to the block. No further data bytes are added, if the segment to be added exceeds the remaining free buffer size.

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: Control parameter for the enumeration block (type: E_EnumCmdType [▶ 297]). eEnumCmd_First adds the first segment, eEnumCmd_Next adds the next segment. No other parameters are used.

pBuffer: Address (pointer) for the target buffer variable. The address can be determined with the ADR operator.

cbBuffer: Maximum available size (in bytes) of the target buffer variables. The size can be determined with the SIZEOF operator.

pSegment: Address (pointer) for the next data segment to be added (optional, may be zero). The address can also be determined with the ADR operator.

cbSegment: Size of the next data segment to be added (optional, may be zero). The size can also be determined with the SIZEOF operator.

VAR_OUTPUT

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

bOk: TRUE = success, FALSE = buffer overflow or faulty input parameters.

cbSize: Current buffer fill status (number of data bytes in the buffer).

Example:

In the following example, the large data segment is converted to a hexadecimal string after the smaller data segments have been consolidated (for test purposes).

```
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
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.48 FB_MemBufferSplit

This function block splits a memory area (data buffer) into several smaller segments of certain maximum length as required. The function block returns a smaller partial segment, if the length of the last segment is smaller than required.

VAR_INPUT

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

eCmd: Control parameter for the function block (type: E_EnumCmdType [▶ 297]). eEnumCmd_First returns the first segment, eEnumCmd_Next returns the next segment. No other parameters are used.

pBuffer: Address (pointer) of the data buffer to be divided. The address can be determined with the ADR operator.

cbBuffer: Length of the data buffer to be divided. The length can be determined with the SIZEOF operator.

cbSize: Maximum segment size into which the data buffer is to be divided.

VAR_OUTPUT

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

bOk: TRUE = success, FALSE = error, invalid parameter value or no further segment available.

pSegment: Address (pointer) to the next data segment.

cbSegment: Length (bytes) of the next data segment.

bEOS: End of segment. TRUE = last segment. FALSE = further segments follow.

Example:

In the following example the *buffer* variable is divided into 5-byte segments. The returned segments are converted to a hexadecimal string for test purposes.

```
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
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.49 FB_MemRingBuffer

The function block FB_MemRingBuffer can be used to write data sets with different lengths in a ring buffer or to read previously written data sets from the ring buffer. The written data sets are read in the same order as they were previously written to the ring buffer, based on the FIFO principle, i.e. the oldest entries are read first. The buffer memory is made available to the function block via the *pBuffer*/*cbBuffer* input variables. Writing/reading of data sets is controlled via action calls. The function block features the following tasks:

- **A_AddTail** (writes a new data set into the ring buffer.)
- **A_GetHead** (reads the oldest data set in the ring buffer, but does not remove it.)

- **A_RemoveHead** (reads and removes the oldest data set from the ring buffer.)
- **A_reset** (deletes all data sets in the ring buffer.)

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: The address of the PLC variable or of a buffer variable that contains the value data that is to be written. The address can be determined with the ADR operator. The programmer is himself responsible for dimensioning the buffer variable in such a way that *cbWrite* data bytes can be taken from it.

cbWrite: The number of value data bytes that are to be written. (In the case of string variables this includes the final null).

pRead: The address of the PLC variables or of a buffer variable into which the value data that has been read is to be copied. The address can be determined with the ADR operator. The programmer is himself responsible for dimensioning the buffer variable in such a way that it can accept cbRead data bytes. The size of the buffer variables in bytes must be greater than or equal to the size of the data record that is to be read.

cbRead: The number of value data bytes to be read. If the buffer size is too small, data is not copied. The function block reports a buffer underflow error (bOk = FALSE), and the buffer size required for the next data record that is to be read is returned at the *cbReturn* output.

pBuffer: Address of a PLC variables (e.g. ARRAY[...] OF BYTES) to be used as buffer memory by the function block. The address can be determined with the ADR operator.

cbBuffer: Max. byte size of the PLC variable to be used as buffer memory. The size can be determined with the SIZEOF operator.

VAR_OUTPUT

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

bOk: Returns TRUE, if a new data set was added or removed successfully. Returns FALSE in the event of a buffer overflow or if no more entries are available in the buffer.

nCount: Returns the current number of queued data sets.

cbSize: Returns the current number of assigned data bytes in the buffer. The number of assigned data bytes is always greater than the actual number of written value data. Each data set is complemented with additional information, so that it can be located later.

cbReturn: The number of value data bytes successfully read. If a read buffer underflow error has occurred, this output supplies the necessary read buffer size in bytes. In this case is the *cbRead* length is too small.

Example:

See: Example: Memory ring FiFo (FB_MemRingBuffer). [▶ 332]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.50 FB_MemRingBufferEx

The function block FB_MemRingBufferEx can be used to write data sets with different lengths in a ring buffer or to read previously written data sets from the ring buffer. The written data sets are read in the same order as they were previously written to the ring buffer, based on the FIFO principle, i.e. the oldest entries are read first. The buffer memory is made available to the function block via the *pBuffer*/*cbBuffer* input variables. Writing/reading of data sets is controlled via action calls.

The functionality of this function block is similar to that of the FB_MemRingBuffer [▶ 94] function block. During reading of data sets the FB_MemRingBuffer copies the data into an external buffer variable. FB_MemRingBufferEx only provides a reference for the data set (address pointer/length). The application must then copy the data itself for further processing.

The function block features the following tasks:

- **A_AddTail** (writes a new data set into the ring buffer.)
- **A_GetHead** (returns a reference: address pointer/length of the oldest data set in the ring buffer, but does not remove it.)
- **A_FreeHead** (reads and removes the oldest data set from the ring buffer. The returned address pointer/length is zero! The free memory segment is released for a new data set.)
- **A_reset** (deletes all data sets in the ring buffer.)
- **A_GetFreeSize** (returns the byte size of the largest free memory segment in the buffer)

VAR_INPUT

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

pWrite: The address of the PLC variable or of a buffer variable that contains the value data that is to be written. The address can be determined with the ADR operator. The programmer is himself responsible for dimensioning the buffer variable in such a way that *cbWrite* data bytes can be taken from it.

cbWrite: The number of value data bytes that are to be written. (In the case of string variables this includes the final null). The size can be determined with the SIZEOF operator.

pBuffer: Address of a PLC variables (e.g. ARRAY[...] OF BYTES) to be used as buffer memory by the function block. The address can be determined with the ADR operator.

cbBuffer: Max. byte size of the PLC variable to be used as buffer memory. The size can be determined with the SIZEOF operator.

VAR_OUTPUT

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

bOk: Returns TRUE, if a new data set was added or removed successfully. Returns FALSE in the event of a buffer overflow or if no more entries are available in the buffer.

pRead: After a call of the action: *A_GetHead*, if successful (bOk=TRUE) this variable returns a reference (address pointer) for the oldest data set in the ring buffer. It returns zero if no more data sets are available in the ring buffer.

cbRead: After a call of the action: *A_GetHead*, if successful (bOk=TRUE) this variable returns the length of the oldest data set in the ring buffer. It returns zero if no more data sets are available in the ring buffer.

nCount: Returns the current number of queued data sets.

cbSize: Returns the current number of assigned data bytes in the buffer. The number of assigned data bytes is always greater than the actual number of written value data. Each data set is complemented with additional information, so that it can be located later.

cbFree: After a call of the action: *A_GetFreeSize*, returns the byte size of the largest free memory segment in the buffer. The data sets must use continuous addresses in the buffer memory, because the function block returns a reference for the data sets. This automatically leads to segmentation at the end of the buffer. This memory cannot be used if the new data set is greater than the free segment at the end of the buffer.

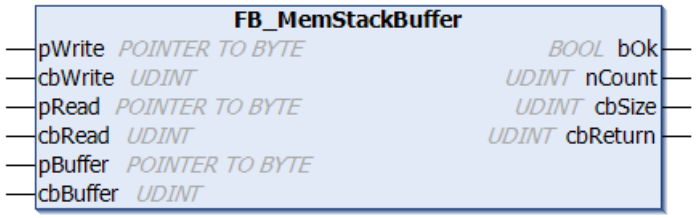
Example:

See: Example: Memory ring FiFo (FB_MemRingBufferEx) [▶ 333].

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.51 FB_MemStackBuffer



The function block FB_MemStackBuffer can be used to write data sets with different lengths in a buffer or to read previously written data sets from the buffer. The data sets are read based on the LIFO principle (last in - first out), i.e. in the reverse order in which they were written into the buffer. In other words, the latest entries are read first. The buffer memory is made available to the function block via the *pBuffer* and *cbBuffer* input variables. Writing/reading of data sets is controlled via action calls. The function block features the following tasks:

- **A_Push():** Writes a new data set into the buffer;
- **A_Top():** Reads the last added/latest data set from the buffer, but does not remove it;
- **A_Pop():** Reads and removes the last added/latest data set from the buffer;
- **A_Reset():** Deletes all data sets from the buffer;

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: The address of the PLC variable or of a buffer variable that contains the value data that is to be written. The address can be determined with the ADR operator. The programmer is himself responsible for dimensioning the buffer variable in such a way that *cbWrite* data bytes can be taken from it.

cbWrite: The number of value data bytes that are to be written. (In the case of string variables this includes the final null).

pRead: The address of the PLC variables or of a buffer variable into which the value data that has been read is to be copied. The address can be determined with the ADR operator. The programmer is himself responsible for dimensioning the buffer variable in such a way that it can accept cbRead data bytes. The size of the buffer variables in bytes must be greater than or equal to the size of the data record that is to be read.

cbRead: The number of value data bytes to be read. If the buffer size is too small, data is not copied. The function block reports a buffer underflow error (bOk = FALSE), and the buffer size required for the next data record that is to be read is returned at the *cbReturn* output.

pBuffer: Address of a PLC variables (e.g. ARRAY[...] OF BYTES) to be used as buffer memory by the function block. The address can be determined with the ADR operator.

cbBuffer: Max. byte size of the PLC variable to be used as buffer memory. The size can be determined with the SIZEOF operator.

VAR_OUTPUT

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

bOk: Returns TRUE, if a new data set was added or removed successfully. Returns FALSE in the event of a buffer overflow or if no more entries are available in the buffer.

nCount: Returns the current number of queued data sets.

cbSize: Returns the current number of assigned data bytes in the buffer. The number of assigned data bytes is always greater than the actual number of written value data. Each data set is complemented with additional information, so that it can be located later.

cbReturn: The number of value data bytes successfully read. If a read buffer underflow error has occurred, this output supplies the necessary read buffer size in bytes. In this case is the *cbRead* length is too small.

Example:

The following example illustrates a simple application of the function block. Strings of different lengths are to be buffered. A rising edge at *bReset* clears the buffer. If *bAdd* is set to TRUE, 10 new strings are written into the buffer. If *bRemove* is TRUE the string that was written last is removed from the buffer. A rising edge at *bGet* results in the string that was written last to be read but not removed.

Declaration part:

```
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
```

Program code:

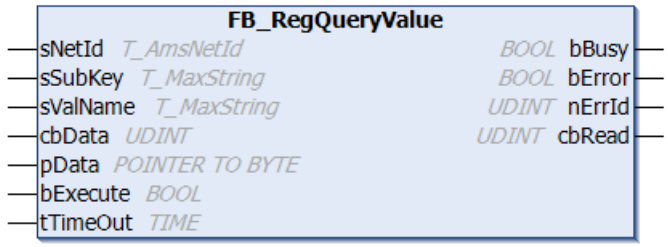
```
IF bReset THEN(* Clear buffer *)
    bReset := FALSE;
    fbStack.A_Reset( pBuffer := ADR( buffer ), cbBuffer := SIZEOF( buffer ) );
END_IF
```

```
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), cbRea
d := 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), cbRea
d := SIZEOF(getEntry) );
    IF fbStack.bOk THEN(* Success *)
        ;
    ELSE(* Buffer is empty *)
        ;
    END_IF
END_IF
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.52 FB_RegQueryValue



The system registry is a hierarchically structured tree. A node in the tree is referred to as a key. Each key can may contain subkeys and data values. The function block "FB_RegQueryValue" can be used to read individual system registry values from the branch with the predefined handle **HKEY_LOCAL_MACHINE**. If successful *cbData* data bytes are copied into the buffer with the address *pData*. The function block can be used to read any value types (e.g. REG_DWORD, REG_SZ) or binary data with unlimited byte length (REG_BINARY).

Comment:

The strings *sSubKey* and *sValueName* may not be empty!

HKEY_LOCAL_MACHINE\SOFTWARE\ for 64 bit operating systems

In a 64 bit Windows operating system all registry entries of and for 32 bit applications are not stored under HKEY_LOCAL_MACHINE\SOFTWARE\ but under HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\.
The function blocks FB_RegQueryValue and FB_RegSetValue work automatically below the WOW6432Node folder like any 32 bit application when a registry entry below the SOFTWARE folder is selected. The redirection is performed automatically by the operating system.

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: Here you can enter a string with the network address of the TwinCAT computer whose system registry is to be read (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

sSubKey: String with the subkey name (type: T_MaxString).

sValName: String with the value name (type: T_MaxString).

cbData: The number of value data bytes to be read.

pData: Address of a data buffer/variable into which the value data are to be copied. The address can be determined with the ADR operator. The programmer is responsible for dimensioning the data buffer such that it can accommodate cbData data bytes.

bExecute: the block is activated by a positive edge at this input.

tTimeOut: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

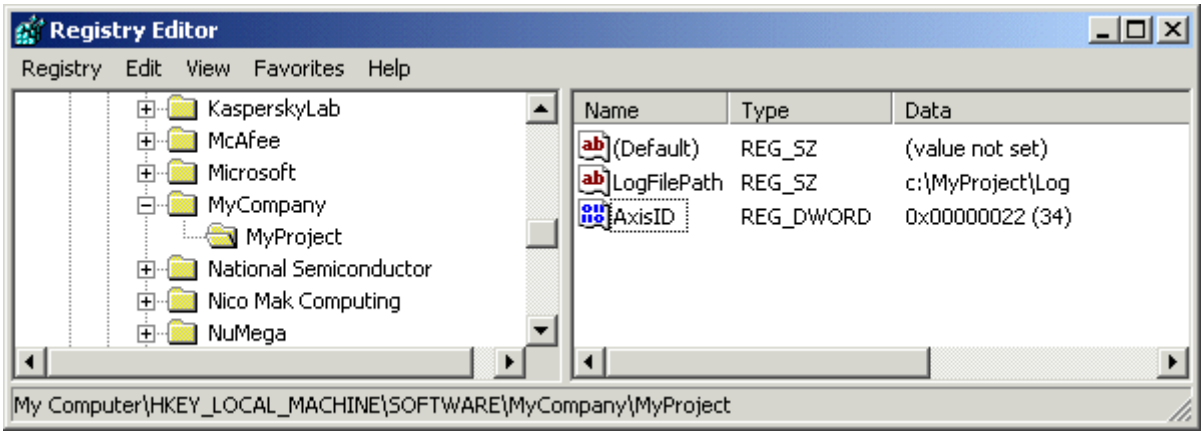
nErrId: Supplies the ADS error number [▶ 353] or the command-specific error code (table) when the *bError* output is set.

cbRead: The number of value data bytes successfully read.

Error Codes	Error description
0x00	No error
0x01	The key with the name **sSubKey** could not be opened/found.
0x02	The key value with the name **sValName** could not be opened/found.

Examples:

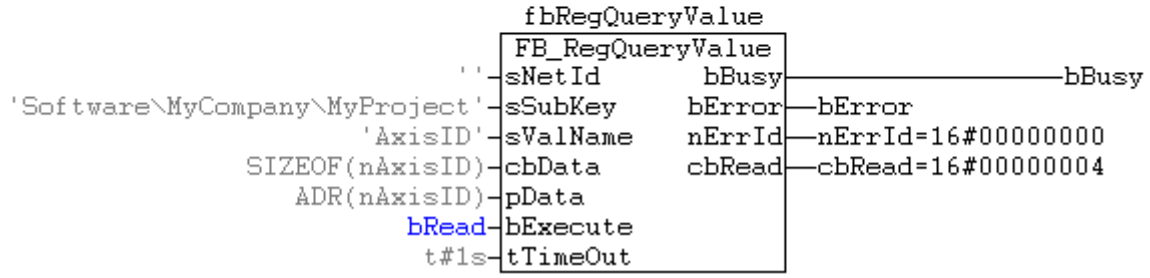
The values *AxisId* and *LogFilePath* are to be read from the system registry.




```
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:

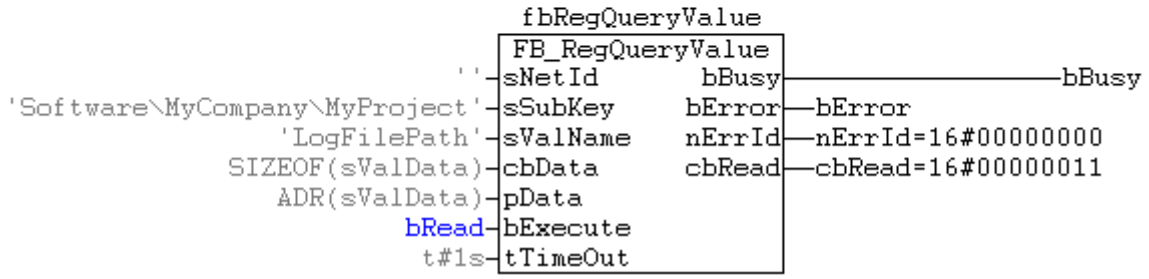
```
fbRegQueryValue
    bRead = TRUE
    bBusy = FALSE
    bError = FALSE
    nErrId = 16#00000000
    cbRead = 16#00000004
    nAxisID = 16#00000022
```



The value 0x22 of the registry was read in the PLC variable *nAxisId* .

Read REG_SZ-Value :

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

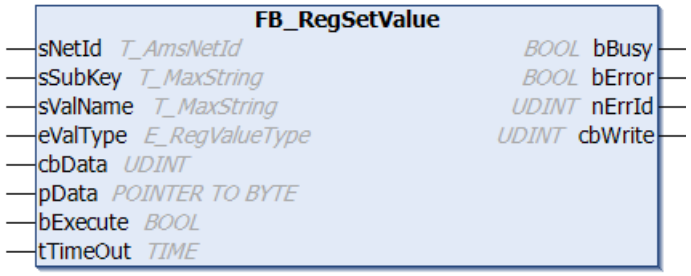


The string 'c:\MyProject\Log' of the registry was read in the PLC variable *sValData.*

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Utilities (System)

3.53 FB_RegSetValue



The system registry is a hierarchically structured tree. A node in the tree is referred to as a key. Each key can may contain subkeys and data values.

The function block "FB_RegSetValue" can be used to write or generate individual key values or new key names and values (subkeys+values) in the branch with the predefined handle **HKEY_LOCAL_MACHINE**. Any number of value types (e.g. REG_DWORD, REG_SZ) or a maximum of 500 bytes of binary data (REG_BINARY) can be written into the system registry. If a key value does not yet exist, it will automatically be created.

Comment:

The strings *sSubKey* and *sValueName* may not be empty!

HKEY_LOCAL_MACHINE\SOFTWARE\ for 64 bit operating systems

In a 64 bit Windows operating system all registry entries of and for 32 bit applications are not stored under HKEY_LOCAL_MACHINE\SOFTWARE\ but under HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\.
The function blocks FB_RegQueryValue and FB_RegSetValue work automatically below the WOW6432Node folder like any 32 bit application when a registry entry below the SOFTWARE folder is selected. The redirection is performed automatically by the operating system.

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: Here you can enter a string with the network address of the TwinCAT computer whose system registry is to be written (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

sSubKey: String with the subkey name (type: T_MaxString).

sValName: String with the value name (type: T_MaxString).

eValType: The data type format of the registry data to be written, e.g: REG_DWORD or REG_SZ (type: E_RegValueType [▶ 300]).

cbData: The number of value data bytes that are to be written. (In the case of string variables this includes the final null).

pD ata: Address of a data buffer/PLC variable containing the value data. The address can be determined with the ADR operator. The programmer is responsible for dimensioning the data buffer such that *cbData* data bytes can be read from it.

bExecute: the block is activated by a positive edge at this input.

tTimeOut: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrId: Supplies the ADS error number [▶ 353] or the command-specific error code (table) when the *bError* output is set.

cbWrite: Number of successfully written value data bytes.

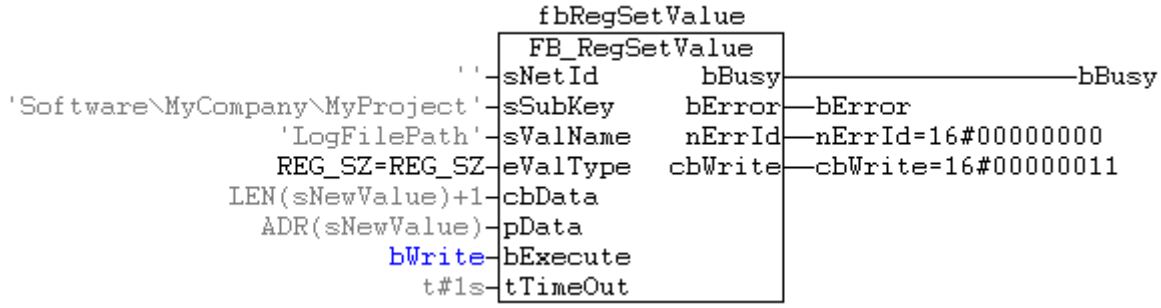
Error Codes	Error description
0x00	No error
0x01	The key with the name **sSubKey** could not be opened/found.
0x02	The key value with the name **sValName** could not be opened/found.

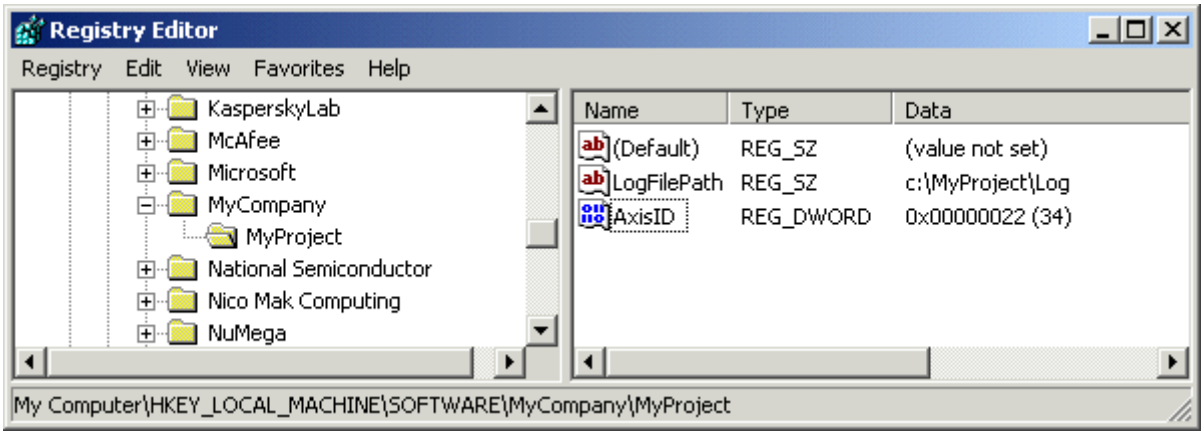
Example:

In the branch with the predefined handle *HKEY_LOCAL_MACHINE* a subkey *'SOFTWARE\MyCompany\MyProject'* with the key name '*LogFileName'* , the type *REG_SZ*

and the value '*c:\MyProject\Log"* is to be created and set.

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

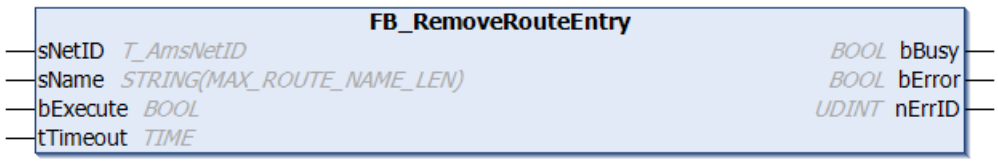




Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Utilities (System)

3.54 FB_RemoveRouteEntry



The function block can be used to remove an existing connection to a TwinCAT system from the list of AMS router connections (remote routes).

VAR_INPUT

```
VAR_INPUT
    sNetID   : T_AmsNetID;
    sName    : STRING(MAX_ROUTE_NAME_LEN);
    bExecute : BOOL;
    tTimeout : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

sNetID: Here you can specify a string with the network address of the TwinCAT computer on which the AMS router connection is to be removed (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

sName: Name of the connection to be removed. The maximum string length is limited by a constant (default: 31 characters).

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

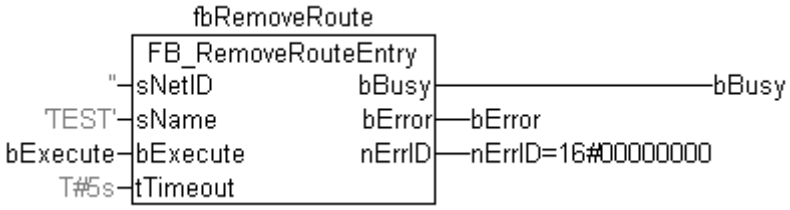
bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrId: Supplies the ADS error number [▶ 353] when the *bError* output is set.

Example:

The connection with the name "TEST" is to be removed from the list of AMS router connections on the local TwinCAT system. The connection is removed if a rising edge is detected at the bExecute variable.

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



Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.55 FB_ScopeServerControl

The function block "FB_ScopeServerControl" enables the PLC to collect data for subsequent display with TwinCAT Scope 2.

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: This parameter can be used to specify a string containing the network address of the TwinCAT target system (type: T_AmsNetID). For the local computer this string may be empty.

eReqState:: Requested scope server status (type: E_ScopeServerState [▶ 301]).

sConfigFile:: Full path with the name of the configuration file (e.g.: 'C:\TwinCAT\TwinCATScope2\First.sv2').

sSaveFile: Full path with the name of data file (e.g: 'C:\TwinCAT\TwinCATScope2\First.svd').

tTimeout:: Maximum permitted time for the internal ADS commands

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

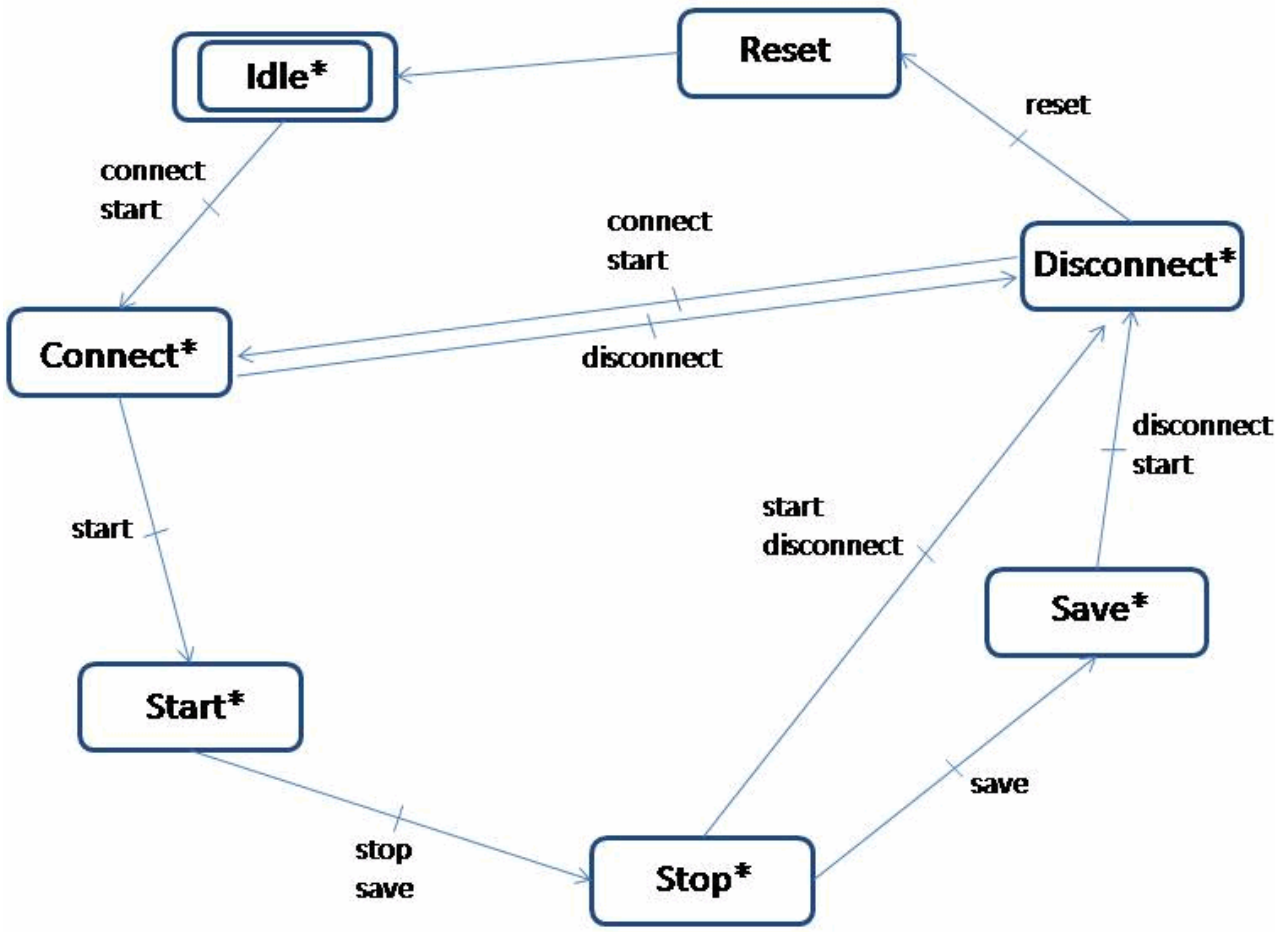
bDone: Is set if the requested status was activated.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrorId: Shows the error number if the error output bError is set. As a rule this can be an ADS error number [▶ 353] or a specific error code [▶ 352] of this library.

Only one target system is permitted for the configuration file (*.sv2).

State diagram:



*) to reset State Change is always possible

This state diagram shows the possible transitions for eReqState. If there is an other state change requested bError will be set.

TYPE E_ScopeServerState

```
TYPE E_ScopeServerState
(
   SCOPE_SERVER_IDLE,
   SCOPE_SERVER_CONNECT,
   SCOPE_SERVER_START,
   SCOPE_SERVER_STOP,
   SCOPE_SERVER_SAVE,
   SCOPE_SERVER_DISCONNECT,
   SCOPE_SERVER_RESET
);
```

Example:

Declaration part:

```
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
```

Implementation of the FB_ScopeServerSample

```
CASE nState OF
0:
    eReqState := SCOPE_SERVER_START;
    nState := 10;
10:
    IF fbScopeServerControl.bDone AND bExternalTriggerEvent
    THEN
        bTriggerTimer := TRUE;
        nState := 20;
    END_IF
20:
    IF fbTimer.Q THEN
        eReqState := SCOPE_SERVER_SAVE;
        bTriggerTimer := FALSE;
        nState := 30;
    END_IF
30:
    IF fbScopeServerControl.bDone THEN
        eReqState := SCOPE_SERVER_DISCONNECT;
    END_IF
END_CASE
fbTimer(IN:=bTriggerTimer, PT:=t#10s);
fbScopeServerControl( sNetId:= '',
                      eReqState:= eReqState,
                      sConfigFile:= 'C:\twinCat\scope\test.sv2',
                      sSaveFile:= 'C:\twinCat\scope\test.svd',
                      tTimeout:= t#5s,
                      bBusy=>bBusy,
                      bDone=>bDone,
                      bError=>bError,
                      nErrorId=>nErrorId);
```

This example demonstrates a long-term recording with the Scope Server.

The existing configuration (Test.sv2) is loaded for this purpose. In this example Test.sv2 was stored in order to run in the ring buffer. Data logging will not finish until FB_ScopeServerControl becomes active. If an internal trigger event (e.g. an error event) occurs, a timer is started, and 10 seconds later the data are saved in Test.svd. In this way the data file contains information from before and after the trigger event.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.56 FB_SetTimeZoneInformation

This function block can be used to modify or set the time zone settings of the operating system.

Time settings

The operating system may change some time settings after the new time zone settings were set.

The time can be reset with the function block: NT_SetLocalTime [▶ 122].

VAR_INPUT

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

sNetID: Here you can enter a string with the network address of the TwinCAT computer whose time zone configuration is to be changed (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

tzInfo: Structure with the new time zone settings that are to be set (type: ST_TimeZoneInformation [▶ 315]).

bExecute: the block is activated by a positive edge at this input.

tTimeout: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

bBusy: When the function block is activated this output is set. It remains set until a feedback is received.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

nErrId: Supplies the ADS error number [▶ 353] when the *bError* output is set.

Sample:

On the local TwinCAT system the time zone: "Western Europe Standard Time" is to be set. As an example a constant: **WEST_EUROPE_TZI** with suitable parameter values was already declared in the PLC library. To configure other time zones, the *tzInfo* input of the function block must be assigned corresponding values (see description of the ST_TimeZoneInformation [▶ 315] structure).

```
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
```

The declaration section:

```
PROGRAM MAIN
VAR
    fbGet    : FB_GetTimeZoneInformation;
    fbSet    : FB_SetTimeZoneInformation;
    tzi_get  : ST_TimeZoneInformation;
    tzID     : E_TimeZoneID;
    bGet     : BOOL := TRUE;
    bSet     : BOOL := FALSE;
END_VAR
```

A rising edge at the *bSet* variables results in setting of the required time zone setting. For verification the current settings can be read with a rising edge at the *bGet* variable.

```
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
```

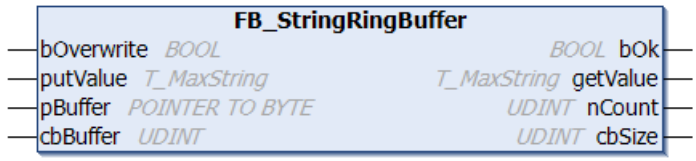
Further time and time zone functions and function blocks:

- FB_TzSpecificLocalTimeToSystemTime [▶ 115]
- FB_TzSpecificLocalTimeToFileTime [▶ 29]
- FB_SystemTimeToTzSpecificLocalTime [▶ 111]
- FB_FileTimeTimeToTzSpecificLocalTime [▶ 27]
- FB_GetTimeZoneInformation [▶ 77]
- NT_SetLocalTime [▶ 122]
- NT_GetTime [▶ 120]
- NT_SetTimeToRTCTime [▶ 123]
- F_TranslateFileTimeBias [▶ 238]
- FB_LocalSystemTime [▶ 89]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Utilities (System)

3.57 FB_StringRingBuffer



The function block FB_StringRingBuffer can be used to write string variables into the ring buffer or read previously written string variables from the ring buffer. The written strings are read in the same order as they were previously written to the ring buffer, based on the FIFO principle, i.e. the oldest entries are read first. The buffer memory is made available to the function block via the *pBuffer*/*cbBuffer* input variables. Writing/reading of strings is controlled via action calls. The function block features the following tasks:

- **A_AddTail** (writes a new string into the ring buffer.)
- **A_GetHead** (reads the oldest string in the ring buffer, but does not remove it.)
- **A_RemoveHead** (reads and removes the oldest string from the ring buffer.)
- **A_Reset** (deletes all strings from the ring buffer.)

VAR_INPUT

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

bOverwrite : If TRUE the oldest entries are overwritten in the event of a buffer overflow. If FALSE an error reported in the event of a buffer overflow (bOk = FALSE).

putValue: String to be written into the ring buffer (type: T_MaxString).

pBuffer: Address of a PLC variables (e.g. ARRAY[...] OF BYTES) to be used as buffer memory by the function block. The address can be determined with the ADR operator.

cbBuffer: Max. byte size of the PLC variable to be used as buffer memory. The size can be determined with the SIZEOF operator.

VAR_OUTPUT

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

bOk: Returns TRUE, if a new string was added or removed successfully. Returns FALSE in the event of a buffer overflow or if no more entries are available in the buffer.

getValue: This output supplies the last string that was read from the ring buffer (type: T_MaxString).

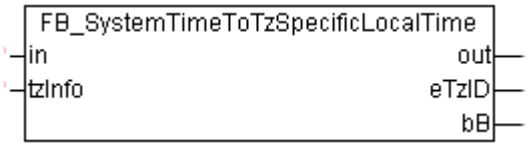
nCount: Returns the current number of queued strings.

cbSize: Returns the current number of assigned data bytes in the buffer. The number of assigned data bytes is always greater than the actual number of written value data. Each string is complemented with additional information, so that it can be located later.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.58 FB_SystemTimeToTzSpecificLocalTime



The function block converts the UTC time (structured system time format) to local time (structured system time format), taking into account the specified time zone information. The function block FB_FileTimeToTzSpecificLocalTime [▶ 27] has similar functionality but uses a different time format (file time format).

The block is only suitable for conversion of **continuous** UTC timestamp information. The function block uses the time zone information to calculate the required time steps (summer/winter time changeover) in local time. Time steps in UTC input time are not permitted and lead to incorrect conversion. The reason: the block stores the last converted time internally, so that it can detect the B times (see below) from the UTC input time and the stored value when the local time is changed.

The block is associated with an action: A_Reset(). If this action is called the block outputs and the locally stored (last converted) time are reset to zero.

VAR_INPUT

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

in: UTC time (structured system time format) to be converted (type: TIMESTRUCT [▶ 321]).

tzInfo: Structure variable with the current time zone information of the operating system (type: ST_TimeZoneInformation [▶ 315]).

VAR_OUTPUT

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

out: Converted local time (structured system time format) (type: TIMESTRUCT [▶ 321]).

eTzID: Additional summer/winter time information (type: E_TimeZoneID [▶ 302]).

bB: TRUE => B time (e.g.:**02:05:00 CET B**), FALSE => other time (e.g.: **02:05:00 CEST A**). This output is set if the local time jumps back and is reset once the duplicate local time has passed.

Example:

```
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, wHou
r := 16, wMinute := 46, wSecond := 31, wMilliseconds := 99 ) *)
    fbToLocal : FB_SystemTimeToTzSpecificLocalTime;
END_VAR
```

```
fbToLocal( in := in, tzInfo := WEST_EUROPE_TZI, out => out );
```

Further time and time zone functions and function blocks:

- FB_TzSpecificLocalTimeToSystemTime [▶ 115]
- FB_TzSpecificLocalTimeToFileTime [▶ 29]
- FB_FileTimeTimeToTzSpecificLocalTime [▶ 27]
- FB_GetTimeZoneInformation [▶ 77]
- FB_SetTimeZoneInformation [▶ 108]
- NT_SetLocalTime [▶ 122]
- NT_GetTime [▶ 120]
- NT_SetTimeToRTCTime [▶ 123]
- F_TranslateFileTimeBias [▶ 238]
- FB_LocalSystemTime [▶ 89]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.59 FB_TzSpecificLocalTimeToFileTime64

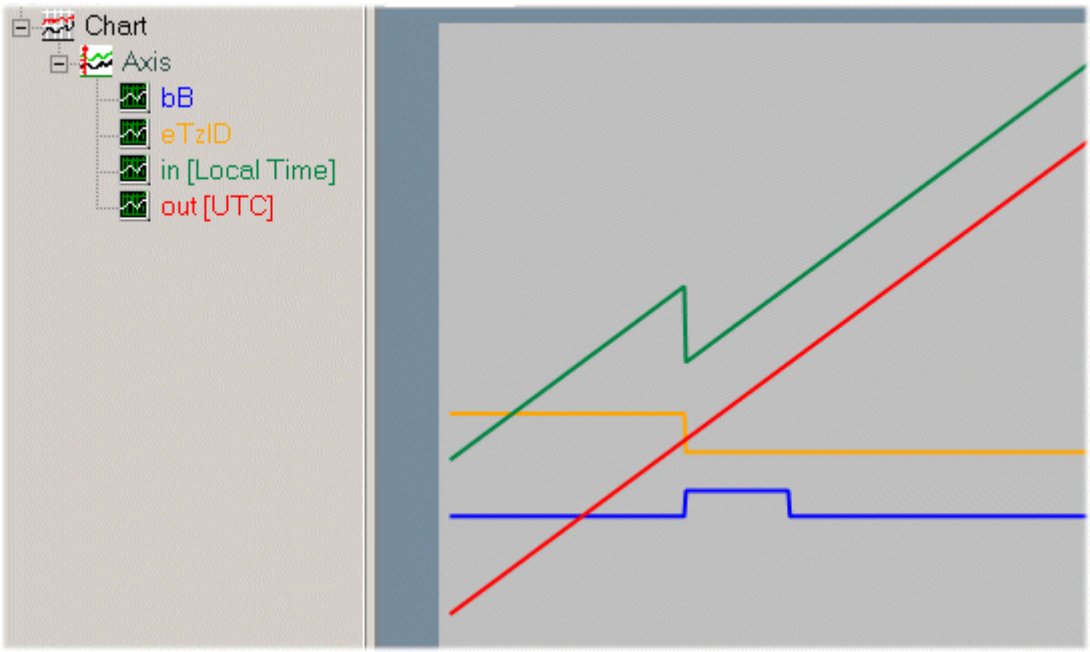
The function block converts the local time (file time format) to UTC time (file time format), taking into account the specified time zone information. The function block: FB_TzSpecificLocalTimeToSystemTime [▶ 115] has a similar function, the difference being that it converts to a different time format (structured system time format).

The function block is only suitable for conversion of **continuous** local timestamp information. Step changes in local time caused by summer/winter time changeover are permitted and are correctly detected by the function block. Arbitrary changes in local time result in incorrect conversion. The reason: the last converted time is stored internally in function block in order to be able to identify the summer time/winter time

information and the B times (see below) when the local time is reset. The function block is associated with an action: A_Reset(). If this action is called the function block outputs and the locally stored (last converted) time are reset to zero.

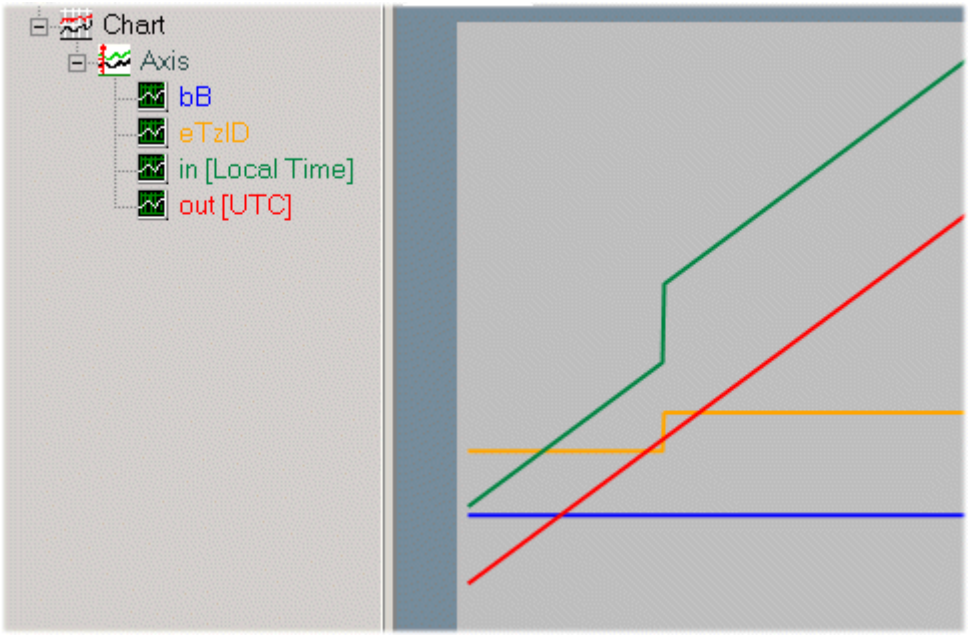
The step changes in the local time are problematic, since they have to be converted to a linear UTC time. It is therefore advisable to use the (continuous) UTC time for time stamping tasks and to convert the time to respective local time only for display purposes (e.g. in a visualization).

1. Graphic representation of the changeover from summer time to winter time (tzInfo = WEST_EUROPE_TZI):



The local time (green) jumps back. The UTC output time (red) continuous. The local time: **02h:59m:59s:999ms..** is directly followed by: **02h:00m:00s:000ms..** The times between 2h and 3h occur twice. The duplicate time before the changeover is referred to as **02:05:00 CEST A**, for example, the time after the changeover as **02:05:00 CET B.** The output variable *bB* indicates whether it is the first or the second *pass*. During the second *pass* the *bB* output variable (blue) is set to TRUE. The *bB* output variable is automatically reset once the duplicate time has passed. The time zone ID (orange) changes from *eTimeZoneID_Daylight* (summer time) to *eTimeZoneID_Standard* (winter time).

2. Graphic representation of the change-over from winter time to summer time (tzInfo = WEST_EUROPE_TZI):



The local time (green) jumps forward. The UTC output time (red) continuous. The local time: **2h:59m:59s:999ms..** is directly followed by: **3h:00m:00s:000ms..** The time zone ID (orange) changes from *eTimeZoneID_Standard* (winter time) to *eTimeZoneID_Daylight* (summer time).

VAR_INPUT

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

in: Local time (file time format) that is to be converted (type: T_FILETIME [▶ 317]).

tzInfo: Structure variable with the current time zone information of the operating system (type: ST_TimeZoneInformation [▶ 315]).

VAR_OUTPUT

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

out: Converted UTC time (file time format) (type: T_FILETIME64 [▶ 317]).

eTzID: Additional summer/winter time information (type: E_TimeZoneID [▶ 302]).

bB: TRUE => B-time (e.g.:**02:05:00 CET B**), FALSE => remaining time (e.g.: **02:05:00 CEST A**). This output is set if the local time jumps back and is reset once the duplicate local time has passed.

Sample:

The local time: DT#2011-09-02-11:01:31 is converted into UTC time: DT#2011-09-02-09:01:31.

```
PROGRAM MAIN
VAR
    in      : DT := DT#2011-09-02-11:01:31;(* Local time *)
    out     : DT;(* UTC time *)
    fbToUTC : FB_TzSpecificLocalTimeToFileTime64;
END_VAR
```

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

Further time and time zone functions and function blocks:

- FB_TzSpecificLocalTimeToSystemTime [▶ 115]

- FB_SystemTimeToTzSpecificLocalTime [▶ 111]
- FB_FileTime64ToTzSpecificLocalTime [▶ 61]
- FB_GetTimeZoneInformation [▶ 77]
- FB_SetTimeZoneInformation [▶ 108]
- NT_SetLocalTime [▶ 122]
- NT_GetTime [▶ 120]
- NT_SetTimeToRTCTime [▶ 123]
- F_TranslateFileTime64Bias [▶ 150]
- FB_LocalSystemTime [▶ 89]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.44.0

3.60 FB_TzSpecificLocalTimeToSystemTime

The function block converts the local time (structured system time format) to UTC time (structured system time format), taking into account the specified time zone information. The function block FB_TzSpecificLocalTimeToFileTime [▶ 29] has similar functionality but uses a different time format (file time format).

The block is only suitable for conversion of **continuous** local timestamp information. Step changes in local time caused by summer/winter time changeover are permitted and are correctly detected by the block. Arbitrary changes in local time result in incorrect conversion. The reason: the last converted time is stored internally in block in order to be able to identify the summer time/winter time information and the B times (see below) when the local time is reset. The block is associated with an action: A_Reset(). If this action is called the block outputs and the locally stored (last converted) time are reset to zero.

The step changes in the local time are problematic, since they have to be converted to a linear UTC time. It is therefore advisable to use the (continuous) UTC time for time stamping tasks and to convert the time to respective local time only for display purposes (e.g. in a visualization).

Further information can be found in the documentation for the FB_TzSpecificLocalTimeToFileTime [▶ 29] function block.

VAR_INPUT

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

in: Local time (structured system time format) to be converted (type: TIMESTRUCT [▶ 321]).

tzInfo: Structure variable with the current time zone information of the operating system (type: ST_TimeZoneInformation [▶ 315]).

VAR_OUTPUT

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

out: Converted UTC time (structured system time format) (type: TIMESTRUCT [▶ 321]).

eTzID: Additional summer/winter time information (type: E_TimeZoneID [▶ 302]).

bB: TRUE => B time (e.g.:**02:05:00 CET B**), FALSE => other time (e.g.: **02:05:00 CEST A**). This output is set if the local time jumps back and is reset once the duplicate local time has passed.

Example:

```
PROGRAM MAIN
VAR
    in      : TIMESTRUCT := ( wYear := 2011, wMonth := 4, wDay := 29, wHour := 16, wMinute := 46, wS
econd := 31, wMilliseconds := 99 );(* Local time *)
    out     : TIMESTRUCT;
(* UTC time result is:= ( wYear := 2011, wMonth := 4, wDay := 29, wHour := 14, wMinute := 46, wSecon
d := 31, wMilliseconds := 99 ) *)
    fbToUTC : FB_TzSpecificLocalTimeToSystemTime;
END_VAR
```

```
fbToUTC( in := in, tzInfo := WEST_EUROPE_TZI, out => out );
```

Further functions and function blocks for time and time zone:

- FB_TzSpecificLocalTimeToFileTime [▶ 29]
- FB_SystemTimeToTzSpecificLocalTime [▶ 111]
- FB_FileTimeToTzSpecificLocalTime [▶ 27]
- FB_GetTimeZoneInformation [▶ 77]
- FB_SetTimeZoneInformation [▶ 108]
- NT_SetLocalTime [▶ 122]
- NT_GetTime [▶ 120]
- NT_SetTimeToRTCTime [▶ 123]
- F_TranslateFileTimeBias [▶ 238]
- FB_LocalSystemTime [▶ 89]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.61 FB_WritePersistentData

The function block FB_WritePersistentData is an extended version of the WritePersistentData [▶ 142] function block. The system behavior for writing the persistent data [▶ 349] (data consistency/task cycle time overrun) can be influenced via the MODE parameter.

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    PORT    : UINT;
    START   : BOOL;
```

```
    TMOUT    : TIME := DEFAULT_ADS_TIMEOUT;
    MODE     : E_PersistentMode;
END_VAR
```

NETID: This parameter can be used to specify the AmsNetID of the TwinCAT computer whose persistent data are to be saved (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

PORT: The PORT parameter specifies the runtime system whose persistent data is to be stored.

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

MODE : Mode in which the persistent data are to be written (type: E_PersistentMode [▶ 300]).

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Example:

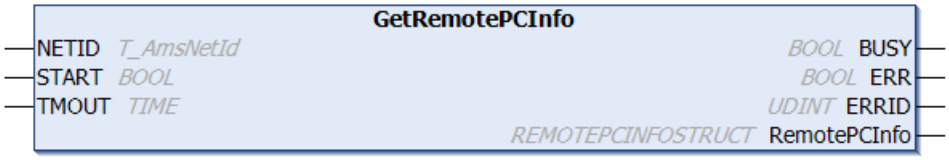
See: Writing of persistent data: system behavior. [▶ 349]

See: Example in the documentation for the WritePersistentData function block. [▶ 142]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.62 GetRemotePCInfo



The function block "GetRemotePCInfo" can be used to read information about configured remote PCs in the TwinCAT router. After successful execution the structure "RemotePCInfo" contains the NetIDs and names of the remote PCs as strings in the order in which they were stored in the TwinCAT router. The function block allows router information relating to either a local or to a remote TwinCAT system to be read.

VAR_INPUT

```
VAR_INPUT
    NETID    : T_AmsNetId;
    START    : BOOL;
    TMOUT    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: This parameter can be used to specify the AmsNetID of the TwinCAT computer whose router information is to be read via configured remote PCs (type: T_AmsNetID). To determine the remote PCs of the local TwinCAT system, an empty string can be specified.

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY          : BOOL;
    ERR           : BOOL;
    ERRID         : UDINT;
    RemotePCInfo  : REMOTEPCINFOSTRUCT;
END_VAR
```

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

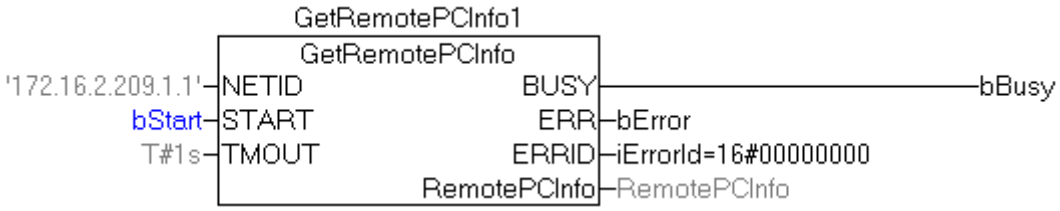
ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

RemotePCInfoStruct: Structure containing information about the configured remote PCs (type: REMOTEPCINFOSTRUCT [▶ 305]).

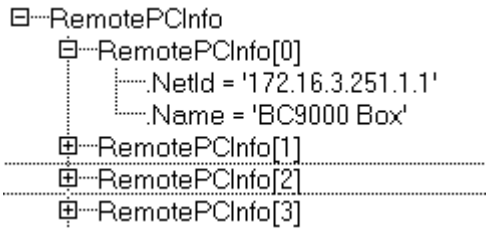
Example:

```
PROGRAM MAIN
VAR
    GetRemotePCInfo1     : GetRemotePCInfo;
    RemotePCInfo         : REMOTEPCINFOSTRUCT;
    bBusy                : BOOL;
    bError               : BOOL;
    iErrorId             : UDINT;
    bStart               : BOOL;
END_VAR
```



Online View:

NetIDs and names of the configured remote PCs.



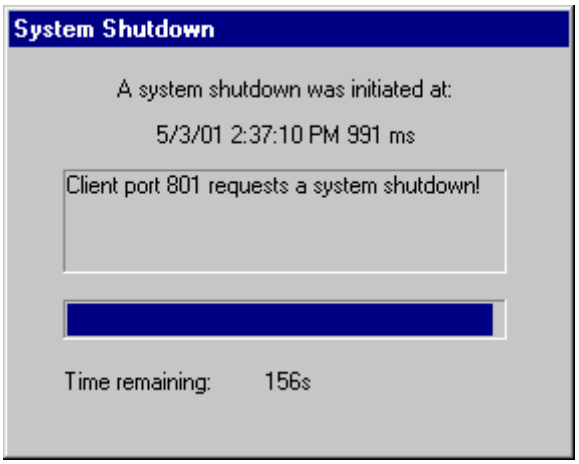
Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.63 NT_AbortShutdown

The "NT_AbortShutdown" function block can be used to abort a shutdown command that has previously been called with the NT_Shutdown [▶ 124] function block.

When calling the NT_Shutdown [▶ 124] function block, a delay period can be given as a parameter. The remaining time is indicated in a message window.



Only after the delay time has elapsed is the operating system shut down. During this time, the shutdown process can be interrupted from the PLC with the aid of the "NT_AbortShutdown" function block.

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    START   : BOOL;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: It is possible here to provide the AmsNetId of the TwinCAT computer on which the shutdown process is to be aborted (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

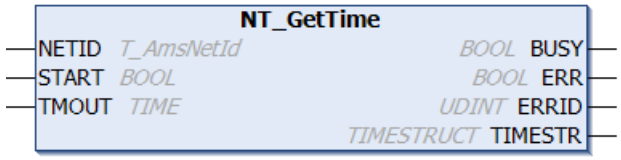
ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Utilities (System)

3.64 NT_GetTime



The function block "NT_GetTime" can be used to determine the local Windows system time of a TwinCAT system (the local Windows system time is displayed in the taskbar). The year, month, day, day of the week, hour, minute, second and millisecond are stored in variables of the structure TIMESTRUCT [▶ 321].

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    START   : BOOL;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: This parameter can be used to specify the AmsNetID of the TwinCAT computer whose local Windows system time is to be determined (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY    : BOOL;
    ERR     : BOOL;
    ERRID   : UDINT;
    TIMESTR : TIMESTRUCT;
END_VAR
```

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

TIMESTR: Structure with the local Windows system time (type: TIMESTRUCT [▶ 321]).

Example:

See: Sample: Software clocks (RTC, RTC_EX, RTC_EX2) [▶ 347].

Further time and time zone functions and function blocks:

- FB_TzSpecificLocalTimeToSystemTime [▶ 115]
- FB_TzSpecificLocalTimeToFileTime [▶ 29]
- FB_SystemTimeToTzSpecificLocalTime [▶ 111]
- FB_FileTimeTimeToTzSpecificLocalTime [▶ 27]

- FB_GetTimeZoneInformation [▶ 77]
- FB_SetTimeZoneInformation [▶ 108]
- NT_SetLocalTime [▶ 122]
- NT_SetTimeToRTCTime [▶ 123]
- F_TranslateFileTimeBias [▶ 238]
- FB_LocalSystemTime [▶ 89]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64, TC OS ARMT2)	Tc2_Utilities (System)

3.65 NT_Reboot

The Windows NT operating system can be restarted with the aid of the "NT_Reboot" function block. The function largely corresponds to the Restart command on the Windows taskbar. A delay before execution of the Restart command can be defined via the DELAY parameter.

VAR_INPUT

```
VAR_INPUT
    NETID  : T_AmsNetId;
    DELAY  : DWORD;
    START  : BOOL;
    TMOUT  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: It is possible here to provide the AmsNetId of the TwinCAT computer that is to be re-booted (type: T_AmsNetID). If the restart is to take place on the local computer, an empty string can be entered.

DELAY: The delay time, in seconds, before the Restart command is executed.

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Utilities (System)

3.66 NT_SetLocalTime

The function block "NT_SetLocalTime" can be used to set the local Windows system time of a TwinCAT system (the local Windows system time is displayed in the taskbar).

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    START   : BOOL;
    TIMESTR : TIMESTRUCT;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: This parameter can be used to specify the AmsNetID of the TwinCAT computer whose local Windows system time is to be set (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

START: the block is activated by a positive edge at this input.

TIMESTR: Structure with the new local Windows system time (type: TIMESTRUCT [▶ 321]).

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Further functions and function blocks for time and time zone:

- FB_TzSpecificLocalTimeToSystemTime [▶ 115]
- FB_TzSpecificLocalTimeToFileTime [▶ 29]
- FB_SystemTimeToTzSpecificLocalTime [▶ 111]
- FB_FileTimeToTzSpecificLocalTime [▶ 27]
- FB_GetTimeZoneInformation [▶ 77]
- FB_SetTimeZoneInformation [▶ 108]
- NT_GetTime [▶ 120]

- NT_SetTimeToRTCTime [▶ 123]
- F_TranslateFileTimeBias [▶ 238]
- FB_LocalSystemTime [▶ 89]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Utilities (System)

3.67 NT_SetTimeToRTCTime

This functionality is not available in the PLC runtime system under Windows CE!

The function block "Nt_SetTimeToRtcTime" can be used to synchronize the local Windows system time (displayed in the taskbar) with the real-time clock of the PC (RTC time in the BIOS).

Comments

When the function block is called, the real-time clock of the TwinCAT PC is compared with the local Windows system time, and the local Windows system time is corrected by the difference. Time zones and summer time are taken into account. Please note that the correction can lead to time jumps during measurements or log book entries.

When the local Windows system time is set, the operating system automatically sets the RTC time to the new local Windows system time. Due to the conversion and delay the new RTC time is inevitably subject to a small error. The error is in the millisecond range. This means that the real-time clock is falsified a little each time NT_SetTimeToRTCTime is called. In order to minimize deviations over a prolonged period, the compensation should be carried out every 24 hours, for example, rather than during each PLC cycle.

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    SET     : BOOL;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: This parameter can be used to specify the AmsNetID of the TwinCAT computer on which the local Windows system time is to be synchronized (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

SET: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Further functions and function blocks for time and time zone:

- FB_TzSpecificLocalTimeToSystemTime [▶ 115]
- FB_TzSpecificLocalTimeToFileTime [▶ 29]
- FB_SystemTimeToTzSpecificLocalTime [▶ 111]
- FB_FileTimeToTzSpecificLocalTime [▶ 27]
- FB_GetTimeZoneInformation [▶ 77]
- FB_SetTimeZoneInformation [▶ 108]
- NT_SetLocalTime [▶ 122]
- NT_GetTime [▶ 120]
- F_TranslateFileTimeBias [▶ 238]
- FB_LocalSystemTime [▶ 89]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64)	Tc2_Utilities (System)

3.68 NT_Shutdown

This functionality is not available under Windows CE!

The Windows NT operating system can be shut down with the aid of the "NT_Shutdown" function block. The function largely corresponds to the Shut Down command on the Windows taskbar. A delay before execution of the Shut Down command can be defined via the DELAY parameter.

Notes:

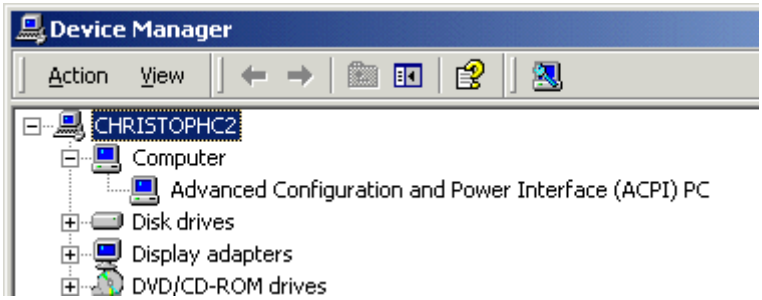
Newer operating systems (e.g. Windows 2000) perform with the aid of the "NT_Shutdown" function block the "Shutdown with Power OFF" (the computer switches its power OFF). This function can only be used on systems which are ACPI conform (Advanced Configuration and Power Interface). The ACPI functions should be activated in BIOS before the installation of the operating system.

The ACPI-functions have to be supported by the motherboard and the power supply of the PC. A change afterwards is not recognized by the operating system. If there is an ACPI-supporting PC, you can check e. g. at Windows 2000 in the following way:

1. In the "System Manager" open the folder "system".

2. on the tab "Hardware" choose the "Device Manager".

In the navigation tree with the devices now you can read at "Computer": "Advanced Configuration and Power Interface (ACPI) PC".



The default TwinCAT settings are to perform shutdown with power OFF. You can disable the power OFF function in windows registry. Please add following entry:

"**DisableACPIPowerOff**"**REG_DWORD = 0x00000001** in Registry under:
"**HKEY_LOCAL_MACHINE\SOFTWARE\Beckhoff\TwinCAT\System**"

VAR_INPUT

```
VAR_INPUT
    NETID  : T_AmsNetId;
    DELAY  : DWORD;
    START  : BOOL;
    TMOUT  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: It is possible here to provide the AmsNetId of the TwinCAT computer on which the operating system is to be shut down (type: T_AmsNetID). If it is the local computer that is to be shut down, an empty string can be entered.

DELAY: The delay time, in seconds, before the Shut Down command is executed.

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

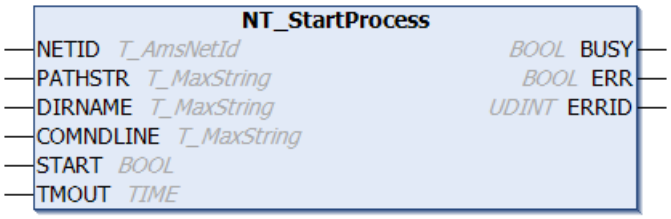
ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7)	Tc2_Utilities (System)

3.69 NT_StartProcess



The "NT_StartProcess" function block can be used to start a Windows application from the PLC. The function block can also be used to run applications on a remote PC.

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: It is possible here to provide the AmsNetId of the TwinCAT computer on which the application is to be started (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

PATHSTR: Whole path of the application to be executed as a string (type: T_MaxString, e.g. "C:\WINNT\NOTEPAD.EXE").

DIRNAME: Working directory of the application to be executed as a string (type: T_MaxString, e.g. "C:\WINNT").

COMNDLINE: Command line parameter (type: T_MaxString, e.g.: "win.ini").

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

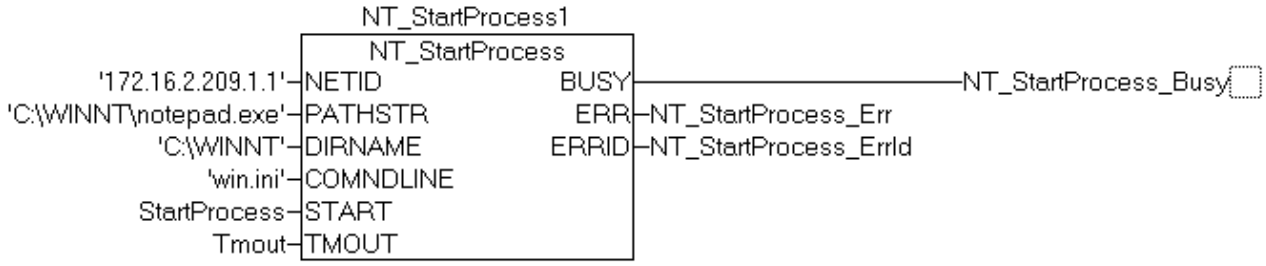
BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] or the Win32 error code [▶ 356] when the ERR output is set (Platform SDK: Win32 API).

Example:

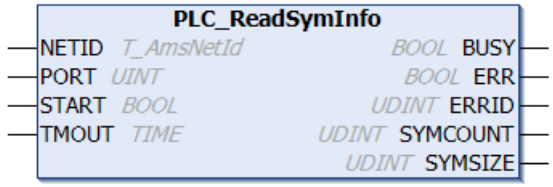
```
NT_StartProcess1      : NT_StartProcess;
NT_StartProcess_Busy  : BOOL;
NT_StartProcess_Err   : BOOL;
NT_StartProcess_ErrId : UDINT;
StartProcess          : BOOL;
Tmout                 : TIME;
```



Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (WES7/Win7/Win10: TC RT x86/x64, WEC6/7: TC RT x86, WEC7: TC CE7 ARMV7, TC/BSD: TC RT x64)	Tc2_Utilities (System)

3.70 PLC_ReadSymInfo



The "PLC_ReadSymInfo" function block permits information regarding the symbols (variables) of a PLC run-time system.

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    PORT    : T_AmsPort;
    START   : BOOL;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: It is possible here to provide the AmsNetId of the TwinCAT computer whose symbol information is to be found (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

PORT: The port number of a PLC run-time system (type: T_AmsPort).

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY     : BOOL;
    ERR      : BOOL;
    ERRID    : UDINT;
    SYMCOUNT : UDINT;
    SYMSIZE  : UDINT;
END_VAR
```

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

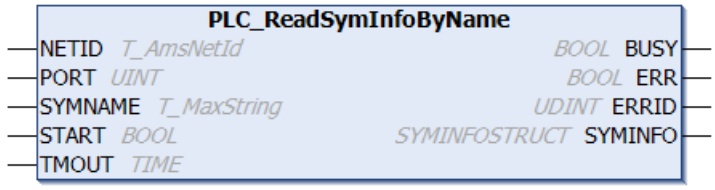
SYMCOUNT: The number of symbols in the PLC run-time system.

SYMSIZE: Length of the data, in bytes, in which the symbol information is stored.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.71 PLC_ReadSymInfoByName



The function block "PLC_ReadSymInfoByName" allows additional information about a PLC symbolic variables (e.g. data type identification, index group, index offset, comment...) to be read using the symbol name. After successful execution, the data is available in the **SymInfo** data structure, whose type is SYMINFOSTRUCT. The maximum length of the symbol name that is transferred to the function block as a parameter is limited to 255 characters.

VAR_INPUT

```
VAR_INPUT
    NETID    : T_AmsNetId;
    PORT     : T_AmsPort;
    SYMNAME  : T_MaxString;
    START    : BOOL;
    TMOUT    : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: The AmsNetId of the TwinCAT computer on which the function is to be executed can be entered here (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

PORT: The port number of the PLC run-time system to which the symbolic variable belongs (type: T_AmsPort).

SYMNAME: The symbol name of the PLC variable whose information is to be read (type: T_MaxString, max. 255 characters, including the whole path, e.g. 'MAIN.INIT_TASK.VARINT').

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY     : BOOL;
    ERR      : BOOL;
    ERRID    : UDINT;
    SymInfo  : SYMINFOSTRUCT;
END_VAR
```

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

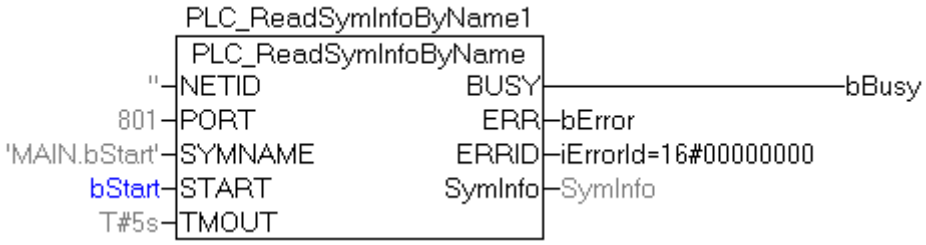
ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

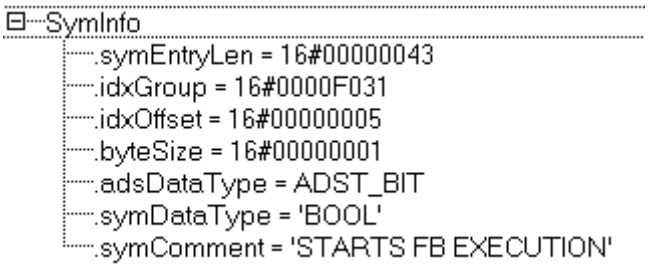
SymInfo: A structure with additional information on the symbolic variable (type: SYMINFOSTRUCT [▶ 316]).

Example:

```
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
```



Online View:



The data obtained in this way has the following meanings:

symEntryLen = 16#43: The actual length of the entry in the symbol table is 67 bytes;

idxGroup = 16#F031: It is a variable from the PLC process image of the physical outputs;

idxOffset = 16#5: The variable is located at byte offset zero and bit offset 5;

byteSize = 16#1: The variable's value occupies one byte in memory;

adsDataType = ADST_BIT: The ADS data type ID;

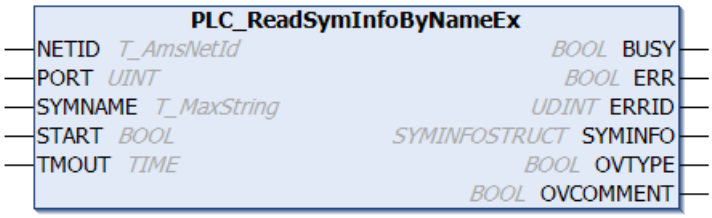
symDataType = BOOL: The data type identification in the PLC;

symComment = 'STARTS FB EXECUTION': Comment that the user has added to the variable definition line.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.72 PLC_ReadSymInfoByNameEx



The function block "PLC_ReadSymInfoByNameEx" has similar functionality as the function block PLC_ReadSymInfoByName [▶ 128]. Both function blocks can read symbol information through the symbol name. The difference between the two blocks is that the block described here does not report an error if the available buffer size is exceeded and may output incomplete information. In this case the comment and/or the data type ID may have been truncated. Two additional output variables indicate this, i.e. *OVTYPE* and *OVCOMMENT*, so that the application can respond accordingly.

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    PORT    : T_AmsPort;
    SYMNAME : T_MaxString;
    START   : BOOL;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: The AmsNetId of the TwinCAT computer on which the function is to be executed can be entered here (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

PORT: The port number of the PLC run-time system to which the symbolic variable belongs (type: T_AmsPort).

SYMNAME: The symbol name of the PLC variable whose information is to be read (type: T_MaxString, max. 255 characters, including the whole path, e.g. 'MAIN.INIT_TASK.VARINT').

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY      : BOOL;
    ERR       : BOOL;
    ERRID     : UDINT;
    SymInfo   : SYMINFOSTRUCT;
    OVTYPE    : BOOL;
    OVCOMMENT : BOOL;
END_VAR
```

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

SymInfo: A structure with additional information on the symbolic variable (type: SYMINFOSTRUCT [▶ 316]).

OVTYPE: Indicates whether the string with the data type identification has caused an overflow (TRUE). The string with the data type identification may have been truncated.

OVCOMMENT: Indicates whether the string with the symbol comment has caused an overflow (TRUE). The string with the comment may have been truncated.

Example:

See function block documentation: PLC_ReadSymInfoByName [▶ 128]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.73 PLC_Reset

The "PLC_Reset" function block can be used to reset a PLC run-time system. When the PLC is reset, the PLC variables are filled with their initial values, and the execution of the PLC program is stopped.

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    PORT    : T_AmsPort;
    RESET   : BOOL;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: It is possible here to provide the AmsNetId of the TwinCAT computer on which the PLC run-time system is to be reset (type: T_AmsNetID). If the PLC reset is to be carried out on the local computer, an empty string can be entered.

PORT: Contains the ADS port number of the PLC run-time system that is to be reset (type: T_AmsPort).

RESET: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

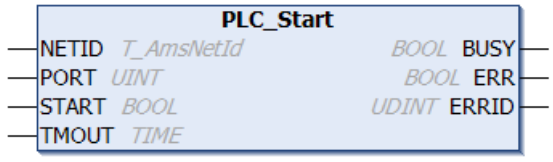
ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.74 PLC_Start



The function block "PLC_Start" can be used to start a PLC run-time system on a TwinCAT computer. The function block can, for instance, be used to start the PLC on a remote PC.

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    PORT    : T_AmsPort;
    START   : BOOL;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: This parameter can be used to specify the AmsNetID of the TwinCAT computer on which the PLC is to be started (type: T_AmsNetID). If the PLC is to be started on the local computer, an empty string can be entered.

PORT: Contains the ADS port number of the PLC run-time system that is to be started (type: T_AmsPort).

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.75 PLC_Stop

The function block "PLC_Stop" can be used to stop a PLC run-time system on a TwinCAT computer. The function block can, for instance, be used to stop the PLC on a remote or a local PC.

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    PORT    : T_AmsPort;
    STOP    : BOOL;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: This parameter can be used to specify the AmsNetID of the TwinCAT computer on which the PLC is to be stopped (type: T_AmsNetID). If the PLC to be stopped is on the local computer, an empty string can be entered.

PORT: Contains the ADS port number of the PLC run-time system that is to be stopped (type: T_AmsPort).

STOP: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

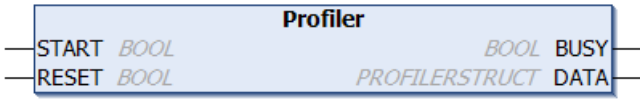
ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.76 Profiler

This functionality is not available in the PLC under Windows CE!



The "Profiler" function block can be used to allow the execution time of PLC code to be measured. Internally, an instance of the GETCPUACCOUNT function block is called. The measurement is started by a rising edge at the START input, and is stopped by a falling edge. The measurements are evaluated internally, and are then made available for further processing at the DATA output in a structure of type PROFILERSTRUCT [▶ 304]. As well as the current, minimum and maximum execution times, the function block calculates the mean execution time for the last 10 measurements. The number of averaged measured values can be configured via the global variable MAX_AVERAGE_MEASURES [▶ 323] between 2 and 100. The times measured are given in microseconds. The output variable DATA.MeasureCycle [▶ 304] provides information about the number of measurements that have already been carried out. In order to measure the execution time for a specific segment of the PLC program the measurement must be started by a rising edge at the START input when the segment to be measured starts, and stopped by a falling edge at the START input at the end of the segment. All values at the DATA output can be reset if a rising edge is generated at the RESET input at the same time as the rising edge at START. The old measured values are then reset when a new measurement starts and are recalculated from the subsequent calls of the function block.

Comment:

The times measured can differ from the actual values, since a certain amount of time is needed just for the call of the GETCPUACCOUNT function block. This time depends on the particular computer, and is included in the times that are found.

VAR_INPUT

```
VAR_INPUT
    START   : BOOL;
    RESET   : BOOL;
END_VAR
```

START: A rising edge at this input starts the measurement of the execution time. A falling edge at this input stops the measurement, and causes the current, minimum, maximum and mean execution times to be recalculated. The variable DATA.MeasureCycle [▶ 304] is incremented at the same time.

RESET: All variables at the DATA output are reset if a rising edge is generated at this input at the same time as a rising edge at the START input. The old values for the current, minimum, maximum and mean execution times are reset, and are re-calculated for following measurements.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY  : BOOL;
    DATA  : PROFILERSTRUCT;
END_VAR
```

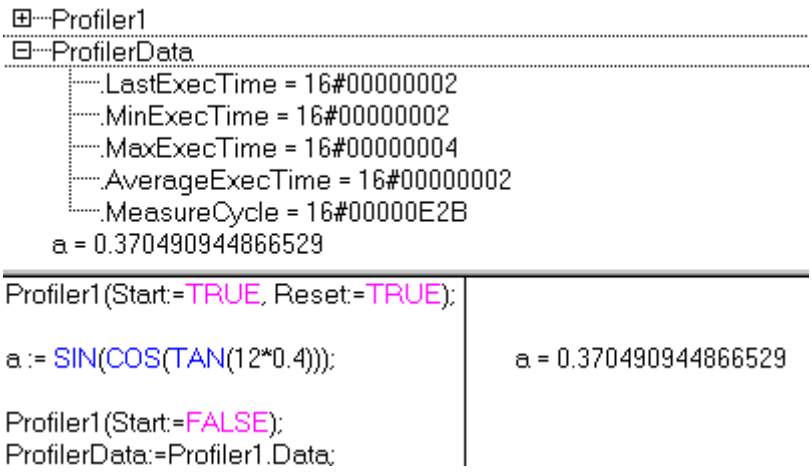
BUSY: This input is set at the start of the measuring procedure, and remains set until the time measurement has been completed. Once the BUSY output has been reset, the latest times are available at the DATA output.

DATA: structure of type PROFILERSTRUCT [▶ 304] with the measured times [in µs].

Example 1:

```
PROGRAM ProfilerTest_ST
VAR
   Profiler1     : PROFILER;
   ProfilerData  : PROFILERSTRUCT;
   a             : LREAL;
END_VAR
```

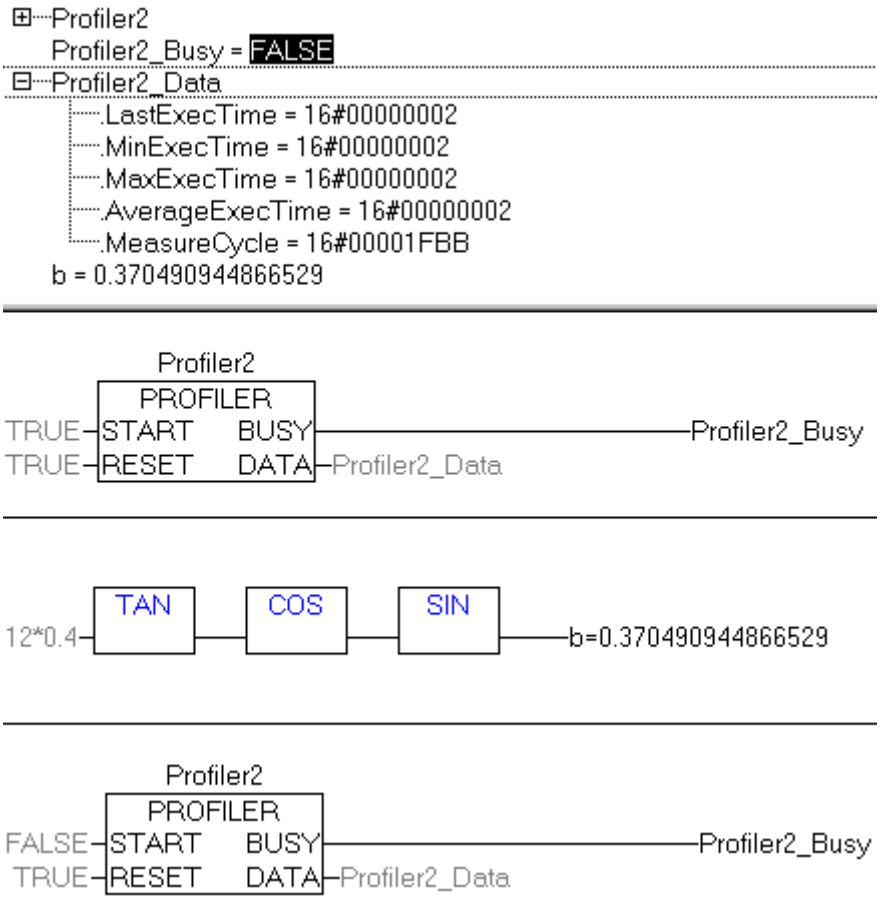
Online display of the measured times:



Example 2:

```
PROGRAM ProfilerTest_FUP
VAR
   Profiler2      :PROFILER;
   Profiler2_Busy :BOOL;
   Profiler2_Data :PROFILERSTRUCT;
   b              :LREAL;
END_VAR
```

Online display of the measured times:



Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64)	Tc2_Utilities (System)

3.77 RTC

The function block "RTC" (Real Time Clock) can be used to realize an internal software clock in the TwinCAT PLC. The clock must be initialized with a starting date and time. After the initialization the time and date are updated with each call of the function block. A CPU system clock is used to calculate the current time and date. The function block should be called in every PLC cycle, so that the current time can be calculated. The current date and time are available in the usual DATE_AND_TIME (DT) format at the function block's output. Multiple instances of the RTC function block can be created within one PLC program.

RTC time can differ from the reference time

The way the system works means that the RTC time can differ from the reference time. The difference depends on the PLC's cycle time, the value of the basic system ticks, and on the hardware being used.

In order to avoid larger deviations the RTC instance should be synchronized cyclically (e.g. with a radio clock or with the local Windows system time). The local Windows system time you can be synchronized with a reference time via the SNTP protocol.

VAR_INPUT

```
VAR_INPUT
   EN   : BOOL;
   PDT : DATE_AND_TIME;
END_VAR
```

EN: The function block is re-initialized with a specified date and time by a rising edge at this input.

PDT: (Preset Date and Time) The initialisation values for the function block's date and time. A rising edge at the EN input will cause the function block to adopt this value.

VAR_OUTPUT

```
VAR_OUTPUT
   Q     : BOOL;
   CDT   : DATE_AND_TIME;
END_VAR
```

Q: This output is set if the function block has been initialized at least once. If the output is set, the values for the date and time at the PDT output are valid.

CDT: Current date and time of the RTC instance. The CDT output is only updated when the function block is called. For this reason, instances of the function block should be called once in each PLC cycle.

Example:

See: Example: Software clocks (RTC, RTC_EX, RTC_EX2). [▶ 347]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.78 RTC_EX

The "RTC_EX" (Extended Real Time Clock) function block allows an internal Software clock to be implemented in TwinCAT PLC. The clock must be initialized with a starting date and time. After the initialization the time and date are updated with each call of the function block. A CPU system clock is used to calculate the current time and date. The function block should be called in every PLC cycle, so that the current time can be calculated. The current date and time are available in the usual DATE_AND_TIME (DT) format at the function block's output. In contrast to the RTC [▶ 135] function block, RTC_EX has a precision of one millisecond. Multiple instances of the RTC_EX function block can be created within one PLC program.

RTC_EX time can differ from the reference time

The way the system works means that the RTC_EX time can differ from the reference time. The difference depends on the PLC's cycle time, the value of the basic system ticks, and on the hardware being used.

In order to avoid larger deviations the RTC_EX instance should be synchronized cyclically (e.g. with a radio clock or with the local Windows system time). The local Windows system time you can be synchronized with a reference time via the SNTP protocol.

VAR_INPUT

```
VAR_INPUT
   EN     : BOOL;
   PDT    : DATE_AND_TIME;
   PMSEK : DWORD;
END_VAR
```

EN: The RTC_EX function block is re-initialized with a specified date, time and millisecond by a rising edge at this input.

PDT: (Preset Date and Time) The initialization values for the function block's date and time. A rising edge at the EN input will cause the function block to adopt this value.

PMSEK: (Preset Milliseconds) The initialisation value for the milliseconds. A rising edge at the EN input will cause the function block to adopt this value.

VAR_OUTPUT

```
VAR_OUTPUT
   Q     : BOOL;
   CDT   : DATE_AND_TIME;
   CMSEK : DWORD;
END_VAR
```

Q: This output is set if the function block has been initialized at least once. If the output is set, the values for the date, time and milliseconds at the PDT and CMSEK outputs are valid.

CDT: Current date and time of the RTC_EX instance. The CDT output is only updated when the function block is called. For this reason, instances of the function block should be called once in each PLC cycle.

CMSEK: (Current Milliseconds) The milliseconds output.

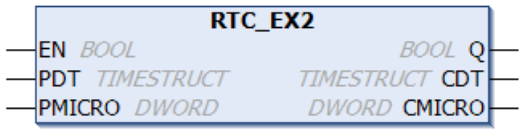
Example:

See: Example: Software clocks (RTC, RTC_EX, RTC_EX2). [▶ 347]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.79 RTC_EX2



The "RTC_EX2" (Extended Real Time Clock) function block allows an internal Software clock to be implemented in TwinCAT PLC. The clock must be initialized with a starting date and time. After the initialization the time and date are updated with each call of the function block. A CPU system clock is used to calculate the current time and date. The function block should be called in every PLC cycle, so that the current time can be calculated. At the function block output the current date and time is available in the Windows system time format. In contrast to the RTC [▶ 135] function block, RTC_EX2 offers microsecond accuracy. Multiple instances of the RTC_EX2 function block can be created within one PLC program.

RTC_EX2 time can differ from the reference time

The way the system works means that the RTC_EX2 time can differ from the reference time. The difference depends on the PLC's cycle time, the value of the basic system ticks, and on the hardware being used.

In order to avoid larger deviations the RTC_EX2 instance should be synchronized cyclically (e.g. with a radio clock or with the local Windows system time). The local Windows system time you can be synchronized with a reference time via the SNTP protocol.

VAR_INPUT

```
VAR_INPUT
   EN       : BOOL;
   PDT      : TIMESTRUCT;
   PMICRO   : DWORD;
END_VAR
```

EN: The RTC_EX2 function block is re-initialized with a specified date, time and millisecond by a rising edge at this input.

PDT: (Preset Date and Time) The initialisation values for the function block's date and time (type: TIMESTRUCT [▶ 321]). A rising edge at the EN input will cause the function block to adopt this value.

PMICRO: (Preset Milliseconds) The initialisation value for the milliseconds. A rising edge at the EN input will cause the function block to adopt this value.

VAR_OUTPUT

```
VAR_OUTPUT
   Q       : BOOL;
   CDT     : TIMESTRUCT;
   CMICRO  : DWORD;
END_VAR
```

Q: This output is set if the function block has been initialized at least once. If the output is set, the values for the date, time and milliseconds at the PDT and CMICRO outputs are valid.

CDT: Current date and time of the RTC_EX2 instance (type: TIMESTRUCT [▶ 321]). The CDT output is only updated when the function block is called. For this reason, instances of the function block should be called once in each PLC cycle.

CMICRO: (Current Microseconds) The microseconds output.

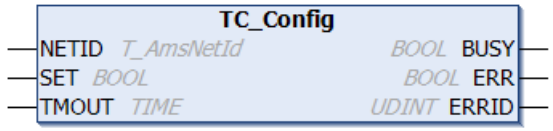
Example:

See: Example: Software clocks (RTC, RTC_EX, RTC_EX2). [▶ 347]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.80 TC_Config



A TwinCAT system in RUN mode (green TwinCAT system icon) can be switched to CONFIG mode (blue TwinCAT system icon) via the function block "TC_Config". If the system is already in CONFIG mode, it is first switched to STOP mode (red TwinCAT system icon) and then to CONFIG mode.

VAR_INPUT

```
VAR_INPUT
    NETID  : T_AmsNetId;
    SET    : BOOL;
    TMOUT  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: This parameter can be used to specify the AmsNetId of the TwinCAT computer that is to be switched to CONFIG mode (type: T_AmsNetID). If it is to be run on the local TwinCAT computer, an empty string can be entered.

SET: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.81 TC_CpuUsage

The "TC_CpuUsage" function block allows the current CPU loading of a TwinCAT system to be determined. This function corresponds to the display of CPU loading in the TwinCAT system menu under the real-time settings.

VAR_INPUT

```
VAR_INPUT
    NETID : T_AmsNetId;
    START : BOOL;
    TMOUT : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: It is possible here to provide the AmsNetId of the TwinCAT computer whose CPU loading is to be determined (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY   : BOOL;
    ERR    : BOOL;
    ERRID : UDINT;
    USAGE : UDINT;
END_VAR
```

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

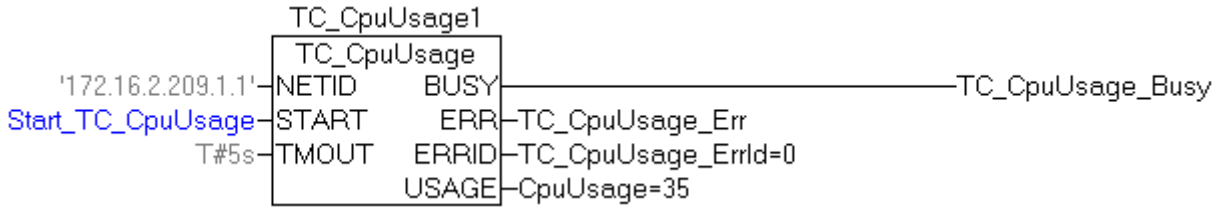
ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

USAGE: The current CPU loading of a TwinCAT system in %.

Example:

```
PROGRAM MAIN
VAR
    TC_CpuUsage1       : TC_CpuUsage;
    Start_TC_CpuUsage  : BOOL;
    TC_CpuUsage_Busy   : BOOL;
    TC_CpuUsage_Err    : BOOL;
    TC_CpuUsage_ErrId  : UDINT;
    CpuUsage           : UDINT;
END_VAR
```



In the example the TwinCAT system is using 35% of the total available CPU computing time.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.82 TC_Restart

The function block "TC_Restart" can be used to restart the TwinCAT system. The function corresponds to the Restart command on the TwinCAT system menu (on the right of the Windows taskbar). Restarting the TwinCAT system involves the TwinCAT system first being stopped, and then immediately started again.

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    RESTART : BOOL;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: It is possible here to provide the AmsNetId of the TwinCAT computer on which the TwinCAT system is to be restarted (type: T_AmsNetID). If the restart is to take place on the local computer, an empty string can be entered.

RESTART: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.83 TC_Stop

The function block "TC_Stop" can be used to stop the TwinCAT system. The function corresponds to the Stop command on the TwinCAT system menu (on the right of the Windows taskbar).

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    STOP    : BOOL;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: It is possible here to provide the AmsNetId of the TwinCAT computer on which the TwinCAT system is to be stopped (type: T_AmsNetID). If the TwinCAT system to be stopped is on the local computer, an empty string can be entered.

STOP: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

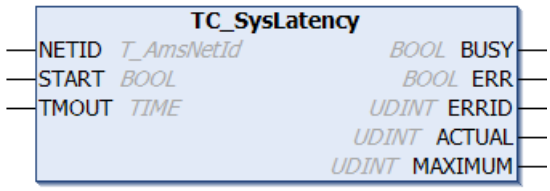
ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.84 TC_SysLatency



The function block "TC_SysLatency" can be used to determine the current and maximum latency time of a TwinCAT system. The function corresponds to the TwinCAT latency time display in the TwinCAT system menu under real-time settings.

VAR_INPUT

```
VAR_INPUT
    NETID   : T_AmsNetId;
    START   : BOOL;
    TMOUT   : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: This parameter can be used to specify the AmsNetID of the TwinCAT computer whose latency time is to be determined (type: T_AmsNetID). If it is to be run on the local computer, an empty string can be entered.

START: the block is activated by a positive edge at this input.

TMOUT: States the length of the timeout that may not be exceeded by execution of the ADS command.

VAR_OUTPUT

```
VAR_OUTPUT
    BUSY    : BOOL;
    ERR     : BOOL;
    ERRID   : UDINT;
    ACTUAL  : UDINT;
    MAXIMUM : UDINT;
END_VAR
```

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an error should occur during the transfer of the command, then this output is set once the BUSY output was reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

ACTUAL: The current latency time of a TwinCAT system in µs.

MAXIMUM: The maximum latency time of a TwinCAT system in µs (maximum latency time since the TwinCAT system was last started).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

3.85 WritePersistentData

If persistent variables are defined in a PLC runtime system, their current values are normally saved in a .bootdata file in the TwinCAT\Boot folder when stopping/shutting down the TwinCAT system (following the last PLC cycle). Before writing the current persistent data to the file, a backup of the old persistent data is made by renaming the system's old .bootdata file to .bootdata-old.

A file is created for every runtime system that is configured.

The next time the system starts, the .bootdata file is read and the persistent variables in the runtime system are initialized with the values from the file.

This backup file (.bootdata-old) of the persistent data is read at system startup if the file (.bootdata) containing the persistent data does not exist. This is an exception, but it can occur, for example, if an IPC without UPS experiences a power failure and TwinCAT could not shut down properly.

With the WritePersistentData function block you can initiate the saving of the persistent data from the PLC program and ensure that an up-to-date .bootdata file with the persistent data is available. The PORT input parameter specifies the runtime system whose persistent data is to be saved.

VAR_INPUT

```
VAR_INPUT
    NETID  : T_AmsNetId;
    PORT   : T_AmsPort;
    START  : BOOL;
    TMOUT  : TIME := DEFAULT_ADS_TIMEOUT;
END_VAR
```

NETID: Network address of the TwinCAT computer on which the ADS command is to be executed (type: T_AmsNetID). An empty string can be entered for the local computer.

PORT: ADS port number of the PLC runtime system whose persistent data is to be saved (type: T_AmsPort).

START: The function block is activated by a positive edge at this input.

TMOUT: Timeout time that may not be exceeded when executing the ADS command.

VAR_OUTPUT

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

BUSY: When the function block is activated this output is set. It remains set until a feedback is received.

ERR: If an ADS error should occur during the transfer of the command, then this output is set once the BUSY output is reset.

ERRID: Supplies the ADS error number [▶ 353] when the ERR output is set.

Example:

```
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;
```

See also: Appendix > System behavior when writing persistent data [▶ 349]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4 Functions

4.1 Time functions

4.1.1 DT_TO_FILETIME64

The function "DT_TO_FILETIME64" can be used to convert a PLC variable in DATE_AND_TIME format (DT) to FILETIME format (64 bit).

FUNCTION DT_TO_FILETIME64 : T_FILETIME64 [▶ 317]

VAR_INPUT

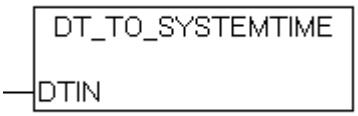
```
VAR_INPUT
    DTIN  : DT;
END_VAR
```

DTIN: The date and time to be converted, in DATE_AND_TIME format.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.44.0

4.1.2 DT_TO_SYSTEMTIME



The "DT_TO_SYSTEMTIME" function allows a PLC variable in DATE_AND_TIME format (DT) to be converted to a Windows system time structure. The system time has a resolution of 1ms, while the resolution of DATE_AND_TIME is 1s. The "wMilliseconds" variable in the system time structure therefore always returns the value zero.

FUNCTION DT_TO_SYSTEMTIME: TIMESTRUCT [▶ 321]

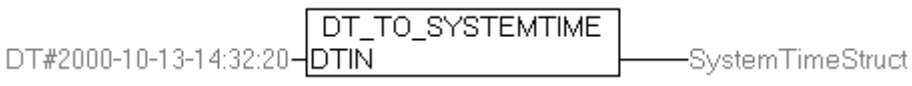
VAR_INPUT

```
VAR_INPUT
    DTIN  : DT;
END_VAR
```

DTIN: The date and time to be converted, in DATE_AND_TIME format.

Example:

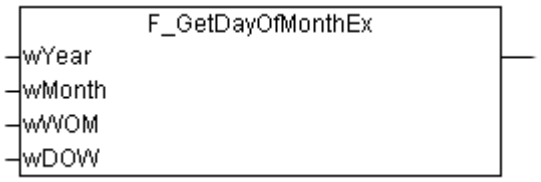
```
PROGRAM SystemTimeTest
VAR
    SystemTimeStruct : TIMESTRUCT;
END_VAR
```



Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.3 F_GetDayOfMonthEx



The function calculates the date of the first, second etc. weekday in a particular month and year (e.g. the date of the second Monday in January 2011).

FUNCTION F_GetDayOfMonthEx: WORD

VAR_INPUT

```
VAR_INPUT
    wYear  : WORD(1601..30827);
    wMonth : WORD(1..12);
    wWOM   : WORD(1..5);
    wDOW   : WORD(0..6);
END_VAR
```

wYear: year (1601 to 30827).

wMonth: month (1 to 12).

wWOM: week in month (1 of 5). The value 1 corresponds to week 1, 2 to week 2 and 5 to the last week (even if the month does not have 5 weeks).

wDOW: day of the week (0 to 6). 0 = Sunday, 1 = Monday... 6 = Saturday.

Return parameter	Description
0	Error, wrong or invalid function parameter
> 0	No error. Day of the month

Example:

The example determines the date of the second Monday in August 2011. The result is: 8.

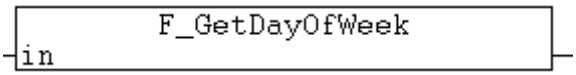
```
PROGRAM P_Dok_F_GetDayOfMonthEx
VAR
    wYear   : WORD := 2011;
    wMonth  : WORD := 8;
    wWOM    : WORD(1..5) := 2;(* Week of month: 2 = Second week *)
    wDOW    : WORD(0..6) := 1;(* Day of week 1 = Monday *)
    wDay    : WORD;(* Day of month *)
END_VAR
```

```
wDay := F_GetDayOfMonthEx( wYear, wMonth, wWOM, wDOW );
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.4 F_GetDayOfWeek



The function returns the number of the weekday according to DIN 1355 / ISO 8601. According to this standard the weekdays are numbered as follows: Monday = 1, Tuesday = 2, ... Sunday = 7.

FUNCTION F_GetDayOfWeek: WORD

VAR_INPUT

```
VAR_INPUT
    in : DT;
END_VAR
```

in: The date whose weekday number is to be determined.

Example:

```
PROGRAM MAIN
VAR
    dtFirst   : DT := DT#2008-01-01-00:00;
    dayOfWeek : WORD;
END_VAR
```

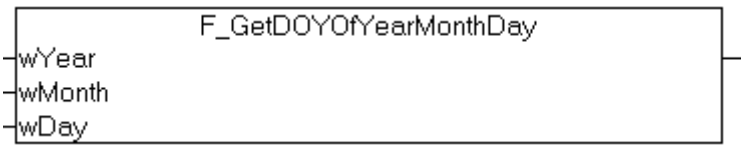
```
dayOfWeek := F_GetDayOfWeek(dtFirst);
```

The result is 2 (Tuesday)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.5 F_GetDOYOfYearMonthDay



The function calculates the number of the day in the year.

FUNCTION F_GetDOYOfYearMonthDay: WORD

VAR_INPUT

```
VAR_INPUT
    wYear  : WORD;
    wMonth : WORD;
    wDay   : WORD;
END_VAR
```

wYear: Year (0 ~ 2999).

wMonth: Month (1 ~ 12).

wDay : Day (1 ~ 31).

Return parameter	Description
0	Error, wrong wYear, wMonth or wDay parameter value

Return parameter	Description
> 0	No error. Number of the days in the year (1 ~ 366)

Example:

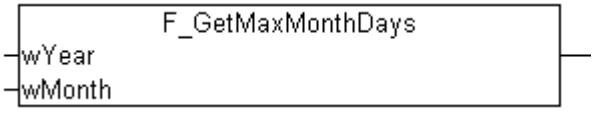
```
PROGRAM P_TEST_DOY
VAR
    wYear  : WORD;
    wDOY   : WORD;
    wMonth : WORD;
    wDay   : WORD;
END_VAR
```

```
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 *)
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.6 F_GetMaxMonthDays



The function returns the maximum number of days of the month in a certain month and year.

FUNCTION F_GetMaxMonthDays: WORD

VAR_INPUT

```
VAR_INPUT
    wYear  : WORD;
    wMonth : WORD;
END_VAR
```

wYear: Year.

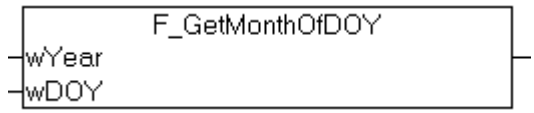
wMonth: Month (1 to 12).

Return parameter	Description
0	Error, wrong wMonth parameter value
> 0	No error. Maximum number of days of the month.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.7 F_GetMonthOfDOY



The function calculates the month based on the day number in the year.

FUNCTION F_GetMonthOfDOY: WORD

VAR_INPUT

```
VAR_INPUT
    wYear   : WORD;
    wDOY    : WORD;
END_VAR
```

wYear: Year (0 ~ 2999).

wDOY: Number of the day in the specified year whose month is to be determined (1 ~ 366).

Return parameter	Description
0	Error, wrong wYear or wDOY parameter value.
> 0	No error. Month (1 ~ 12).

Example:

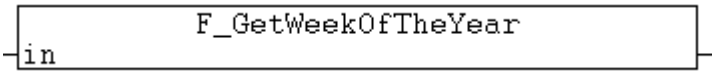
```
PROGRAM P_TEST_DOY
VAR
    wYear   : WORD;
    wDOY    : WORD;
    wMonth : WORD;
END_VAR
```

```
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 *)
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.8 F_GetWeekOfTheYear



The function returns the calendar week number for a specified date according to the DIN 1355 / ISO 8601 standard.

- The first calendar week is defined as the first week that contains a **minimum of four days of the new year (DIN 1355 / ISO 8601)**;
- The calendar weeks start on a Monday. Each calendar week contains 7 days;
- The returned value in the first calendar week has the number 1;
- 29, 30 and 31 December may belong to the first calendar week of the following year;

- 1, 2 and 3 January may belong to the last calendar week of the previous year;

FUNCTION F_GetWeekOfTheYear: WORD

VAR_INPUT

```
VAR_INPUT
    in : DT;
END_VAR
```

in: The date whose calendar week is to be determined.

Example:

```
PROGRAM MAIN
VAR
    dtNow      : DT := DT#2008-03-17-12:00;
    weekOfYear : WORD;
END_VAR


weekOfYear := F_GetWeekOfTheYear(dtNow);
```

The result is 12.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.9 F_TranslateFileTime64Bias

This function converts the input time to the time in another time zone based on the specified bias time shift. The function can be used to convert the local time to UTC time (Universal Time Coordinates) and vice versa, for example.

FUNCTION F_TranslateFileTime64Bias: T_FILETIME64 [▶ 317]

VAR_INPUT

```
VAR_INPUT
    in    : T_FILETIME64;
    bias  : DINT;
    toUTC : BOOL;
END_VAR
```

in: Input time that is to be converted (type: T_FILETIME64 [▶ 317]).

bias: Difference between UTC time and local time in minutes (positive or negative values are permitted).

toUTC: This parameter can be used to specify the direction in which the input time is to be converted.

toUTC	Direction	Internal formula
FALSE	UTC -> local time	Local time := UTC - bias
TRUE	Local time -> UTC	UTC := local time + bias

Sample:

The *in* variable contains the time to be converted. The *bToUTC* variable determines the conversion direction. If *bToUTC* = TRUE the local time is converted to UTC time, if *bToUTC* = FALSE the UTC time is converted to local time. The *WEST_EUROPE_TZI* constant contains the time zone information for Western Europe. The required bias value is calculated from the time zone information in the constant and the current bDST setting (Daylight Saving Time). The current time zone information of a TwinCAT system can alternatively be determined with the function block: FB_GetTimeZoneInformation [▶ 77].

Important notice: Data type DT was selected for the input time because of the visual control option in online mode. Conversions to other time formats are not necessarily recommend since the conversion functions can be very computing-intensive.

```
PROGRAM MAIN
VAR
    bDST    : BOOL := TRUE;(* TRUE => Daylight saving time, FALSE => Standard time *)
    bToUTC  : BOOL := FALSE;
(* TRUE => Convert local time to UTC time, FALSE => Convert UTC time to local time *)
    in      : DT := DT#2011-08-29-15:15:31;
    out     : DT;
    bias    : DINT;
END_VAR
```

```
IF bDST THEN
    bias := WEST_EUROPE_TZI.bias + WEST_EUROPE_TZI.daylightBias;
ELSE
    bias := WEST_EUROPE_TZI.bias + WEST_EUROPE_TZI.standardBias;
END_IF

out := FILETIME64_TO_DT( F_TranslateFileTime64Bias( DT_TO_FILETIME64( in ), bias, bToUTC ) );
```

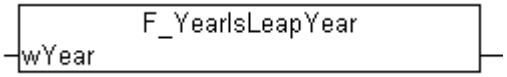
Further time and time zone functions and function blocks:

- FB_TzSpecificLocalTimeToSystemTime [▶ 115]
- FB_TzSpecificLocalTimeToFileTime64 [▶ 112]
- FB_SystemTimeToTzSpecificLocalTime [▶ 111]
- FB_FileTime46ToTzSpecificLocalTime [▶ 61]
- FB_GetTimeZoneInformation [▶ 77]
- FB_SetTimeZoneInformation [▶ 108]
- NT_SetLocalTime [▶ 122]
- NT_GetTime [▶ 120]
- NT_SetTimeToRTCTime [▶ 123]
- FB_LocalSystemTime [▶ 89]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.44.0

4.1.10 F_YearIsLeapYear



The function determines whether a year is a leap year.

FUNCTION F_YearIsLeapYear: BOOL

VAR_INPUT

```
VAR_INPUT
    wYear : WORD;
END_VAR
```

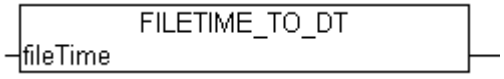
wYear: Year.

Return parameter	Description
TRUE	Changeover year
FALSE	No leap year

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.11 FILETIME64_TO_DT



The function "FILETIME64_TO_DT" converts the time in FILETIME format to DATE_AND_TIME format (DT). The DT format has a smaller value range than the FILETIME format and only offers second accuracy. For this reason the FILETIME value to be converted is limited. The permitted minimum corresponds to value *DT#1970-01-01-00:00:00* and the maximum to value *DT#2106-02-06-06:28:15*. Milliseconds are not considered in the conversion and are rounded down to the DATE_AND_TIME return value accordingly.

FUNCTION FILETIME64_TO_DT : DT

VAR_INPUT

```
VAR_INPUT
    fileTime : T_FILETIME64;
END_VAR
```

fileTime: The time to be converted in the FILETIME format (type: T_FILETIME64 [▶ 317]).

Sample:

```
PROGRAM MAIN
VAR
    timeAsFileTime : T_FILETIME64;
    timeAsDT       : DT;
END_VAR
```

```
timeAsFileTime := F_GetSystemTime();
timeAsDT := FILETIME64_TO_DT( timeAsFileTime );
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.44.0

4.1.12 FILETIME64_TO_ISO8601

This function converts the Windows system time in the T_FILETIME64 format to a string with the format of the ISO 8601 standard.

The result conforms to the following pattern: **YYYY-MM-DDThh:mm:ss.xxxTZD**

FUNCTION FILETIME64_TO_ISO8601 : STRING(39)

VAR_INPUT

```
VAR_INPUT
    fileTime   : T_FILETIME64; (* Time to be converted (file time format), 64-
bit value representing the number of 100-nanosecond intervals since January 1, 1601 *)
    nBias      : INT;          (* Specifies the current bias, in minutes, for local time translation
 on this computer.
                                The bias is the difference, in minutes, between Coordinated Universa
l Time (UTC) and local time.
                                UTC = local time + bias *)
    bUTC       : BOOL;         (* Specifies whether the fileTime is UTC or local time. *)
    nPrecision : USINT(0..9);  (* Precision. Number of decimal places of seconds. (0..9) *)
END_VAR
```

fileTime: Specifies the time to be converted (Type: T_FILETIME64 [▶ 317]).

nBias: Specifies the current time offset in minutes between the Coordinated Universal Time (UTC) and the local time. The following applies: UTC = local time + time offset

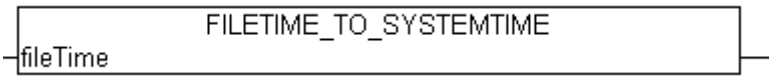
bUTC: Specifies whether the time specified at the input corresponds to the UTC or the local time.

nPrecision: Specifies the accuracy of the seconds representation as a number of decimal places.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.46.0

4.1.13 FILETIME64_TO_SYSTEMTIME



The function "FILETIME64_TO_SYSTEMTIME" converts the time in FILETIME format into the "readable" SYSTEMTIME format. The conversion fails if the most significant bit of the 64-bit FILETIME variables is set. In this case the TIMESTRUCT member variables have the value zero.

FUNCTION FILETIME64_TO_SYSTEMTIME: TIMESTRUCT [▶ 321]

VAR_INPUT

```
VAR_INPUT
    fileTime : T_FILETIME64;
END_VAR
```

fileTime: The time to be converted in the FILETIME format (type: T_FILETIME64 [▶ 317]).

Sample:

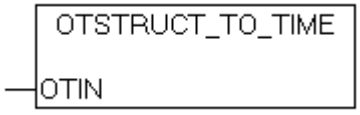
```
PROGRAM MAIN
VAR
    timeAsFileTime   : T_FILETIME64;
    timeAsSystemTime : TIMESTRUCT;
END_VAR
```

```
timeAsFileTime := F_GetSystemTime();
timeAsSystemTime := FILETIME64_TO_SYSTEMTIME( timeAsFileTime );
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.44.0

4.1.14 OTSTRUCT_TO_TIME



The function "OTSTRUCT_TO_TIME" can be used to convert a structure with resolved milliseconds, seconds, minutes, hours, days and weeks into a TIME variable.

FUNCTION OTSTRUCT_TO_TIME: TIME

VAR_INPUT

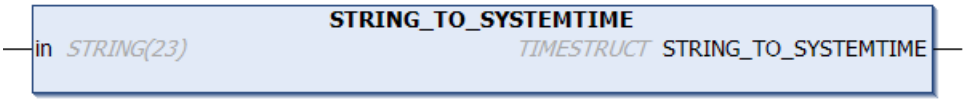
```
VAR_INPUT
    OTIN : OTSTRUCT;
END_VAR
```

OTIN: The structure to be converted (type: OTSTRUCT [▶ 303]).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.15 STRING_TO_SYSTEMTIME



The function converts a string into the Windows SYSTEMTIME format.

FUNCTION STRING_TO_SYSTEMTIME: TIMESTRUCT [▶ 321]

VAR_INPUT

```
VAR_INPUT
    in : STRING(23);
END_VAR
```

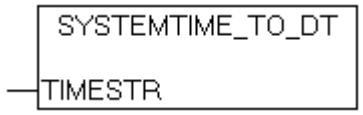
in: The string to be converted. The string must have the following format: **'YYYY-MM-DD-hh:mm:ss.xxx'**

- YYYY: Year (1601..9999);
- MM: Month (01..12);
- DD: Day (01..31);
- hh: Hour (00..23);
- mm: Minute (00..59);
- ss: Second (00..59);
- xxx: Millisecond (000..999);

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.16 SYSTEMTIME_TO_DT



The "SYSTEMTIME_TO_DT" function allows the Windows system time structure to be converted to the DATE_AND_TIME format (DT) usual in a PLC. The system time has a resolution of 1ms, while the resolution of DATE_AND_TIME is 1s. The milliseconds from the system time are used in the course of the conversion to determine the direction of rounding for the returned DATE_AND_TIME value. To disable rounding, set the wMilliseconds element in the Windows system time structure to zero.

FUNCTION SYSTEMTIME_TO_DT: DT

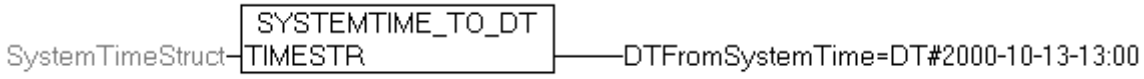
VAR_INPUT

```
VAR_INPUT
    TIMESTR : TIMESTRUCT;
END_VAR
```

TIMESTR: The structure with the Windows system time requiring conversion (type: TIMESTRUCT [▶ 321]).

Example:

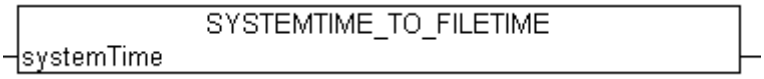
```
PROGRAM SystemTimeTest
VAR
    SystemTimeStruct  : TIMESTRUCT;
    DTFromSystemTime  : DT;
END_VAR
```



Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.17 SYSTEMTIME_TO_FILETIME64



The function can be used to convert the Windows system time structure into the Filetime format. The day of the week wDayOfWeek of the system time variable is ignored. The system time year must be greater than 1601 and less than 30827.

FUNCTION SYSTEMTIME_TO_FILETIME64 : T_FILETIME64 [▶ 317]

VAR_INPUT

```
VAR_INPUT
    systemTime : TIMESTRUCT;
END_VAR
```

systemTime: The structure with the Windows system time requiring conversion (type: TIMESTRUCT [▶ 321]).

Return parameter	Description
0	Error, wrong system time parameter value.

Return parameter	Description
> 0	No error. File time.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.44.0

4.1.18 SYSTEMTIME_TO_ISO8601

This function converts the Windows system time structure to a string with the format of the ISO 8601 standard.

The result conforms to the following pattern: **YYYY-MM-DDThh:mm:ss.xxxTZD**

FUNCTION SYSTEMTIME_TO_ISO8601 : STRING(39)

VAR_INPUT

```
VAR_INPUT
    systemTime  : TIMESTRUCT;  (* Input time in system time format (struct) *)
    nBias       : INT;         (* Specifies the current bias, in minutes, for local time translation
 on this computer.
                                The bias is the difference between Coordinated Universal Time (UTC)
and local time.
                                UTC = local time + bias *)
    bUTC        : BOOL;        (* Specifies whether the systemTime is UTC or local time. *)
    nPrecision  : USINT(0..9); (* Precision. Number of decimal places of seconds. (0..9) *)
END_VAR
```

systemTime: Structure with the Windows system time to be converted (type: TIMESTRUCT [▶ 321]).

nBias: Specifies the current time offset in minutes between the Coordinated Universal Time (UTC) and the local time. The following applies: UTC = local time + time offset

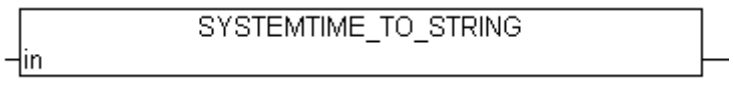
bUTC: Specifies whether the time specified at the input corresponds to the UTC or the local time.

nPrecision: Specifies the accuracy of the seconds representation as a number of decimal places.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.46.0

4.1.19 SYSTEMTIME_TO_STRING



The function converts the Windows system time structure into a string with the following format: **YYYY-MM-DD-hh:mm:ss.xxx**:

- YYYY: Year (1601..9999)
- MM: Month (01..12);
- DD: Day (01..31)
- hh: Hour (00..23)
- mm: Minutes (00..59)
- ss: Second (00..59)

- xxx: Millisecond (000..999)

FUNCTION SYSTEMTIME_TO_STRING: STRING(24)

VAR_INPUT

```
VAR_INPUT
    in : TIMESTRUCT;
END_VAR
```

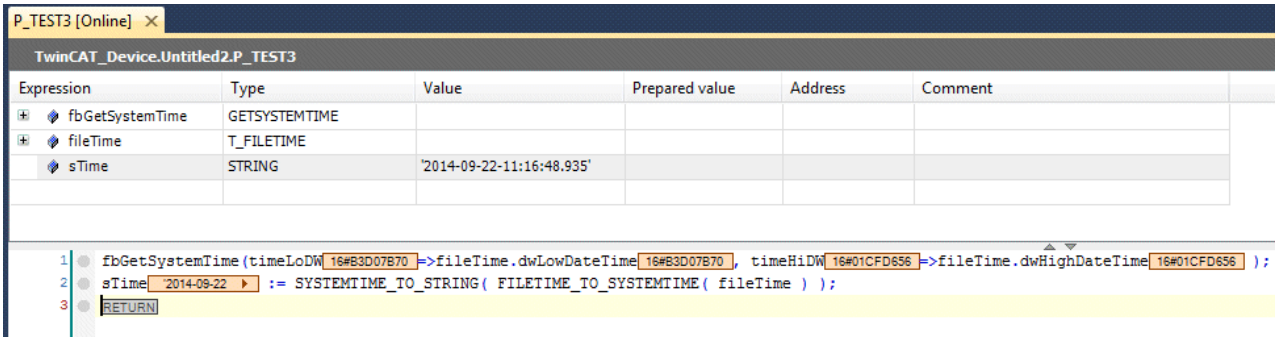
in: The structure with the Windows system time requiring conversion (type: TIMESTRUCT [▶ 321]).

Example:

```
PROGRAM MAIN
VAR
    fbGetSystemTime  : GETSYSTEMTIME;
    fileTime         : T_FILETIME;
    sTime            : STRING;
END_VAR
```

```
fbGetSystemTime(timeLoDW=>fileTime.dwLowDateTime, timeHiDW=>fileTime.dwHighDateTime );
sTime := SYSTEMTIME_TO_STRING( FILETIME_TO_SYSTEMTIME( fileTime ) );
```

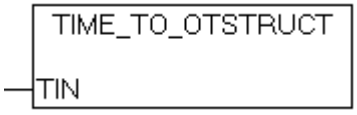
Online view:



Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.1.20 TIME_TO_OTSTRUCT



The function "TIME_TO_OTSTRUCT" can be used to convert a TIME constant or variable into a structure with the resolved milliseconds, seconds, minutes, hours, days and weeks.

FUNCTION TIME_TO_OTSTRUCT: OTSTRUCT [▶ 303]

VAR_INPUT

```
VAR_INPUT
    TIN : TIME;
END_VAR
```

TIN: The TIME variable to be converted.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.2 Extended STRING functions

4.2.1 CHAR_TO_WCHAR

The function converts a variable of the data type STRING into a variable of the data type WSTRING (with null termination).

FUNCTION CHAR_TO_WCHAR : WSTRING(1)

VAR_INPUT

```
VAR_INPUT
    sTextIn : STRING(1);
END_VAR
```

sTextIn: STRING variable to be converted.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.2 CONCAT2

The function concatenates two strings of the data type STRING of any length and checks whether the resulting string is longer than a specified output string. In this case the string is truncated.

The function returns

- TRUE if the concatenation was successful.
- FALSE if the resulting string is longer than the output string and doesn't fit in the given output buffer. The memory requirement for the resulting string is then larger than that for the output string. The string is then truncated.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION CONCAT2 : BOOL

VAR_INPUT

```
VAR_INPUT
    pSrcString1 : POINTER TO STRING;
    pSrcString2 : POINTER TO STRING;
    pDstString  : POINTER TO STRING;
    nDstSize    : UDINT;
END_VAR
```

pSrcString1: Pointer to the first of the STRING variables to be concatenated (input string)

pSrcString2: Pointer to the second of the STRING variables to be concatenated (input string)

pDstString: Pointer to the resulting STRING variable following the concatenation (output string)

nDstSize: Size of the resulting STRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.3 DATA_TO_HEXSTR2

The function converts binary data into a hexadecimal string. This function can be used to convert simple data types and structure variables. If the maximum possible length of the output is exceeded, a dot character is added to the result string ('.') and the conversion is aborted. The remaining data bytes are not converted.

FUNCTION DATA_TO_HEXSTR2 : UDINT

VAR_INPUT

```
VAR_INPUT
    pSrcData    : POINTER TO BYTE;  // pointer to data buffer
    nSrcSize    : UDINT;            // size of data buffer in bytes (= number of bytes to be convert
ed)
    pDstHexStr  : POINTER TO STRING;// pointer to destination buffer
    nDstSize    : UDINT;            // size of destination buffer in bytes
    bLoCase     : BOOL;             // default: use "ABCDEF", if TRUE use "abcdef" characters
END_VAR
```

pSrcData: Start address (pointer) for the binary data to be converted. The address can be determined with the ADR operator.

nSrcSize: Max. size (in bytes) of the binary data to be converted. The size can be determined with the SIZEOF operator.

pDstHexStr: Start address (pointer) to the destination buffer into which the converted hexadecimal string is to be written. The address can be determined with the ADR operator.

nDstSize: Max. available size (in bytes) of the destination buffer. The size can be determined with the SIZEOF operator.

bLoCase: This parameter determines whether upper or lower case letters are to be used in the conversion. TRUE = lower case letters, FALSE = upper case letters.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.5.1.0

4.2.4 DELETE2

The function removes nLen characters from a string, starting at position nPos.

The function returns

- TRUE if the characters were successfully removed.
- FALSE if the resulting string is longer than the output string and doesn't fit in the given output buffer. The memory requirement for the resulting string is then larger than that for the output string. The string is then truncated.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION DELETE2 : BOOL

VAR_INPUT

```
VAR_INPUT
    pSrcString : POINTER TO STRING;
    pDstString : POINTER TO STRING;
    nDstSize   : UDINT;
    nLen       : UDINT
    nPos       : UDINT
END_VAR
```

pSrcString: Pointer to the STRING variable (input string)

pDstString: Pointer to the resulting STRING variable (output string)

nDstSize: Size of the resulting STRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

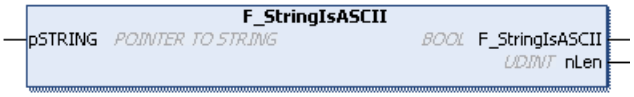
nLen: Number of characters to be removed

nPos: Position of the first character to be removed; the following characters are also to be removed (nPos = 1 = first character)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.35.0

4.2.5 F_StringIsASCII



The function checks whether a string contains only ASCII characters (0x000 to 0x7F) and returns the number of ASCII characters. The string is directly compatible to UTF-8 if it contains only ASCII characters.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION F_StringIsASCII : BOOL

The return value is TRUE if the string contains only ASCII characters.

VAR_INPUT

```
VAR_INPUT
    pSTRING : POINTER TO STRING;
END_VAR
```

pString: pointer to the STRING variable.

VAR_OUTPUT

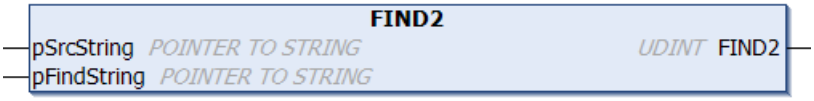
```
VAR_OUTPUT
    nLen : UDINT;
END_VAR
```

nLen: Number of ASCII characters in the string

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.6 FIND2



The function finds a string, which may occur more than once, in another string.

The function returns

- is the position of the first character of the first string that was found.
- the value 0 if the string was not found.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION FIND2 : UDINT

VAR_INPUT

```
VAR_INPUT
    pSrcString  : POINTER TO STRING;
    pFindString : POINTER TO STRING;
END_VAR
```

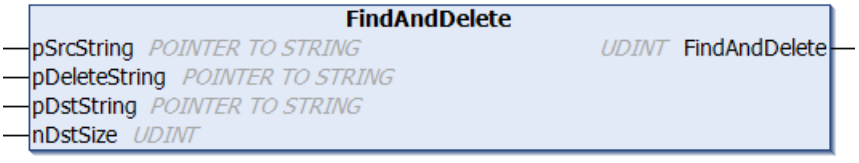
pSrcString: Pointer to the STRING variable whose string is to be searched

pFindString: Pointer to the STRING variable whose string is being searched for

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.35.0

4.2.7 FindAndDelete



The function finds a string, which may occur more than once, in another string and removes it.

The function returns

- the number of removed strings.
- the value 0 if the string was not found.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION FindAndDelete : UDINT

VAR_INPUT

```
VAR_INPUT
    pSrcString    : POINTER TO STRING;
    pDeleteString : POINTER TO STRING;
    pDstString    : POINTER TO STRING;
    nDstSize      : UDINT;
END_VAR
```

pSrcString: Pointer to the STRING variable whose string is to be searched (input string)

pDeleteString: Pointer to the STRING variable whose string is to be searched for and removed (input string)

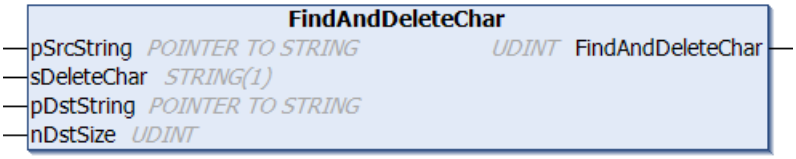
pDstString: Pointer to the resulting STRING variable (output string)

nDstSize: Size of the resulting STRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.35.0

4.2.8 FindAndDeleteChar



The function finds a character, which may occur more than once, in a string and removes it.

The function returns

- the number of removed characters.
- the value 0 if the character was not found.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION FindAndDeleteChar : UDINT

VAR_INPUT

```
VAR_INPUT
    pSrcString    : POINTER TO STRING;
    sDeleteChar   : STRING(1);
    pDstString    : POINTER TO STRING;
    nDstSize      : UDINT;
END_VAR
```

pSrcString: Pointer to the STRING variable whose string is to be searched (input string)

sDeleteChar: Character to be removed

pDstString: Pointer to the resulting STRING variable (output string)

nDstSize: Size of the resulting STRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.35.0

4.2.9 FindAndReplace

The function finds a string, which may occur more than once, in another string and replaces it with another string.

The function returns

- the number of replaced strings.
- the value 0 if the string was not found.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION FindAndReplace : UDINT

VAR_INPUT

```
VAR_INPUT
    pSrcString    : POINTER TO STRING;
    pDeleteString : POINTER TO STRING;
    pInsertString : POINTER TO STRING;
    pDstString    : POINTER TO STRING;
    nDstSize      : UDINT;
END_VAR
```

pSrcString: Pointer to the STRING variable whose string is to be searched (input string)

pDeleteString: Pointer to the STRING variable whose string is to be replaced (input string)

pInsertString: Pointer to the STRING variable whose string is to replace the other string (input string)

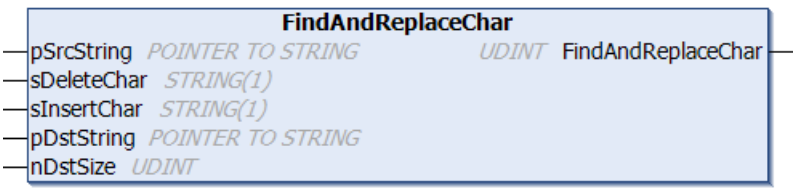
pDstString: Pointer to the resulting STRING variable (output string)

nDstSize: Size of the resulting STRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.35.0

4.2.10 FindAndReplaceChar



The function finds a character, which may occur more than once, in a string and replaces it with another character.

The function returns

- the number of replaced characters.
- the value 0 if the character was not found.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION FindAndReplaceChar : UDINT

VAR_INPUT

```
VAR_INPUT
    pSrcString    : POINTER TO STRING;
    sDeleteChar   : STRING(1);
    sInsertChar   : STRING(1);
    pDstString    : POINTER TO STRING;
    nDstSize      : UDINT;
END_VAR
```

pSrcString: Pointer to the STRING variable whose string is to be searched (input string)

sDeleteChar: Character to be replaced

sInsertChar: Character to replace the other character

pDstString: Pointer to the resulting STRING variable (output string)

nDstSize: Size of the resulting STRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.35.0

4.2.11 FindAndSplit

The function splits a string into two strings.

The character string is searched for a separator (e.g. "\") from left to right. The strings on both sides of this first occurrence of the separator are output.

The search direction can be changed with the parameter bSearchFromRight so that the string is searched from right to left.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION FindAndSplit : BOOL

The return value is TRUE if the separator was found and the division and output of the strings was successful.

VAR_INPUT

```
VAR_INPUT
    pSeparator        : POINTER TO STRING;
    pSrcString        : POINTER TO STRING;
    pLeftString       : POINTER TO STRING;
    nLeftSize         : UDINT;
    pRightString      : POINTER TO STRING;
    nRightSize        : UDINT;
    bSearchFromRight  : BOOL;
END_VAR
```

pSeparator: Pointer to the STRING variable that represents the separator.

pSrcString: Pointer to the STRING variable that represents the source string.

pLeftString: Pointer to the STRING variable to which the separated left-hand string is to be output.

nLeftSize: Maximum size of the separated left string.

pRightString: Pointer to the STRING variable to which the separated right-hand string is to be output.

nRightSize: Maximum size of the separated right string.

bSearchFromRight: If the input is set, the search direction changes so that the string is searched for the separator from right to left.

Sample "Splitting into several substrings":

This sample shows how to split a string "machines/machine1/module2/data/tx' into several strings ['machines', 'machine1', 'module2', 'data', 'tx']. To this end the function FindAndSplit() is called repeatedly in a loop.

```
PROGRAM MAIN
VAR
    sSrc         : STRING(255) := 'machines/machine1/module2/data/tx';
    sSeparator   : STRING(1) := '/';
    aSplit       : ARRAY[1..cMax] OF STRING(255);
    bResultSplit : BOOL;
    i            : UDINT;
END_VAR
VAR CONSTANT
    cMax         : UDINT := 9;
END_VAR
```

```
aSplit[1] := sSrc;
FOR i:=1 TO cMax-1 DO
    bResultSplit := FindAndSplit( pSeparator     := ADR(sSeparator),  pSrcString := ADR(aSplit[i]),
                                  pLeftString    := ADR(aSplit[i]),   nLeftSize  := SIZEOF(aSplit[i]
),
                                  pRightString   := ADR(aSplit[i+1]), nRightSize := SIZEOF(aSplit[i+
1]),
                                  bSearchFromRight := FALSE );
    IF NOT bResultSplit THEN
        EXIT;
    END_IF
END_FOR
```

Sample "Merging several strings":

This sample shows how to combine several strings ['machines', 'machine1', 'module2', 'data', 'tx'] into one string 'machines/machine1/module2/data/tx'.

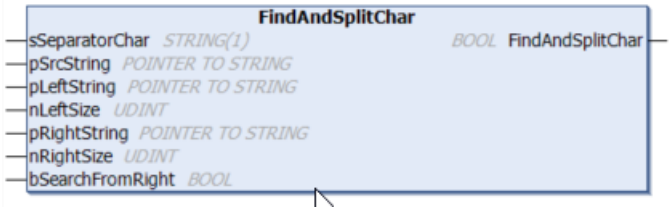
```
PROGRAM MAIN
VAR
    sSeparator    : STRING(1) := '/';
    aSplit        : ARRAY[1..cMax] OF STRING(255) := ['machines', 'machine1', 'module2', 'data',
'tx'];
    sJoined       : STRING(255);
    bResultConcat : BOOL;
    i             : UDINT;
END_VAR
VAR CONSTANT
    cMax          : UDINT := 5;
END_VAR
```

```
sJoined := aSplit[1];
FOR i:=1 TO cMax-1 DO
    bResultConcat := CONCAT2(pSrcString1 := ADR(sJoined), pSrcString2 := ADR(sSeparator), pDstString
 := ADR(sJoined), nDstSize := SIZEOF(sJoined));
    IF NOT bResultConcat THEN
        EXIT;
    END_IF
    bResultConcat := CONCAT2(pSrcString1 := ADR(sJoined), pSrcString2 := ADR(aSplit[i+1]), pDstStrin
g := ADR(sJoined), nDstSize := SIZEOF(sJoined));
    IF NOT bResultConcat THEN
        EXIT;
    END_IF
END_FOR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024.11	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.39.0

4.2.12 FindAndSplitChar



The function splits a string into two strings.

The character string is searched for a separator (e.g. "\") from left to right. The strings on both sides of this first occurrence of the separator are output.

The search direction can be changed with the parameter bSearchFromRight so that the string is searched from right to left.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION FindAndSplitChar : BOOL

The return value is TRUE if the separator was found and the division and output of the strings was successful.

VAR_INPUT

```
VAR_INPUT
    sSeparatorChar    : STRING(1);
    pSrcString        : POINTER TO STRING;
    pLeftString       : POINTER TO STRING;
    nLeftSize         : UDINT;
    pRightString      : POINTER TO STRING;
    nRightSize        : UDINT;
    bSearchFromRight  : BOOL;
END_VAR
```

sSeparatorChar: Character representing the separator.

pSrcString: Pointer to the STRING variable that represents the source string.

pLeftString: Pointer to the STRING variable to which the separated left-hand string is to be output.

nLeftSize: Maximum size of the separated left string.

pRightString: Pointer to the STRING variable to which the separated right-hand string is to be output.

nRightSize: Maximum size of the separated right string.

bSearchFromRight: If the input is set, the search direction changes so that the string is searched for the separator from right to left.

Sample "Splitting into several substrings":

This sample shows how to split a string "machines/machine1/module2/data/tx' into several strings ['machines', 'machine1', 'module2', 'data', 'tx']. To this end the function FindAndSplitChar() is called repeatedly in a loop.

```
PROGRAM MAIN
VAR
    sSrc         : STRING(255) := 'machines/machine1/module2/data/tx';
    aSplit       : ARRAY[1..cMax] OF STRING(255);
    bResultSplit : BOOL;
    i            : UDINT;
END_VAR
VAR CONSTANT
    cMax         : UDINT := 9;
END_VAR
```

```
aSplit[1] := sSrc;
FOR i:=1 TO cMax-1 DO
    bResultSplit := FindAndSplitChar( sSeparatorChar := '/',  pSrcString := ADR(aSplit[i]),
                                      pLeftString    := ADR(aSplit[i]),   nLeftSize  := SIZEOF(aSpli
t[i]),
                                      pRightString   := ADR(aSplit[i+1]), nRightSize := SIZEOF(aSpli
t[i+1]),
                                      bSearchFromRight := FALSE );
    IF NOT bResultSplit THEN
        EXIT;
    END_IF
END_FOR
```

Sample "Merging several strings":

This sample shows how to combine several strings ['machines', 'machine1', 'module2', 'data', 'tx'] into one string 'machines/machine1/module2/data/tx'.

```
PROGRAM MAIN
VAR
    sSeparator    : STRING(1) := '/';
    aSplit        : ARRAY[1..cMax] OF STRING(255) := ['machines', 'machine1', 'module2', 'data',
'tx'];
    sJoined       : STRING(255);
    bResultConcat : BOOL;
    i             : UDINT;
END_VAR
VAR CONSTANT
    cMax          : UDINT := 5;
END_VAR
```

```
sJoined := aSplit[1];
FOR i:=1 TO cMax-1 DO
    bResultConcat := CONCAT2(pSrcString1 := ADR(sJoined), pSrcString2 := ADR(sSeparator), pDstString
 := ADR(sJoined), nDstSize := SIZEOF(sJoined));
    IF NOT bResultConcat THEN
        EXIT;
    END_IF
    bResultConcat := CONCAT2(pSrcString1 := ADR(sJoined), pSrcString2 := ADR(aSplit[i+1]), pDstStrin
g := ADR(sJoined), nDstSize := SIZEOF(sJoined));
    IF NOT bResultConcat THEN
        EXIT;
    END_IF
END_FOR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024.11	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.39.0

4.2.13 HEXSTR_TO_DATA2

The function converts a hexadecimal string into binary data and returns the number of successfully converted data bytes as result. Only spaces may be used as separators in the hexadecimal string to be converted. Lower and upper case letters are permitted as hex characters. In the event of an error or an illegal character the conversion is aborted and a zero length is returned as result.

FUNCTION HEXSTR_TO_DATA2 : UDINT

VAR_INPUT

```
VAR_INPUT
    pSrcHexStr  : POINTER TO STRING;  // hex string to convert (Example: "AF 34 55 EC")
    pDstData    : POINTER TO BYTE;    // pointer to destination buffer
    nDstSize    : UDINT;              // size of destination buffer in bytes
END_VAR
```

pSrcHexStr: Start address (pointer) to the hexadecimal string to be converted (e.g.: 'AB CD 01 23'). The address can be determined with the ADR operator.

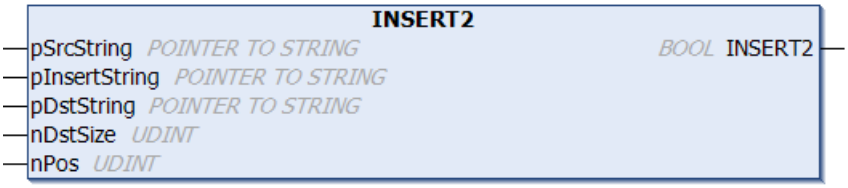
pDstData: Start address (pointer) to the destination buffer into which the converted data bytes are to be written. The address can be determined with the ADR operator.

nDstSize: Max. available size (in bytes) of the destination buffer. The size can be determined with the SIZEOF operator.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.5.1.0

4.2.14 INSERT2



The function inserts a string into another string after position nPos. If nPos = 0, the string is inserted before the first character of the other string.

The function returns

- TRUE if the string was successfully inserted.
- FALSE if the resulting string is longer than the output string and doesn't fit in the given output buffer. The memory requirement for the resulting string is then larger than that for the output string. The string is then truncated.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION INSERT2 : BOOL

VAR_INPUT

```
VAR_INPUT
    pSrcString    : POINTER TO STRING;
    pInsertString : POINTER TO STRING;
    pDstString    : POINTER TO STRING;
    nDstSize      : UDINT;
    nPos          : UDINT;
END_VAR
```

pSrcString: Pointer to the STRING variable whose string is to be searched (input string)

pInsertString: Pointer to the STRING variable whose string is to be inserted in the other string (input string)

pDstString: Pointer to the resulting STRING variable (output string)

nDstSize: Size of the resulting STRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

nPos: Position of the character after which the string is to be inserted

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.35.0

4.2.15 LEN2

The function returns the number of characters in a string (length of the STRING).

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION LEN2 : UDINT

VAR_INPUT

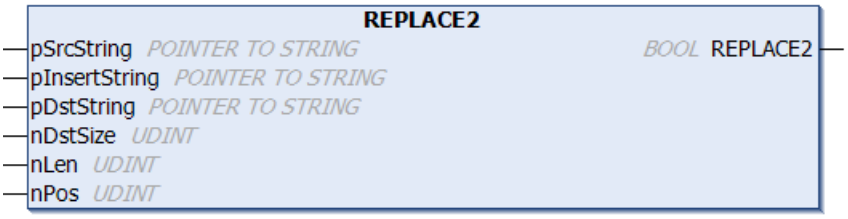
```
VAR_INPUT
    pSTRING : POINTER TO STRING;
END_VAR
```

pString: pointer to the STRING variable

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.16 REPLACE2



The function replaces nLen characters of a string with another string, starting at position nPos.

The function returns

- TRUE if the characters were successfully replaced.
- FALSE if the resulting string is longer than the output string and doesn't fit in the given output buffer. The memory requirement for the resulting string is then larger than that for the output string. The string is then truncated.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION REPLACE2 : BOOL

VAR_INPUT

```
VAR_INPUT
    pSrcString    : POINTER TO STRING;
    pInsertString : POINTER TO STRING;
    pDstString    : POINTER TO STRING;
    nDstSize      : UDINT;
    nLen          : UDINT;
    nPos          : UDINT;
END_VAR
```

pSrcString: Pointer to the STRING variable whose string is to be searched (input string)

pInsertString: Pointer to the STRING variable whose string is to replace the characters (input string)

pDstString: Pointer to the resulting STRING variable (output string)

nDstSize: Size of the resulting STRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

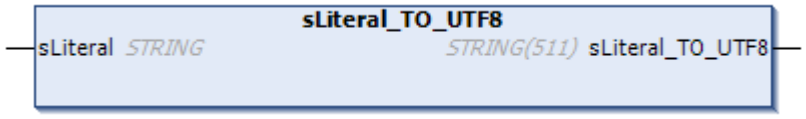
nLen: Number of characters to be replaced

nPos: Position of the character to be removed; the following characters are also to be removed (nPos = 1 = first character)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.35.0

4.2.17 sLiteral_TO_UTF8



This function converts any character string of the data type STRING into a character string in UTF-8 format. The function is particularly suitable for the assignment of literals.

When assigning literals to a UTF-8 STRING, the rules are:

- Literals that only use the ASCII character set can be directly assigned.
- Literals that use the STRING character set can be assigned by means of sLiteral_TO_UTF8().
- Literals that use the WSTRING character set can be assigned by means of wsLiteral_TO_UTF8() [▶ 177].

An empty string will be returned if the literal is longer than the possible output character string.

FUNCTION sLiteral_TO_UTF8 : STRING(511)

VAR_INPUT

```
VAR_IN_OUT CONSTANT
    sLiteral   : STRING;
END_VAR
```

sLiteral: STRING character string to be converted.

Examples

```
{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 €.');
```

Documentation for attribute 'TcEncoding' := 'UTF-8'

For more information about strings in UTF-8 format see also the documentation on the 'TcEncoding' attribute.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.34.0

4.2.18 STRING_TO_UTF8

The function converts any string of a variable of the data type STRING to a string in UTF-8 format.

The function returns

- TRUE if the conversion was possible.
- FALSE if the conversion was not possible due to the given character set.

If the input string is longer than the output string, the string will be truncated. The input string is too long to be coded to the output string. The memory requirement for the output string can be higher than that for the input string when converting to UTF-8.

The function stops the conversion after `Tc2_Utilities.Parameterlist.cMaxCharacters`. With appropriate parameterization an infinite loop can be avoided.

FUNCTION STRING_TO_UTF8 : BOOL

VAR_INPUT

```
VAR_INPUT
    pDstUTF8   : PVOID;
    pSrcSTRING : POINTER TO STRING;
    nDstSize   : UDINT;
END_VAR
```

pDstUTF8: Pointer to the string in UTF-8 format (output string)

pSrcSTRING: Pointer to the STRING variable to be converted (input string)

nDstSize: Size of the resulting variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

Documentation for attribute 'TcEncoding' := 'UTF-8'

For more information about strings in UTF-8 format see also the documentation on the 'TcEncoding' attribute.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.19 STRING_TO_WSTRING2

The function converts a variable of the data type STRING into a variable of the data type WSTRING and checks whether the input string is longer than the output string. In this case the string is truncated.

The function returns

- TRUE if the conversion of the complete string was possible.
- FALSE if the input string is longer than the output string and the result doesn't fit in the given output buffer. The memory requirement for the output string is higher than that for the input string. The string is then truncated.

The function stops the conversion after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION STRING_TO_WSTRING2 : BOOL

VAR_INPUT

```
VAR_INPUT
    pDstWSTRING : POINTER TO WSTRING;
    pSrcSTRING  : POINTER TO STRING;
    nDstSize    : UDINT;
END_VAR
```

pDstWSTRING: Pointer to the converted WSTRING variable (output string)

pSrcSTRING: Pointer to the STRING variable to be converted (input string)

nDstSize: Size of the resulting WSTRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.20 STRNCPY

The function copies the character string of a variable of the data type STRING and checks whether the character string was completely copied.

The function returns

- TRUE if it was possible to copy the complete character string (content of the source array).
- FALSE if the character string was truncated when copying. If the input string is longer than the output string, only the number of characters (including null termination) corresponding to the length of the output string will be copied.

FUNCTION STRNCPY : BOOL

VAR_INPUT

```
VAR_INPUT
    pDst     : POINTER TO STRING;
    pSrc     : POINTER TO STRING;
    nDstSize : UDINT;
END_VAR
```

pDst: Pointer to the copied STRING variable (input string)

pSrc: Pointer to the STRING variable to be copied (output string)

nDstSize: Size of the resulting STRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

VAR_OUTPUT

```
VAR_OUTPUT
    nSrcLen  : UDINT;
    nDstLen  : UDINT;
END_VAR
```

nSrcLen: Length of the specified STRING variable to be copied

nDstLen: Length of the copied WSTRING variable

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.21 UTF8_TO_STRING

The function converts a string in UTF-8 format (pointer variable of the data type PVOID) into a string (variable) of the data type STRING.

The function returns

- TRUE if the conversion was possible.
- FALSE if the conversion was not possible due to the given character set.

If the input string is longer than the output string, the string will be truncated. Unknown characters are skipped.

The function stops the conversion after `Tc2_Utilities.Parameterlist.cMaxCharacters`. With appropriate parameterization an infinite loop can be avoided.

FUNCTION UTF8_TO_STRING : BOOL

The return value is TRUE if the conversion was successful.

VAR_INPUT

```
VAR_INPUT
    pDstSTRING : POINTER TO STRING;
    pSrcUTF8   : PVOID;
    nDstSize   : UDINT;
END_VAR
```

pDstSTRING: Pointer to the converted STRING variable (output string)

pSrcUTF8: Pointer variable (input string)

nDstSize: Size of the resulting STRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

VAR_OUTPUT

```
VAR_OUTPUT
    nDstLen        : UDINT;
END_VAR
```

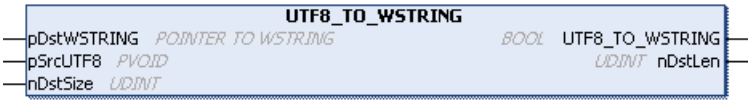
nDstLen: Actual length of the output string as number of characters.

> **Documentation for attribute 'TcEncoding' := 'UTF-8'**
>
> For more information about strings in UTF-8 format see also the documentation on the 'TcEncoding' attribute.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.22 UTF8_TO_WSTRING



The function converts a string in UTF-8 format into a string (variable) of the data type WSTRING.

The function returns

- TRUE if the conversion was possible.
- FALSE if the conversion was not possible due to the given character set.

If the input string is longer than the output string, the string will be truncated. The input string is too long to be coded to the output string. The memory requirement for the output string can be higher than that for the input string when converting to UTF-8. Unknown characters are skipped.

The function stops the conversion after `Tc2_Utilities.Parameterlist.cMaxCharacters`. With appropriate parameterization an infinite loop can be avoided.

FUNCTION UTF8_TO_WSTRING : BOOL

The return value is TRUE if the conversion was successful.

VAR_INPUT

```
VAR_INPUT
    pDstWSTRING : POINTER TO WSTRING;
    pSrcUTF8    : PVOID;
    nDstSize    : UDINT;
END_VAR
```

pDstWSTRING: Pointer to the converted WSTRING variable (output string)

pSrcUTF8: Pointer variable (input string)

nDstSize: Size of the resulting WSTRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

VAR_OUTPUT

```
VAR_OUTPUT
    nDstLen : UDINT;
END_VAR
```

nDstLen: Actual length of the output string as number of characters.

> **Documentation for attribute 'TcEncoding' := 'UTF-8'**
>
> For more information about strings in UTF-8 format see also the documentation on the 'TcEncoding' attribute.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.23 UTF8Len

The function returns the number of characters in a UTF-8 string.

If the string does not correspond to the UTF-8 format, the function returns the value 0.

In addition, the function checks whether all the characters are valid ASCII characters and outputs this via the `bASCII` output.

The function stops the verification after `Tc2_Utilities.Parameterlist.cMaxCharacters`. With appropriate parameterization an infinite loop can be avoided.

FUNCTION UTF8Len : UDINT

The return value returns the number of characters in the UTF-8 string.

VAR_INPUT

```
VAR_INPUT
    pUTF8 : PVOID;
END_VAR
```

pUTF8: pointer to the null-terminated UTF-8 string

VAR_OUTPUT

```
VAR_OUTPUT
    bASCII   : BOOL;
    nSize    : UDINT;
END_VAR
```

bASCII: TRUE if the UTF-8 characters are valid ASCII characters.

nSize: Size of the string in bytes (without zero termination). Depending on the characters, the size in bytes may be larger than the length of the string.

> **Documentation for attribute 'TcEncoding' := 'UTF-8'**
>
> For more information about strings in UTF-8 format see also the documentation on the 'TcEncoding' attribute.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.24 WCHAR_TO_CHAR

The function converts a variable of the data type WSTRING into a variable of the data type STRING (with null termination). The conversion is possible only if the WSTRING character corresponds to the STRING character. Otherwise no character is returned.

FUNCTION WCHAR_TO_CHAR : STRING(1)

VAR_INPUT

```
VAR_INPUT
    sTextIn : WSTRING(1);
END_VAR
```

sTextIn: WSTRING variable to be converted.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.25 WCONCAT2

The function concatenates two strings of the data type WSTRING of any length and checks whether the resulting string is longer than a specified output string. In this case the string is truncated.

The function returns

- TRUE if the concatenation was successful.
- FALSE if the resulting string is longer than the output string and doesn't fit in the given output buffer. The memory requirement for the resulting string is then larger than that for the output string. The string is then truncated.

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop

FUNCTION WCONCAT2 : BOOL

VAR_INPUT

```
VAR_INPUT
    pSrcWString1 : POINTER TO WSTRING;
    pSrcWString2 : POINTER TO WSTRING;
    pDstWString  : POINTER TO WSTRING;
    nDstSize     : UDINT;
END_VAR
```

pSrcWString1: Pointer to the first of the WSTRING variables to be concatenated (input string)

pSrcWString2: Pointer to the second of the WSTRING variables to be concatenated (input string)

pDstWString: Pointer to the resulting WSTRING variable following the concatenation (output string)

nDstSize: Size of the resulting WSTRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.26 WLEN2

The function returns the number of characters in a Unicode string of data type WSTRING (length of the WSTRING).

The function stops the checking of the input length after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION WLEN2 : UDINT

VAR_INPUT

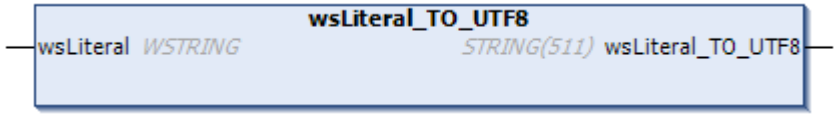
```
VAR_INPUT
    pWSTRING : POINTER TO WSTRING;
END_VAR
```

pWSTRING: pointer to the WSTRING variable

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.27 wsLiteral_TO_UTF8



This function converts any character string of the data type WSTRING into a character string in UTF-8 format. The function is particularly suitable for the assignment of literals.

When assigning literals to a UTF-8 STRING, the rules are:

- Literals that only use the ASCII character set can be directly assigned.
- Literals that use the STRING character set can be assigned by means of sLiteral_TO_UTF8() [▶ 170].
- Literals that use the WSTRING character set can be assigned by means of wsLiteral_TO_UTF8().

An empty string will be returned if the literal is longer than the possible output character string.

FUNCTION wsLiteral_TO_UTF8 : STRING(511)

VAR_INPUT

```
VAR_IN_OUT CONSTANT
    wsLiteral   : WSTRING;
END_VAR
```

sLiteral: WSTRING character string to be converted.

Examples

```
{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 €.");
```

Documentation for attribute 'TcEncoding' := 'UTF-8'

For more information about strings in UTF-8 format see also the documentation on the 'TcEncoding' attribute.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.34.0

4.2.28 WSTRING_TO_STRING2

The function converts a variable of the data type WSTRING into a variable of the data type STRING.

The function returns

- TRUE if the conversion of the complete string was possible.
- FALSE if the input string is longer than the output string and the result doesn't fit in the given output buffer.

Characters that cannot be converted are skipped during the conversion.

The function stops the conversion after `Parameterlist.cMaxCharacters` characters in order to avoid an infinite loop.

FUNCTION WSTRING_TO_STRING2 : BOOL

VAR_INPUT

```
VAR_INPUT
    pDstString  : POINTER TO STRING;
    pSrcWString : POINTER TO WSTRING;
    nDstSize    : UDINT;
END_VAR
```

pDstSTRING: Pointer to the converted STRING variable (output string)

pSrcWSTRING: Pointer to the WSTRING variable to be converted (input string)

nDstSize: Size of the resulting STRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.29 WSTRING_TO_UTF8

The function converts a string of a variable of the data type WSTRING into a string in UTF-8 format.

The function returns

- TRUE if the conversion was possible.
- FALSE if the conversion was not possible due to the given character set.

If the input string is longer than the output string, the string will be truncated. The input string is too long to be coded to the output string. (The memory requirement for the output string can be higher than that for the input character string when converting to UTF-8). Unknown characters are skipped.

The function stops the conversion after `Tc2_Utilities.Parameterlist.cMaxCharacters`. With appropriate parameterization an infinite loop can be avoided.

FUNCTION WSTRING_TO_UTF8 : BOOL

VAR_INPUT

```
VAR_INPUT
    pDstUTF8    : PVOID;
    pSrcWSTRING : POINTER TO WSTRING;
    nDstSize    : UDINT;
END_VAR
```

pDstUTF8: Pointer variable (output string)

pSrcWSTRING: Pointer to the WSTRING variable (input string)

nDstSize: Size of the resulting variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

Documentation for attribute 'TcEncoding' := 'UTF-8'

For more information about strings in UTF-8 format see also the documentation on the 'TcEncoding' attribute.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.2.30 WSTRNCPY

The function copies the string of a variable of the data type WSTRING and checks whether the string was completely copied.

The function returns

- TRUE if it was possible to copy the complete character string (content of the source array).
- FALSE if the character string was truncated when copying. If the input string is longer than the output string, only the number of characters (including null termination) corresponding to the length of the output string will be copied.

FUNCTION WSTRNCPY : BOOL

VAR_INPUT

```
VAR_INPUT
    pDst     : POINTER TO WSTRING;
    pSrc     : POINTER TO WSTRING;
    nDstSize : UDINT;
END_VAR
```

pDst: Pointer to the copied WSTRING variable (input string)

pSrc: Pointer to the WSTRING variable to be copied (output string)

nDstSize: Size of the resulting WSTRING variable (output string) in bytes. The operator SIZEOF() can be used for the assignment.

VAR_OUTPUT

```
VAR_OUTPUT
    nSrcLen  : UDINT;
    nDstLen  : UDINT;
END_VAR
```

nSrcLen: Length of the specified WSTRING variable to be copied

nDstLen: Length of the copied WSTRING variable

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.3 Byte order converting functions

4.3.1 Host Byte Order / Network Byte Order

The byte order is fixed in network protocols. This is called the Network Byte Order. The natural byte order in the TwinCAT system is called the Host Byte Order. In most cases, the required Network Byte Order corresponds to the Big Endian Format (MOTOROLA). However, the TwinCAT PLC system uses the Little Endian Format (Intel). So that the error-free exchange of data between the TwinCAT PLC system and a different platform can take place, the byte order in the application program must be converted accordingly.

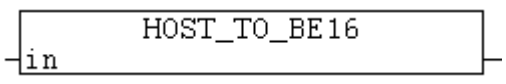
Data that is to be transmitted via a network protocol from the TwinCAT system (host) to an external system can be converted to the network format using the following functions:

- HOST_TO_BE16 [▶ 180]
- HOST_TO_BE32 [▶ 181]
- HOST_TO_BE64 [▶ 181]
- HOST_TO_BE64EX [▶ 182]
- HOST_TO_BE128 [▶ 182]

Conversely, the received network data (external system) can be converted to the host format (TwinCAT system) using the following functions:

- BE16_TO_HOST [▶ 183]
- BE32_TO_HOST [▶ 183]
- BE64_TO_HOST [▶ 183]
- BE64_TO_HOSTEX [▶ 184]
- BE128_TO_HOST [▶ 184]

4.3.2 HOST_TO_BE16



The function performs a host-to-network conversion of a 16-bit number. See also under: Byte Order [▶ 180].

FUNCTION HOST_TO_BE16: WORD

VAR_INPUT

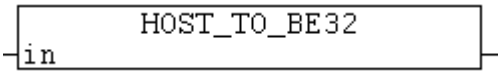
```
VAR_INPUT
    in : WORD;
END_VAR
```

in: Number to be converted.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.3 HOST_TO_BE32



The function performs a host-to-network conversion of a 32-bit number. See also under: Byte Order [▶ 180].

FUNCTION HOST_TO_BE32: DWORD

VAR_INPUT

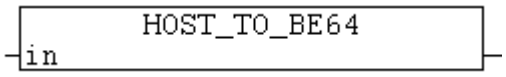
```
VAR_INPUT
    in : DWORD;
END_VAR
```

in: Number to be converted.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.4 HOST_TO_BE64



The function performs a host-to-network conversion of a 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]). See also under: Byte Order [▶ 180].

FUNCTION HOST_TO_BE64: T_ULARGE_INTEGER

VAR_INPUT

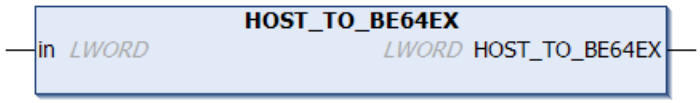
```
VAR_INPUT
    in : T_ULARGE_INTEGER;
END_VAR
```

in: Number to be converted (Type:T_ULARGE_INTEGER [▶ 320]).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.5 HOST_TO_BE64EX



The function performs a host-to-network conversion of a 64-bit number (“native” type: LWORD). See also under: Byte Order [▶ 180].

FUNCTION HOST_TO_BE64EX: LWORD

VAR_INPUT

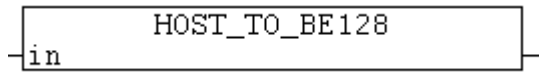
```
VAR_INPUT
    in : LWORD;
END_VAR
```

in: Number to be converted.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.6 HOST_TO_BE128



The function performs a host-to-network conversion of a 128-bit number (“legacy” type: T_UHUGE_INTEGER [▶ 320]). See also under: Byte Order [▶ 180].

FUNCTION HOST_TO_BE128: T_UHUGE_INTEGER

VAR_INPUT

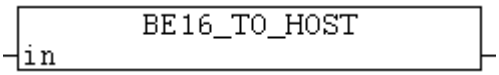
```
VAR_INPUT
    in : T_UHUGE_INTEGER;
END_VAR
```

in: Number to be converted (Type:T_UHUGE_INTEGER [▶ 320]).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.7 BE16_TO_HOST



The function performs a network-to-host conversion of a 16-bit number. See also under: Byte Order [▶ 180].

FUNCTION BE16_TO_HOST: WORD

VAR_INPUT

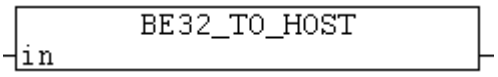
```
VAR_INPUT
    in : WORD;
END_VAR
```

in: Number to be converted.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.8 BE32_TO_HOST



The function performs a network-to-host conversion of a 32-bit number. See also under: Byte Order [▶ 180].

FUNCTION BE32_TO_HOST: DWORD

VAR_INPUT

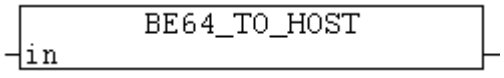
```
VAR_INPUT
    in : DWORD;
END_VAR
```

in: Number to be converted.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.9 BE64_TO_HOST



The function performs a network-to-host conversion of a 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]). See also under: Byte Order [▶ 180].

FUNCTION BE64_TO_HOST: T_ULARGE_INTEGER

VAR_INPUT

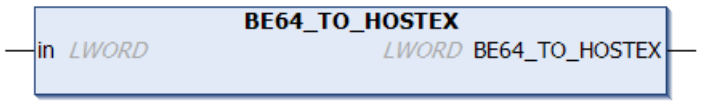
```
VAR_INPUT
    in : T_ULARGE_INTEGER;
END_VAR
```

in: Number to be converted (Type:T_ULARGE_INTEGER [▶ 320]).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.10 BE64_TO_HOSTEX



The function performs a network-to-host conversion of a 64-bit number (“native” type: LWORD). See also under: Byte Order [▶ 180].

FUNCTION BE64_TO_HOSTEX: LWORD

VAR_INPUT

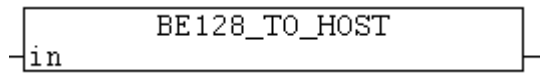
```
VAR_INPUT
    in : LWORD;
END_VAR
```

in: Number to be converted.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.3.11 BE128_TO_HOST



The function performs a network-to-host conversion of a 128-bit number (“legacy” type: T_UHUGE_INTEGER [▶ 320]). See also under: Byte Order [▶ 180].

FUNCTION BE128_TO_HOST: T_UHUGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    in : T_UHUGE_INTEGER;
END_VAR
```

in: Number to be converted (Type T_UHUGE_INTEGER [▶ 320]).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4 FLOAT functions

4.4.1 BOOL_TO_FLOAT

FUNCTION BOOL_TO_FLOAT: LREAL

VAR_INPUT

```
VAR_INPUT
    in    : BOOL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.2 DINT_TO_FLOAT

FUNCTION DINT_TO_FLOAT: FLOAT

VAR_INPUT

```
VAR_INPUT
    in    : DINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.3 FLOAT_TO_BOOL

FUNCTION FLOAT_TO_BOOL: BOOL

VAR_INPUT

```
VAR_INPUT
    in    : LREAL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.4 FLOAT_TO_DINT

FUNCTION FLOAT_TO_DINT: DINT

VAR_INPUT

```
VAR_INPUT
    in        : FLOAT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.5 FLOAT_TO_INT

FUNCTION FLOAT_TO_INT: INT

VAR_INPUT

```
VAR_INPUT
    in            : FLOAT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.6 FLOAT_TO_SINT

FUNCTION FLOAT_TO_SINT: SINT

VAR_INPUT

```
VAR_INPUT
    in            : FLOAT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.7 FLOAT_TO_STRING

FUNCTION FLOAT_TO_STRING: STRING

VAR_INPUT

```
VAR_INPUT
    in            : FLOAT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.8 FLOAT_TO_TIME

FUNCTION FLOAT_TO_TIME: TIME

VAR_INPUT

```
VAR_INPUT
    in          : FLOAT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.9 FLOAT_TO_UDINT

FUNCTION FLOAT_TO_UDINT: UDINT

VAR_INPUT

```
VAR_INPUT
    in          : FLOAT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.10 FLOAT_TO_UINT

FUNCTION FLOAT_TO_UINT: UINT

VAR_INPUT

```
VAR_INPUT
    in          : FLOAT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.11 INT_TO_FLOAT

FUNCTION INT_TO_FLOAT: FLOAT

VAR_INPUT

```
VAR_INPUT
    in    : INT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.12 SINT_TO_FLOAT

FUNCTION SINT_TO_FLOAT: FLOAT

VAR_INPUT

```
VAR_INPUT
    in    : SINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.13 TIME_TO_FLOAT

FUNCTION TIME_TO_FLOAT: FLOAT

VAR_INPUT

```
VAR_INPUT
    in:       TIME;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.14 UDINT_TO_FLOAT

FUNCTION UDINT_TO_FLOAT: FLOAT

VAR_INPUT

```
VAR_INPUT
    in    : UDINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.15 UINT_TO_FLOAT

FUNCTION UINT_TO_FLOAT: FLOAT

VAR_INPUT

```
VAR_INPUT
    in    : UINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.4.16 LrealIsFinite

FUNCTION LrealIsFinite : BOOL

VAR_INPUT

```
VAR_INPUT
    x        : REFERENCE TO LREAL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4020	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.16.0

4.4.17 LrealIsNaN

This function tests whether a value is NaN (Not a Number). If the return value is TRUE, the value is NaN.

```
FUNCTION LrealIsNaN : BOOL
VAR_INPUT
    x        : REFERENCE TO LREAL;
END_VAR
```

The following points rank among the main features of NaN values:

- All arithmetic operations that use NaN as input data return NaN as the result.
- All relational operators =, !=, > < >= <= always return the value False if at least one of the operands is NaN.
- The standard C function `isnan()` or `_isnan()` or the PLC function LrealIsNaN() [▶ 189] (Tc2_Utilities library) returns the value True if the argument has the value NaN.
- The expression `isnan(a)` is equivalent to the expression `!(a == a)` or `NOT(a = a)`.

The fact that NaN values reproduce themselves when used in further calculations is advantageous in that invalid values cannot be overlooked.

⚠ CAUTION

Software malfunctions

NaN values may only be used in PLC libraries, in particular as control values in functions for Motion Control and for drive control, if they are expressly approved! Otherwise NaN values can lead to potentially dangerous malfunctions of the software concerned!

⚠ CAUTION

Floating point exceptions

If NaNs are to be used and processed in the application, the FP exceptions must be switched off. Otherwise, comparisons with NaN can lead to an exception, which will cause a stop of the runtime and possible machine damage.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4020	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.16.0

4.4.18 ReallsFinite

FUNCTION ReallsFinite : BOOL

VAR_INPUT

```
VAR_INPUT
    x        : REFERENCE TO REAL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024.32	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.50.0

4.4.19 ReallsNaN

This function tests whether a value is NaN (Not a Number). If the return value is TRUE, the value is NaN.

```
FUNCTION RealIsNaN : BOOL
VAR_INPUT
    x        : REFERENCE TO REAL;
END_VAR
```

The following points rank among the main features of NaN values:

- All arithmetic operations that use NaN as input data return NaN as the result.
- All relational operators =, !=, > < >= <= always return the value False if at least one of the operands is NaN.
- The standard C function `isnan()` or `_isnan()` or the PLC function LrealIsNaN() [▶ 189] (Tc2_Utilities library) returns the value True if the argument has the value NaN.
- The expression `isnan(a)` is equivalent to the expression `!(a == a)` or `NOT(a = a)`.

The fact that NaN values reproduce themselves when used in further calculations is advantageous in that invalid values cannot be overlooked.

⚠ CAUTION

Software malfunctions

NaN values may only be used in PLC libraries, in particular as control values in functions for Motion Control and for drive control, if they are expressly approved! Otherwise NaN values can lead to potentially dangerous malfunctions of the software concerned!

⚠ CAUTION

Floating point exceptions

If NaNs are to be used and processed in the application, the FP exceptions must be switched off. Otherwise, comparisons with NaN can lead to an exception, which will cause a stop of the runtime and possible machine damage.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024.32	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.50.0

4.5 LCOMPLEX functions

4.5.1 LcomplexIsNaN

The function returns TRUE, when the argument of the type LCOMPLEX has an undefined value (NaN).

FUNCTION LcomplexIsNaN : BOOL

VAR_INPUT

```
VAR_INPUT
    Z   : REFERENCE TO LCOMPLEX;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4020	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >=3.3.16.0

4.5.2 LcomplexAbs

The function returns the absolute value of the complex number transferred.

FUNCTION LcomplexAbs : LREAL

VAR_INPUT

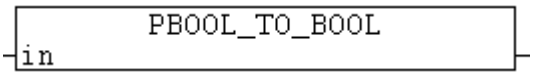
```
VAR_INPUT
    Z   :  LCOMPLEX;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.21.0

4.6 P[TYPE]_TO_[TYPE] converting functions

4.6.1 PBOOL_TO_BOOL



The function returns the content of a BOOL pointer variable.

FUNCTION PBOOL_TO_BOOL: BOOL

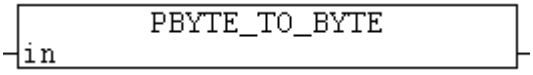
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO BOOL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.2 PBYTE_TO_BYTE



The function returns the content of a BYTE pointer variable.

FUNCTION PBYTE_TO_BYTE: BYTE

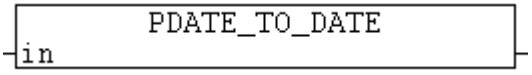
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO BYTE;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.3 PDATE_TO_DATE



The function returns the content of a DATE pointer variable.

FUNCTION PDATE_TO_DATE: DATE

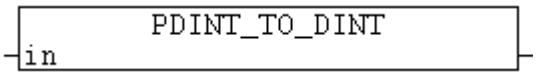
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO DATE;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.4 PDINT_TO_DINT



The function returns the content of a DINT pointer variable.

FUNCTION PDINT_TO_DINT: DINT

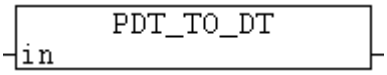
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO DINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.5 PDT_TO_DT



The function returns the content of a DT pointer variable.

FUNCTION PDT_TO_DT: DT

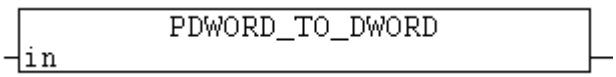
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO DT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.6 PDWORD_TO_DWORD



The function returns the content of a DWORD pointer variable.

FUNCTION PDWORD_TO_DWORD: DWORD

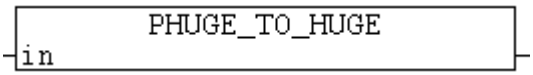
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO DWORD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.7 PHUGE_TO_HUGE



The function returns the content of a T_HUGE_INTEGER [▶ 319] pointer variable.

FUNCTION PHUGE_TO_HUGE: T_HUGE_INTEGER

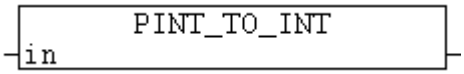
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO T_HUGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.8 PINT_TO_INT



The function returns the content of an INT pointer variable.

FUNCTION PINT_TO_INT: INT

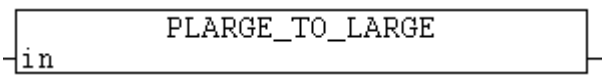
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO INT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.9 PLARGE_TO_LARGE



The function returns the content of a T_LARGE_INTEGER [▶ 320] pointer variable.

FUNCTION PLARGE_TO_LARGE: T_LARGE_INTEGER

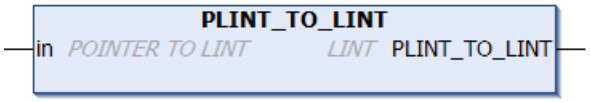
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO T_LARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.10 PLINT_TO_LINT



The function returns the content of a LINT pointer variable.

FUNCTION PLINT_TO_LINT: LINT

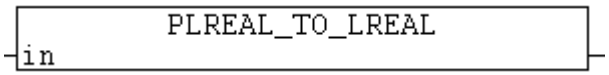
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO LINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.11 PLREAL_TO_LREAL



The function returns the content of a LREAL pointer variable.

FUNCTION PLREAL_TO_LREAL: LREAL

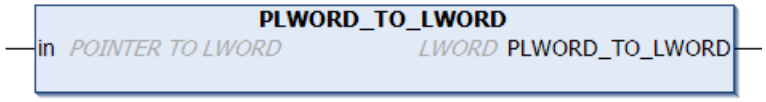
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO LREAL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.12 PLWORD_TO_LWORD



The function returns the content of a LWORD pointer variable.

FUNCTION PLWORD_TO_LWORD: LWORD

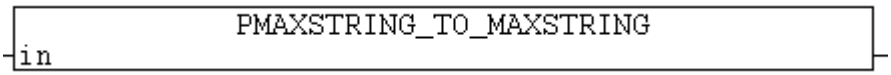
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO LWORD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.13 PMAXSTRING_TO_MAXSTRING



The function returns the content of a T_MaxString pointer variable.

FUNCTION PMAXSTRING_TO_MAXSTRING: T_MaxString

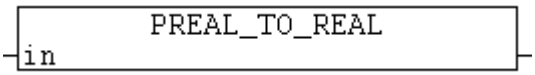
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO T_MaxString;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.14 PREAL_TO_REAL



The function returns the content of a REAL pointer variable.

FUNCTION PREAL_TO_REAL: REAL

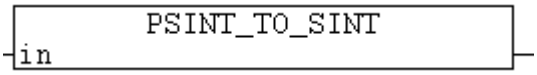
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO REAL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.15 PSINT_TO_SINT



The function returns the content of a SINT pointer variable.

FUNCTION PSINT_TO_SINT: SINT

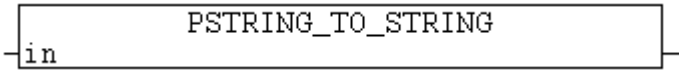
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO SINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.16 PSTRING_TO_STRING



The function returns the content of a STRING pointer variable.

FUNCTION PSTRING_TO_STRING: STRING

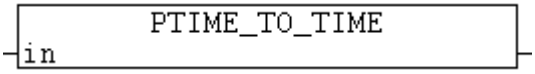
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO STRING;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.17 PTIME_TO_TIME



The function returns the content of a TIME pointer variable.

FUNCTION PTIME_TO_TIME: TIME

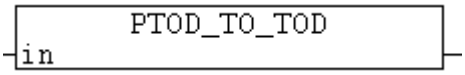
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO TIME;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.18 PTOD_TO_TOD



The function returns the content of a TOD pointer variable.

FUNCTION PTOD_TO_TOD: TOD

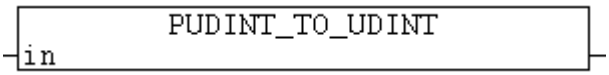
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO TOD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.19 PUDINT_TO_UDINT



The function returns the content of a UDINT pointer variable.

FUNCTION PUDINT_TO_UDINT: UDINT

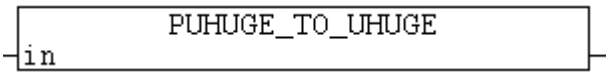
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO UDINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.20 PUHUGE_TO_UHUGE



The function returns the content of a T_UHUGE_INTEGER [▶ 320] pointer variable.

FUNCTION PUHUGE_TO_UHUGE: T_UHUGE_INTEGER

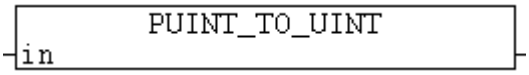
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO T_UHUGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.21 PUINT_TO_UINT



The function returns the content of a UINT pointer variable.

FUNCTION PUINT_TO_UINT: UINT

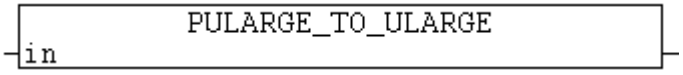
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO UINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.22 PULARGE_TO_ULARGE



The function returns the content of a T_ULARGE_INTEGER [▶ 320] pointer variable.

FUNCTION PULARGE_TO_ULARGE: T_ULARGE_INTEGER

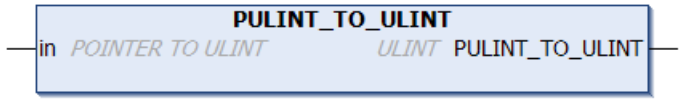
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.23 PULINT_TO_ULINT



The function returns the content of a ULINT pointer variable.

FUNCTION PULINT_TO_ULINT: ULINT

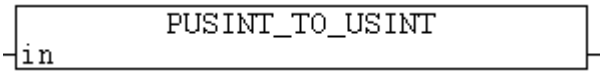
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO ULINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.24 PUSINT_TO_USINT



The function returns the content of a USINT pointer variable.

FUNCTION PUSINT_TO_USINT: USINT

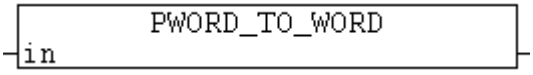
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO USINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.25 PWORD_TO_WORD



The function returns the content of a WORD pointer variable.

FUNCTION PWORD_TO_WORD: WORD

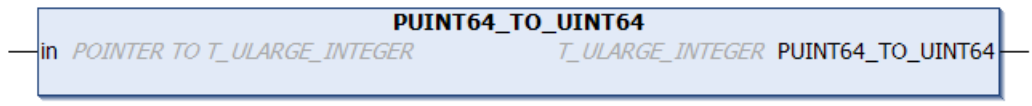
VAR_INPUT

```
VAR_INPUT
    in : POINTER TO WORD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.6.26 PUINT64_TO_UINT64



The function PUINT64_TO_UINT64 returns the content of a T_ULARGE_INTEGER [▶ 320] pointer variable.

FUNCTION PUINT64_TO_UINT64: T_ULARGE_INTEGER

VAR_INPUT

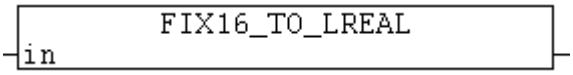
```
VAR_INPUT
    in    :  POINTER TO T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7 16 bit fixed point number functions (signed)

4.7.1 FIX16_TO_LREAL



Converts a signed 16-bit fixed-point number to a floating-point number of type: LREAL.

FUNCTION FIX16_TO_LREAL: LREAL

VAR_INPUT

```
VAR_INPUT
    in : T_FIX16;
END_VAR
```

in: The fixed-point number to be converted (type: T_FIX16 [▶ 317]).

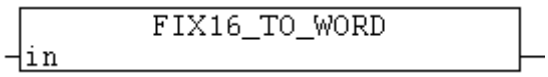
Example:

See function description: LREAL_TO_FIX16 [▶ 205].

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7.2 FIX16_TO_WORD



This function converts a 16-bit fixed-point number to a WORD variable (the WORD variable contains the digits and decimal places for the fixed-point number).

FUNCTION FIX16_TO_WORD: WORD

VAR_INPUT

```
VAR_INPUT
    in : T_FIX16;
END_VAR
```

in: fixed point number to convert (type: T_FIX16 [▶ 317]).

Example:

```
PROGRAM FIX_TO_WORD
VAR
    fp16 : WORD;
END_VAR
```

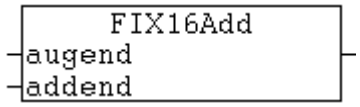
```
fp16 := FIX16_TO_WORD(LREAL_TO_FIX16(12.5, 8));
```

The value of the *fp16* variable is: 2#0000110010000000.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7.3 FIX16Add



This function adds two signed 16-bit fixed-point numbers. The numbers do not have to have the same resolution (number of decimal places). The resolution of the number with the higher number of decimal places is reduced before the addition, i.e. the decimal places of the number with the higher resolution are truncated. The result of the addition is a signed 16-bit fixed-point number.

FUNCTION FIX16Add: T_FIX16 [▶ 317]

VAR_INPUT

```
VAR_INPUT
    augend : T_FIX16;
    addend : T_FIX16;
END_VAR
```

augend: The first summand (type: T_FIX16 [▶ 317]).

addend: The second summand (type: T_FIX16 [▶ 317]).

Example:

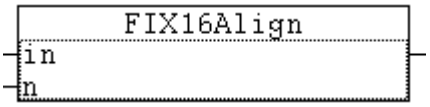
```
PROGRAM FIXADD
VAR
    a, b   : T_FIX16;
    result : LREAL;
END_VAR
```

```
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 *)
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7.4 FIX16Align



This function can be used to change the resolution (number of decimal places) of a signed 16-bit fixed-point number. The function supplies the new fixed-point number as return parameter.

FUNCTION FIX16Align: T_FIX16 [▶ 317]

VAR_INPUT

```
VAR_INPUT
    in : T_FIX16;
    n  : BYTE(0..15);
END_VAR
```

in: Fixed-point number whose resolution is to be modified (type: T_FIX16 [▶ 317]).

n: The new number of decimal places.

Example:

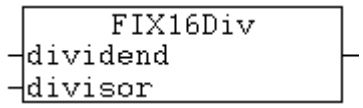
```
PROGRAM FIXALIGN
VAR
    q8, q4 : T_FIX16;
    result : LREAL;
END_VAR
```

```
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 *)
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7.5 FIX16Div



This function divides two signed 16-bit fixed-point numbers. The numbers do not have to have the same resolution (number of decimal places). The resolution of the number with the higher number of decimal places is reduced before the division. i.e. the decimal places of the number with the higher resolution are truncated. The result of the division is a signed 16-bit fixed-point number.

FUNCTION FIX16Div: T_FIX16 [▶ 317]

VAR_INPUT

```
VAR_INPUT
    dividend : T_FIX16;
    divisor  : T_FIX16;
END_VAR
```

dividend: Number that is divided (type: T_FIX16 [▶ 317]).

divisor: Number used for the division (type: T_FIX16 [▶ 317]).

Example:

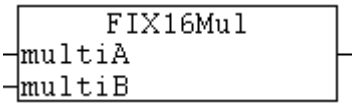
```
PROGRAM FIXDIV
VAR
    a, b   : T_FIX16;
    result : LREAL;
END_VAR
```

```
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 *)
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7.6 FIX16Mul



This function multiplies two signed 16-bit fixed-point numbers. The numbers do not have to have the same resolution (number of decimal places). The resolution of the number with the higher number of decimal places is reduced before the multiplication, i.e. the decimal places of the number with the higher resolution are truncated. The result of the multiplication is a signed 16-bit fixed-point number.

FUNCTION FIX16Mul: T_FIX16 [▶ 317]

VAR_INPUT

```
VAR_INPUT
    multiA : T_FIX16;
    multiB : T_FIX16;
END_VAR
```

multiA: The first multiplier (type: T_FIX16 [▶ 317]).

multiB: The second multiplier (type: T_FIX16 [▶ 317]).

Example:

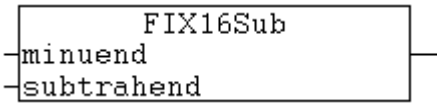
```
PROGRAM FIXMUL
VAR
    a, b   : T_FIX16;
    result : LREAL;
END_VAR
```

```
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 *)
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7.7 FIX16Sub



This function subtracts two signed 16-bit fixed-point numbers. The numbers do not have to have the same resolution (number of decimal places). The resolution of the number with the higher number of decimal places is reduced before the subtraction, i.e. the decimal places of the number with the higher resolution are truncated. The result of the subtraction is a signed 16-bit fixed-point number.

FUNCTION FIX16Sub: T_FIX16 [▶ 317]

VAR_INPUT

```
VAR_INPUT
    minuend    : T_FIX16;
    subtrahend : T_FIX16;
END_VAR
```

minuend: Number from which a value is subtracted (type: T_FIX16 [▶ 317]).

subtrahend: Number that is subtracted (type: T_FIX16 [▶ 317]).

Example:

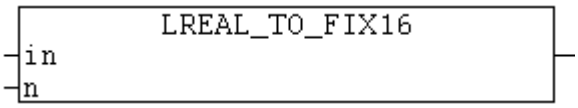
```
PROGRAM FIXSUB
VAR
    a, b   : T_FIX16;
    result : LREAL;
END_VAR
```

```
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 *)
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7.8 LREAL_TO_FIX16



Converts a floating-point number of type: LREAL to a signed 16 bit fixed-point number with the required number of decimal places.

FUNCTION LREAL_TO_FIX16: T_FIX16 [▶ 317]

VAR_INPUT

```
VAR_INPUT
    in : LREAL;
    n  : WORD(0..15) := 15;
END_VAR
```

in: The LREAL number to be converted.

n: Number of required decimal places.

Example:

In the following example several constants are converted to fixed-point numbers. The number of decimal places can be specified for the conversion. Please note that (similar to the conversion of floating-point numbers) rounding errors may occur (q2 and q15 in our example).

```
PROGRAM TEST
VAR
    q2, q4, q8, q12, q15 : T_FIX16;
    r2, r4, r8, r12, r15 : LREAL;
END_VAR
```

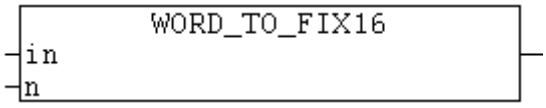
```
q2  := LREAL_TO_FIX16( 0.6, 2 );
q4  := LREAL_TO_FIX16( -0.25, 4 );
q8  := LREAL_TO_FIX16( -0.75, 8 );
q12 := LREAL_TO_FIX16( 2.30078125, 12 );
q15 := LREAL_TO_FIX16( 0.6, 15 );
```

```
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 *)
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.7.9 WORD_TO_FIX16



The function converts a WORD variable to a 16-bit fixed-point number (the WORD variable contains the coded digits and decimal places for the fixed-point number).

FUNCTION WORD_TO_FIX16: T_FIX16 [▶ 317]

VAR_INPUT

```
VAR_INPUT
    in : WORD;(* 16 bit fixed point number *)
    n  : WORD(0..15);(* number of fractional bits *)
END_VAR
```

Example:

```
PROGRAM WORD_TO_FIX
VAR
    double : LREAL;
END_VAR
```

```
double := FIX16_TO_LREAL(WORD_TO_FIX16(2#0000110010000000, 8));
```

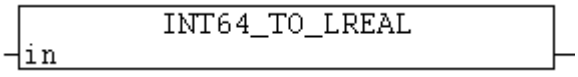
The value of the *double* variable is: 12.5

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8 64 bit functions (signed)

4.8.1 INT64_TO_LREAL



The function converts a TwinCAT 2 signed 64-bit number (“legacy” type: T_LARGE_INTEGER [▶ 320]) into a floating-point number of type LREAL.

FUNCTION INT64_TO_LREAL: LREAL

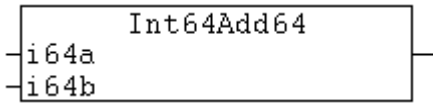
VAR_INPUT

```
VAR_INPUT
    in : T_LARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.2 Int64Add64



The function adds two TwinCAT 2 signed 64-bit numbers (“legacy” type: T_LARGE_INTEGER [▶ 320]). The result is a signed 64-bit number.

FUNCTION Int64Add64: T_LARGE_INTEGER

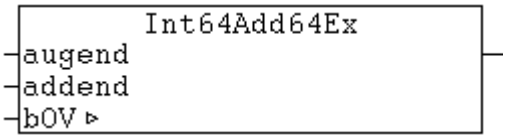
VAR_INPUT

```
VAR_INPUT
    i64a : T_LARGE_INTEGER;
    i64b : T_LARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.3 Int64Add64Ex



The function adds two TwinCAT 2 signed 64-bit numbers (“legacy” type: T_LARGE_INTEGER [▶ 320]). The result is a signed 64-bit number.

FUNCTION Int64Add64Ex: T_LARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    augend : T_LARGE_INTEGER;
    addend : T_LARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

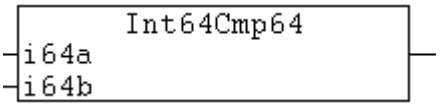
```
VAR_IN_OUT
    bOV    : BOOL;(* TRUE => arithmetic overflow, FALSE => no overflow *)
END_VAR
```

bOV: Arithmetic overflow. TRUE => overflow, FALSE => no overflow.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.4 Int64Cmp64



The function compares two TwinCAT 2 signed 64-bit numbers (“legacy” type: T_LARGE_INTEGER [▶ 320]).

FUNCTION Int64Cmp64: DINT

VAR_INPUT

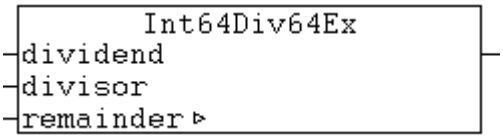
```
VAR_INPUT
    i64a : T_LARGE_INTEGER;
    i64b : T_LARGE_INTEGER;
END_VAR
```

Return parameter	Description
-1	i64a less than *i64b*
0	i64a identical to *i64b*
1	i64a greater than *i64b*

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.5 Int64Div64Ex



The function divides two TwinCAT 2 signed 64-bit numbers (“legacy” type: T_LARGE_INTEGER [▶ 320]). The result is a signed 64-bit number.

FUNCTION Int64Div64Ex: T_LARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    dividend  : T_LARGE_INTEGER;
    divisor   : T_LARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

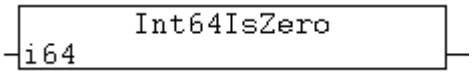
```
VAR_IN_OUT
    remainder : T_LARGE_INTEGER;
END_VAR
```

remainder: Residual.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.6 Int64IsZero



The function returns TRUE if the TwinCAT 2 signed 64-bit number (“legacy” type: T_LARGE_INTEGER [▶ 320]) has the value zero.

FUNCTION Int64isZero: BOOL

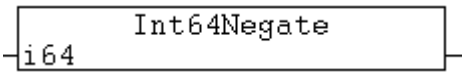
VAR_INPUT

```
VAR_INPUT
    i64 : T_LARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.7 Int64Negate



The function negates a TwinCAT 2 signed 64-bit number (“legacy” type: T_LARGE_INTEGER [▶ 320]).

FUNCTION Int64Negate: T_LARGE_INTEGER

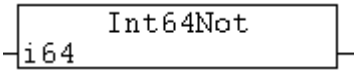
VAR_INPUT

```
VAR_INPUT
    i64 : T_LARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.8 Int64Not



Bitwise NOT of a TwinCAT 2 signed 64-bit number (“legacy” type: T_LARGE_INTEGER [▶ 320]). The result is a signed 64-bit number.

FUNCTION Int64Not: T_LARGE_INTEGER

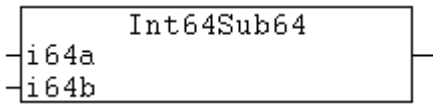
VAR_INPUT

```
VAR_INPUT
    i64 : T_LARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.9 Int64Sub64



The function subtracts two TwinCAT 2 signed 64-bit numbers (“legacy” type: T_LARGE_INTEGER [▶ 320]). The result is a signed 64-bit number.

FUNCTION Int64Sub64: T_LARGE_INTEGER

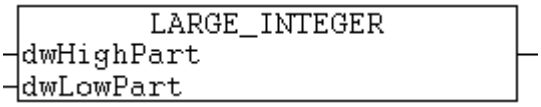
VAR_INPUT

```
VAR_INPUT
    i64a : T_LARGE_INTEGER;
    i64b : T_LARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.10 LARGE_INTEGER



The function initializes a TwinCAT 2 signed 64-bit number (“legacy” type: T_LARGE_INTEGER [▶ 320]).

FUNCTION LARGE_INTEGER: T_LARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    dwHighPart : DWORD;
    dwLowPart  : DWORD;
END_VAR
```

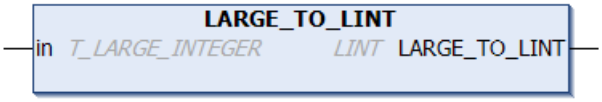
dwHighPart: Upper 32 bits.

dwLowPart: Lower 32 bits.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.11 LARGE_TO_LINT



The function converts a TwinCAT 2 signed 64-bit number (“legacy” type: T_LARGE_INTEGER [▶ 320]) into a TwinCAT 3 signed 64-bit number (“native” type).

FUNCTION LARGE_TO_LINT: LINT

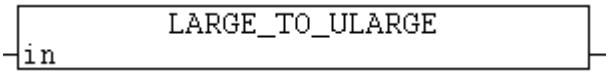
VAR_INPUT

```
VAR_INPUT
    in : T_LARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.12 LARGE_TO_ULARGE



The function converts a TwinCAT 2 signed 64-bit number (“legacy” type: T_LARGE_INTEGER [▶ 320]) into a TwinCAT 2 unsigned 64-bit number (“native” type: T_ULARGE_INTEGER [▶ 320]).

FUNCTION LARGE_TO_ULARGE: T_ULARGE_INTEGER

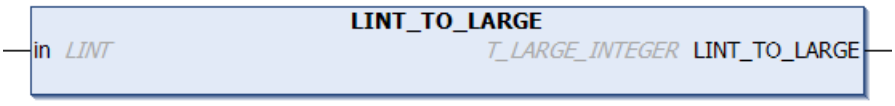
VAR_INPUT

```
VAR_INPUT
    in : T_LARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.13 LINT_TO_LARGE



The function converts a TwinCAT 3 signed 64-bit number (“native” type) into a TwinCAT 2 signed 64-bit number (“legacy” type: T_LARGE_INTEGER [▶ 320]).

FUNCTION LINT_TO_LARGE: T_LARGE_INTEGER

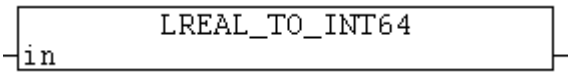
VAR_INPUT

```
VAR_INPUT
    in : LINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.14 LREAL_TO_INT64



The function converts an LREAL number into a TwinCAT 2 signed 64-bit number (“legacy” type: T_LARGE_INTEGER [▶ 320]).

FUNCTION LREAL_TO_INT64: T_LARGE_INTEGER

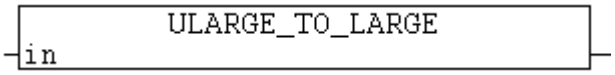
VAR_INPUT

```
VAR_INPUT
    in : LREAL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.8.15 ULARGE_TO_LARGE



The function converts a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]) into a TwinCAT 2 signed 64-bit number (“native” type: T_LARGE_INTEGER [▶ 320]).

FUNCTION ULARGE_TO_LARGE: T_LARGE_INTEGER

VAR_INPUT

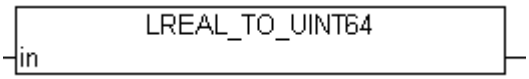
```
VAR_INPUT
    in : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9 64 bit integer functions (unsigned)

4.9.1 LREAL_TO_UINT64



The function converts a LREAL number into a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]).

FUNCTION LREAL_TO_UINT64: T_ULARGE_INTEGER

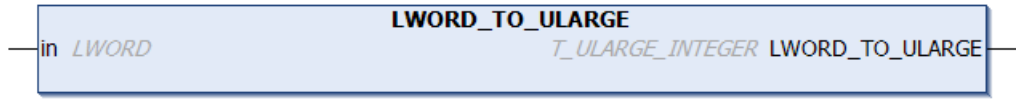
VAR_INPUT

```
VAR_INPUT
    in : LREAL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.2 LWORD_TO_ULARGE



The function converts a TwinCAT 3 unsigned 64 bit number (“native” type) into a TwinCAT 2 unsigned 64 bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]).

FUNCTION LWORD_TO_ULARGE: T_ULARGE_INTEGER

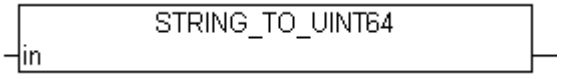
VAR_INPUT

```
VAR_INPUT
    in : LWORD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.3 STRING_TO_UINT64



The function converts a string into a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]).

FUNCTION STRING_TO_UINT64: T_ULARGE_INTEGER

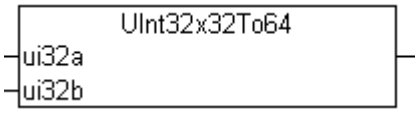
VAR_INPUT

```
VAR_INPUT
    in : STRING(21);
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.4 UInt32x32To64



The function multiplies two unsigned 32-bit numbers. The result is a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]).

FUNCTION UInt32x32To64: T_ULARGE_INTEGER

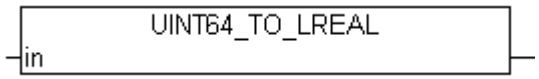
VAR_INPUT

```
VAR_INPUT
    ui32a : DWORD;
    ui32b : DWORD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.5 UINT64_TO_LREAL



The function converts a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]) into a floating-point number of type LREAL.

FUNCTION UINT64_TO_LREAL: LREAL

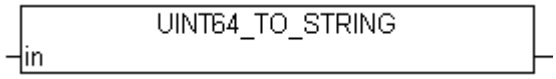
VAR_INPUT

```
VAR_INPUT
    in : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.6 UINT64_TO_STRING



The function converts a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]) into string.

FUNCTION UINT64_TO_STRING: STRING(21)

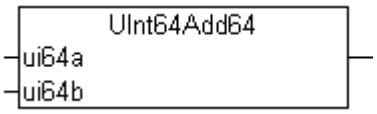
VAR_INPUT

```
VAR_INPUT
    in : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.7 UInt64Add64



The function adds two TwinCAT 2 unsigned 64-bit numbers (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Add64: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.8 UInt64Add64Ex

The function adds two TwinCAT 2 unsigned 64-bit numbers (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Add64Ex: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    augend : T_ULARGE_INTEGER;
    addend : T_ULARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

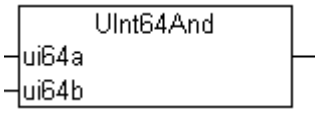
```
VAR_IN_OUT
    bOV    : BOOL;(* TRUE => arithmetic overflow, FALSE => no overflow *)
END_VAR
```

bOV: Arithmetic overflow. TRUE => overflow, FALSE => no overflow.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.9 UInt64And



Bitwise AND of two TwinCAT 2 unsigned 64-bit numbers (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64And: T_ULARGE_INTEGER

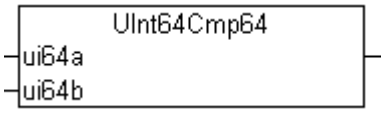
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.10 UInt64Cmp64



The function compares two TwinCAT 2 unsigned 64-bit numbers (“legacy” type: T_ULARGE_INTEGER [▶ 320]).

FUNCTION UInt64Cmp64: DINT

VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Return parameter	Description
-1	ui64a less than *ui64b*
0	ui64a identical to *ui64b*
1	ui64a greater than *ui64b*

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.11 UInt64Div16Ex

The function divides a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]) by a 16-bit unsigned number. The result is an unsigned 64-bit number.

FUNCTION UInt64Div16Ex: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    dividend : T_ULARGE_INTEGER;
    divisor  : WORD;
END_VAR
```

VAR_IN_OUT

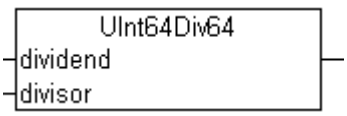
```
VAR_IN_OUT
    remainder : T_ULARGE_INTEGER;
END_VAR
```

remainder: Residual.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.12 UInt64Div64



The function divides two TwinCAT 2 unsigned 64-bit numbers (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Div64: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    dividend  : T_ULARGE_INTEGER;
    divisor   : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.13 UInt64Div64Ex

The function divides two TwinCAT 2 unsigned 64-bit numbers (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Div64Ex: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    dividend  : T_ULARGE_INTEGER;
    divisor   : T_ULARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

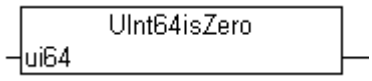
```
VAR_IN_OUT
    remainder : T_ULARGE_INTEGER;
END_VAR
```

remainder: Residual.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.14 UInt64isZero



The function returns TRUE if the TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]) has the value zero.

FUNCTION UInt64isZero: BOOL

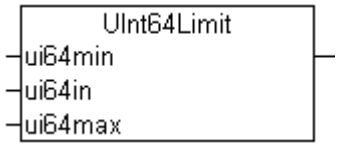
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.15 UInt64Limit



Limitation. The result is a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]).

FUNCTION UInt64Limit: T_ULARGE_INTEGER

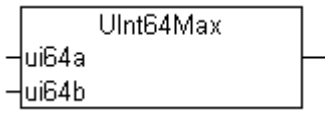
VAR_INPUT

```
VAR_INPUT
    ui64min : T_ULARGE_INTEGER;
    ui64in  : T_ULARGE_INTEGER;
    ui64max : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.16 UInt64Max



Maximum function. Returns the larger of two values (“legacy” type: T_ULARGE_INTEGER [▶ 320]).

FUNCTION UInt64Max: T_ULARGE_INTEGER

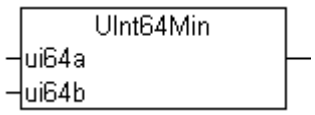
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.17 UInt64Min



Minimum function. Returns the smaller of two values (“legacy” type: T_ULARGE_INEGER [▶ 320]).

FUNCTION UInt64Min: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.18 UInt64Mod64

Modulo division of a TwinCAT 2 unsigned 64-bit number by another number (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Mod64: T_ULARGE_INTEGER

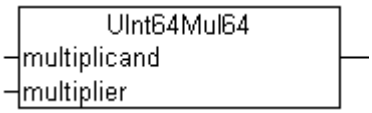
VAR_INPUT

```
VAR_INPUT
    dividend : T_ULARGE_INTEGER;
    divisor  : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.19 UInt64Mul64



The function multiplies two TwinCAT 2 unsigned 64-bit numbers (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Mul64: T_ULARGE_INTEGER

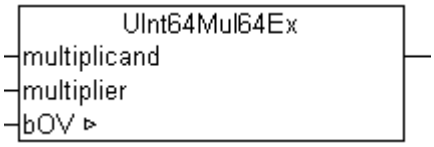
VAR_INPUT

```
VAR_INPUT
    multiplicand : T_ULARGE_INTEGER;
    multiplier   : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.20 UInt64Mul64Ex



The function multiplies two TwinCAT 2 unsigned 64-bit numbers (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Mul64Ex: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    multiplicand  : T_ULARGE_INTEGER;
    multiplier    : T_ULARGE_INTEGER;
END_VAR
```

VAR_IN_OUT

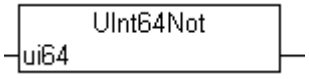
```
VAR_IN_OUT
    bOV : BOOL;(* TRUE => Arithmetic overflow, FALSE => no overflow *)
END_VAR
```

bOV: Arithmetic overflow. TRUE => overflow, FALSE => no overflow.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.21 UInt64Not



Bitwise NOT of a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Not: T_ULARGE_INTEGER

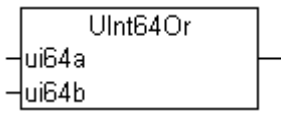
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.22 UInt64Or



Bitwise OR of two TwinCAT 2 unsigned 64-bit numbers (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Or: T_ULARGE_INTEGER

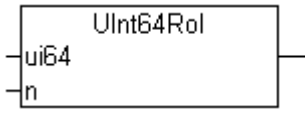
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.23 UInt64Rol



Bitwise left rotation of a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Rol: T_ULARGE_INTEGER

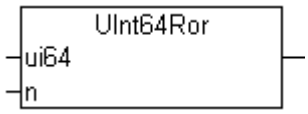
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
    n    : DWORD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.24 UInt64Ror



Bitwise right rotation of a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Ror: T_ULARGE_INTEGER

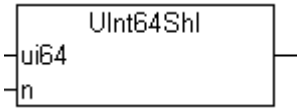
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
    n    : DWORD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.25 UInt64Shl



Bitwise left-shift of a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Shl: T_ULARGE_INTEGER

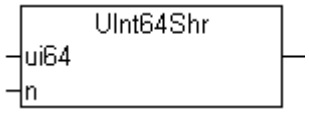
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
    n    : DWORD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.26 UInt64Shr



Bitwise right-shift of a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Shr: T_ULARGE_INTEGER

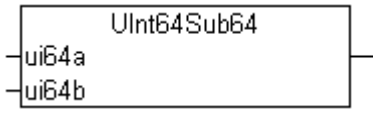
VAR_INPUT

```
VAR_INPUT
    ui64 : T_ULARGE_INTEGER;
    n    : DWORD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.27 UInt64Sub64



The function subtracts two TwinCAT 2 unsigned 64-bit numbers (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Sub64: T_ULARGE_INTEGER

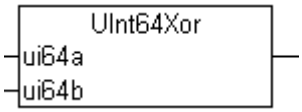
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.28 UInt64Xor



Bitwise XOR of two TwinCAT 2 unsigned 64-bit numbers (“legacy” type: T_ULARGE_INTEGER [▶ 320]). The result is an unsigned 64-bit number.

FUNCTION UInt64Xor: T_ULARGE_INTEGER

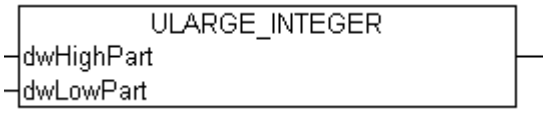
VAR_INPUT

```
VAR_INPUT
    ui64a : T_ULARGE_INTEGER;
    ui64b : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.29 ULARGE_INTEGER



The function initializes a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]).

FUNCTION ULARGE_INTEGER: T_ULARGE_INTEGER

VAR_INPUT

```
VAR_INPUT
    dwHighPart : DWORD;
    dwLowPart  : DWORD;
END_VAR
```

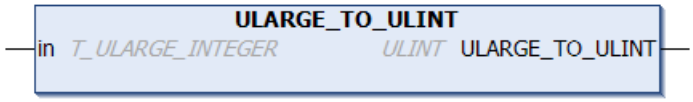
dwHighPart: Upper 32 bits.

dwLowPart: Lower 32 bits.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.30 ULARGE_TO_ULINT



The function converts a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]) into a TwinCAT 3 unsigned 64-bit number (“native” type).

FUNCTION ULARGE_TO_ULINT: ULINT

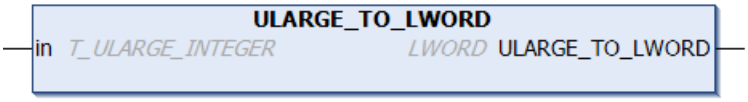
VAR_INPUT

```
VAR_INPUT
    in : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.9.31 ULARGE_TO_LWORD



The function converts a TwinCAT 2 unsigned 64-bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]) into a TwinCAT 3 unsigned 64-bit number (“native” type).

FUNCTION ULARGE_TO_LWORD: LWORD

VAR_INPUT

```
VAR_INPUT
    in : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10 T_Arg help functions

4.10.1 F_ARGCMP

This function compares two variables of type T_Arg and supplies the result of the comparison as return parameter.

FUNCTION F_ARGCMP: DINT

VAR_INPUT

```
VAR_INPUT
    typeSafe  : BOOL;
    arg1      : T_Arg;
    arg2      : T_Arg;
END_VAR
```

typeSafe: If TRUE => identical types can be compared (type-safe comparison). FALSE => different types can be compared (type-independent comparison).

arg1: First variable to be compared (type: T_Arg [▶ 316]).

arg2: Second variable to be compared (type: T_Arg [▶ 316]).

Return parameter	Relationship of the first different byte (type, length, value) in the first and second variable
-3	Length of *arg1* less than *arg2*
-2	Type of *arg1* less than *arg2*
-1	Value of *arg1* less than *arg2*
0	arg1 identical to *arg2*
1	Value of *arg1* greater than *arg2*
2	Type of *arg1* greater than *arg2*
3	Length of *arg1* greater than *arg2*
0xFF	Wrong parameter values, type, length, value of *arg1* or *arg2* = 0

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.2 F_ARGCPY

This function copies the value of a variable of type T_Arg into another variable and supplies the number of successfully copied data bytes as return parameter.

FUNCTION F_ARGCPY: UDINT

VAR_INPUT

```
VAR_INPUT
    typeSafe : BOOL;
END_VAR
```

typeSafe: If TRUE => identical types can be compared (type-safe comparison). FALSE => different types can be compared (type-independent comparison).

VAR_IN_OUT

```
VAR_IN_OUT
    dest : T_Arg;
    src  : T_Arg;
END_VAR
```

dest: Target variable for the copy operation (type: T_Arg [▶ 316]).

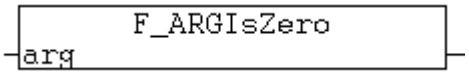
src: Source variable for the copy operation (type: T_Arg [▶ 316]).

Return parameter	Meaning
0	Incorrect parameter values. Type, length or value of *dest* or *src* == 0
> 0	If successful, the number of bytes copied.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.3 F_ARGISZERO



This function returns TRUE if one of the T_Arg member variables has the value zero or was not initialized.

FUNCTION F_ARGISZERO: BOOL

VAR_INPUT

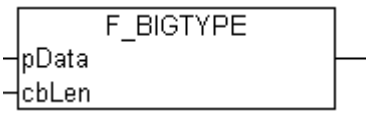
```
VAR_INPUT
    arg : T_Arg;
END_VAR
```

arg: Variable to be checked (type : T_Arg [▶ 316]).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.4 F_BIGTYPE



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a Struct or Array variable.

FUNCTION F_BIGTYPE: T_Arg

VAR_INPUT

```
VAR_INPUT
    pData : POINTER TO BYTE;
    cbLen : DWORD;
END_VAR
```

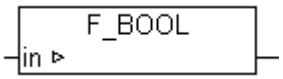
pData: Address pointer (can be determined with the ADR operator).

cbLen: Number of bytes occupied in the memory (can be determined with the SIZEOF operator).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.5 F_BOOL



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a BOOL variable.

FUNCTION F_BOOL: T_Arg

VAR_IN_OUT

```
VAR_IN_OUT
    in : BOOL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.6 F_BYTE

Help function that returns a structure with information (type: T_Arg [▶ 316]) on a BYTE variable.

FUNCTION F_BYTE: T_Arg

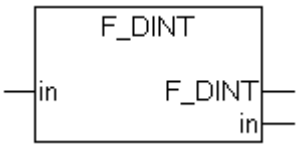
VAR_IN_OUT

```
VAR_IN_OUT
    in : BYTE;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.7 F_DINT



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a DINT variable.

FUNCTION F_DINT: T_Arg

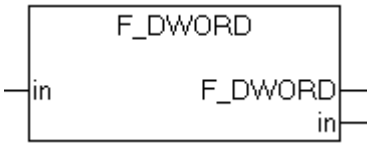
VAR_IN_OUT

```
VAR_IN_OUT
    in : DINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.8 F_DWORD



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a DWORD variable.

FUNCTION F_DWORD: T_Arg

VAR_IN_OUT

```
VAR_IN_OUT
    in : DWORD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.9 F_HUGE

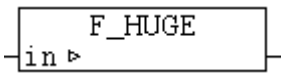


Fig. 1: F_HUGE

Help function that returns a structure with information (type: T_Arg [▶ 316]) on a T_HUGE_INTEGER [▶ 319] variable (signed 128-bit integer, TwinCAT 2 “legacy” type).

FUNCTION F_HUGE: T_Arg

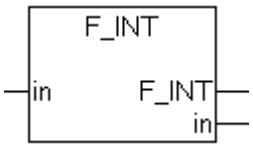
VAR_IN_OUT

```
VAR_IN_OUT
    in : T_HUGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.10 F_INT



Help function that returns a structure with information (type: T_Arg [▶ 316]) on an INT variable.

FUNCTION F_INT: T_Arg

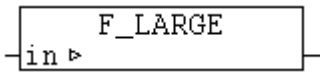
VAR_IN_OUT

```
VAR_IN_OUT
    in : INT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.11 F_LARGE



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a T_LARGE_INTEGER [▶ 320] variable (signed 64-bit integer, TwinCAT 2 “legacy” type).

FUNCTION F_LARGE: T_Arg

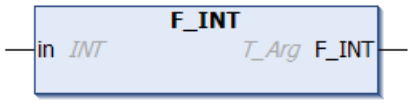
VAR_IN_OUT

```
VAR_IN_OUT
    in : T_LARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.12 F_LINT



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a LINT variable.

FUNCTION F_LINT: T_Arg

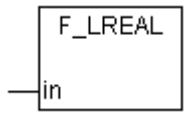
VAR_IN_OUT

```
VAR_IN_OUT
    in : LINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.13 F_LREAL



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a LREAL variable.

FUNCTION F_LREAL: T_Arg

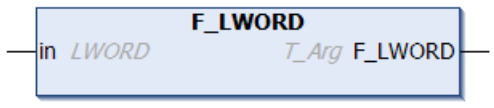
VAR_IN_OUT

```
VAR_IN_OUT
    in : LREAL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.14 F_LWORD



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a LWORD variable.

FUNCTION F_LWORD: T_Arg

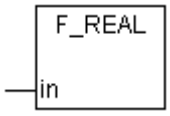
VAR_IN_OUT

```
VAR_IN_OUT
    in : LWORD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.15 F_REAL



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a REAL variable.

FUNCTION F_REAL: T_Arg

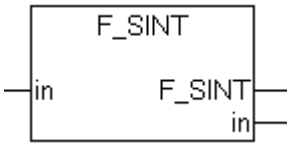
VAR_IN_OUT

```
VAR_IN_OUT
    in : REAL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.16 F_SINT



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a SINT variable.

FUNCTION F_SINT: T_Arg

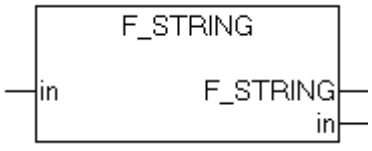
VAR_IN_OUT

```
VAR_IN_OUT
    in : SINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.17 F_STRING



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a T_MaxString variable.

FUNCTION F_STRING: T_Arg

VAR_IN_OUT

```
VAR_IN_OUT
    in : T_MaxString;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.18 F_STRINGEx

Help function that returns a structure with information (type: T_Arg [▶ 316]) about a STRING variable. Unlike the F_STRING [▶ 232] function, the length of the transferred STRING variable is arbitrary.

FUNCTION F_STRINGEx : T_Arg

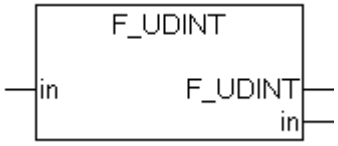
VAR_IN_OUT

```
VAR_IN_OUT CONSTANT
    in : STRING;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4022	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= v3.3.34.0

4.10.19 F_UDINT



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a UDINT variable.

FUNCTION F_UDINT: T_Arg

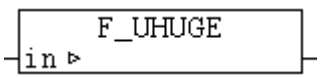
VAR_IN_OUT

```
VAR_IN_OUT
    in : UDINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.20 F_UHUGE



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a T_UHUGE_INTEGER [▶ 320] variable (unsigned 128-bit integer, TwinCAT 2 “legacy” type).

FUNCTION F_UHUGE: T_Arg

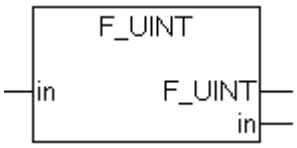
VAR_IN_OUT

```
VAR_IN_OUT
    in : T_UHUGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.21 F_UINT



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a UINT variable.

FUNCTION F_UINT: T_Arg

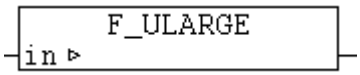
VAR_IN_OUT

```
VAR_IN_OUT
    in : UINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.22 F_ULARGE



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a T_LARGE_INTEGER [▶ 320] variable (signed 64-bit integer, TwinCAT 2 “legacy” type).

FUNCTION F_ULARGE: T_Arg

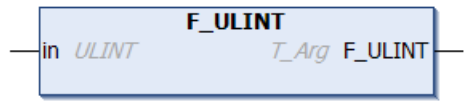
VAR_IN_OUT

```
VAR_IN_OUT
    in : T_ULARGE_INTEGER;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.23 F_ULINT



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a ULINT variable.

FUNCTION F_ULINT: T_Arg

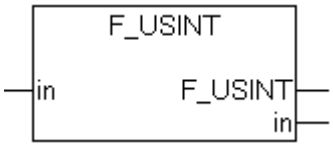
VAR_IN_OUT

```
VAR_IN_OUT
    in : ULINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.24 F_USINT



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a USINT variable.

FUNCTION F_USINT: T_Arg

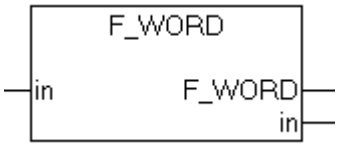
VAR_IN_OUT

```
VAR_IN_OUT
    in : USINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.25 F_WORD



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a WORD variable.

FUNCTION F_WORD: T_Arg

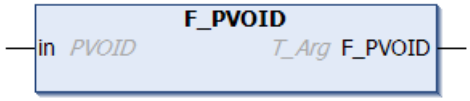
VAR_IN_OUT

```
VAR_IN_OUT
    in : WORD;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.26 F_PVOID



Help function that returns a structure with information (type: T_Arg [▶ 316]) on a BYTE variable.

FUNCTION F_PVOID: T_Arg

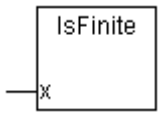
VAR_IN_OUT

```
VAR_IN_OUT
    in : PVOID;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.10.27 IsFinite



The function IsFinite() returns TRUE, if its argument has a finite value (INF < x < +INF). The function returns FALSE, if the argument is infinite or NaN (NaN = Not a number). IsFinite() checks whether the formatting of an LREAL or REAL variable complies with IEEE.

INF numbers may occur in a runtime system if the result of a mathematical operation falls outside the range that can be represented. E.g.:

```
PROGRAM MAIN
VAR
    fSingle : REAL := 12.34;
END_VAR
```

```
(*Cyclic called program code*)
fSingle := fSingle*2;
```

NaN numbers may occur in the runtime system if their actual formatting (memory content) was overwritten through illegal access (e.g. by using the MEMCPY of MEMSET functions). E.g.:

```
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 *)
```

Calling a conversion function with an NaN or INF number as parameter causes an FPU exception on a PC system (x86,x64). This exception subsequently leads to the PLC being stopped. The function IsFinite() enables the value of the variables to be checked, and therefore the FPU exception to be avoided and program execution to be continued.

FUNCTION IsFinite: BOOL

VAR_INPUT

```
VAR_INPUT
    x : T_Arg;
END_VAR
```

x: An auxiliary structure with information about the REAL or LREAL variables to be checked (type: T_Arg [▶ 316]). The structure parameters have to be generated when IsFinite() is called from auxiliary functions F_REAL [▶ 232] or F_LREAL [▶ 231] and transferred as parameters.

Example 1:

In the following example, the formatting of a REAL and an LREAL variable is checked, and an FPU exception is avoided.

```
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
```

Example 2:

In the following case, an FPU exception cannot be avoided through checking with IsFinite():

```
PROGRAM MAIN
VAR
    bigFloat   : LREAL := 3.0E100;
    smallDigit : INT;
END_VAR
```

```
IF IsFinite( F_LREAL( bigFloat ) ) THEN
    smallDigit := LREAL_TO_INT( bigFloat );
END_IF
```

While the bigFloat variable has the right formatting, the variable value is too large for conversion into an INT type. An exception is triggered on a PC system (x86,x64), and the runtime system is stopped.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11 [Obsolete]

4.11.1 Time functions

4.11.1.1 DT_TO_FILETIME

The function "DT_TO_FILETIME" can be used to convert a PLC variable in DATE_AND_TIME format (DT) to FILETIME format (64 bit).

FUNCTION DT_TO_FILETIME: T_FILETIME [▶ 317]

VAR_INPUT

```
VAR_INPUT
    DTIN  : DT;
END_VAR
```

DTIN: The date and time to be converted, in DATE_AND_TIME format.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11.1.2 F_TranslateFileTimeBias

This function converts the input time to the time in another time zone based on the specified bias time shift. The function can be used to convert the local time to UTC time (Universal Time Coordinates) and vice versa, for example.

FUNCTION F_TranslateFileTimeBias: T_FILETIME [▶ 317]

VAR_INPUT

```
VAR_INPUT
    in    : T_FILETIME;
    bias  : DINT;
    toUTC : BOOL;
END_VAR
```

in: Input time to be converted (type: T_FILETIME [▶ 317]).

bias: Difference between UTC time and local time in minutes (positive or negative values are permitted).

toUTC: This parameter can be used to specify the direction in which the input time is to be converted.

toUTC	Direction	Internal formula
FALSE	UTC -> local time	Local time := UTC - bias
TRUE	Local time -> UTC	UTC := local time + bias

Example:

The *in* variable contains the time to be converted. The *bToUTC* variable determines the conversion direction. If *bToUTC* = TRUE the local time is converted to UTC time, if *bToUTC* = FALSE the UTC time is converted to local time. The *WEST_EUROPE_TZI* constant contains the time zone information for Western Europe. The required bias value is calculated from the time zone information in the constant and the current bDST setting (Daylight Saving Time). Alternatively, the current time zone information of a TwinCAT system can be determined with the function block: FB_GetTimeZoneInformation [▶ 77].

Important notice: Data type DT was selected for the input time because of the visual control option in online mode. Conversions to other time formats are not necessarily recommend since the conversion functions can be very computing-intensive.

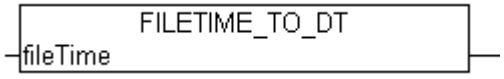
```
PROGRAM MAIN
VAR
    bDST    : BOOL := TRUE;(* TRUE => Daylight saving time, FALSE => Standard time *)
    bToUTC  : BOOL := FALSE;
(* TRUE => Convert local time to UTC time, FALSE => Convert UTC time to local time *)
    in      : DT := DT#2011-08-29-15:15:31;
    out     : DT;
    bias    : DINT;
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 ) );
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11.1.3 FILETIME_TO_DT



The function "FILETIME_TO_DT" converts the time in FILETIME format to DATE_AND_TIME format (DT). The DT format has a smaller value range than the FILETIME format and only offers second accuracy. For this reason the FILETIME value to be converted is limited. The permitted minimum corresponds to value *DT#1970-01-01-00:00:00* and the maximum to value *DT#2106-02-06-06:28:15*. Milliseconds are not considered in the conversion and are rounded down to the DATE_AND_TIME return value accordingly.

FUNCTION FILETIME_TO_DT : DT

VAR_INPUT

```
VAR_INPUT
    fileTime : T_FILETIME;
END_VAR
```

fileTime: The time to be converted in FILETIME format (type: T_FILETIME [▶ 317]).

Example:

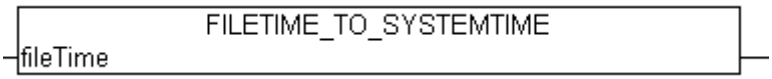
```
PROGRAM MAIN
VAR
    fbSystemTime   : GETSYSTEMTIME;
    timeAsFileTime : T_FILETIME;
    timeAsDT       : DT;
END_VAR
```

```
fbSystemTime( timeLoDW=>timeAsFileTime.dwLowDateTime, timeHiDW=>timeAsFileTime.dwHighDateTime );
timeAsDT := FILETIME_TO_DT( timeAsFileTime );
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11.1.4 FILETIME_TO_SYSTEMTIME



The function "FILETIME_TO_SYSTEMTIME" converts the time in FILETIME format into the "readable" SYSTEMTIME format. The conversion fails if the most significant bit of the 64-bit FILETIME variables is set. In this case the TIMESTRUCT member variables have the value zero.

FUNCTION FILETIME_TO_SYSTEMTIME: TIMESTRUCT [▶ 321]

VAR_INPUT

```
VAR_INPUT
    fileTime : T_FILETIME;
END_VAR
```

fileTime: The time to be converted in FILETIME format (type: T_FILETIME [▶ 317]).

Example:

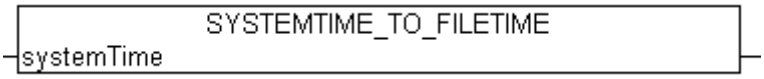
```
PROGRAM MAIN
VAR
    fbSystemTime     : GETSYSTEMTIME;
    timeAsFileTime   : T_FILETIME;
    timeAsSystemTime : TIMESTRUCT;
END_VAR
```

```
fbSystemTime( timeLoDW=>timeAsFileTime.dwLowDateTime, timeHiDW=>timeAsFileTime.dwHighDateTime );
timeAsSystemTime := FILETIME_TO_SYSTEMTIME( timeAsFileTime );
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11.1.5 SYSTEMTIME_TO_FILETIME



The function can be used to convert the Windows system time structure into the Filetime format. The day of the week wDayOfWeek of the system time variable is ignored. The system time year must be greater than 1601 and less than 30827.

FUNCTION SYSTEMTIME_TO_FILETIME: T_FILETIME [▶ 317]

VAR_INPUT

```
VAR_INPUT
    systemTime : TIMESTRUCT;
END_VAR
```

systemTime: The structure with the Windows system time requiring conversion (type: TIMESTRUCT [▶ 321]).

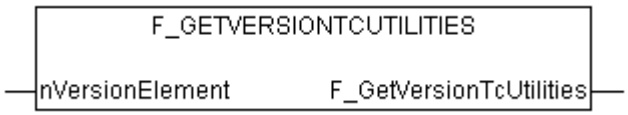
Return parameter	Description
0	Error, wrong system time parameter value.
> 0	No error. File time.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11.2 FUNCTION F_GetVersionTcUtilities

This function is obsolete and should not be used. Please use the global constant: stLibVersion_Tc2_Utilities to read the PLC library version [▶ 322] information.



This function can be used to read PLC library version information.

FUNCTION F_GetVersionTcUtilities : UINT

VAR_INPUT

```
VAR_INPUT
    nVersionElement : INT;
END_VAR
```

nVersionElement : Version element to be read. Possible parameters:

- 1 : major number;
- 2 : minor number;
- 3 : revision number;

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11.3 FLOATIsFinite

Obsolete function

Please use the LrealIsFinite [▶ 189]() function instead.

FUNCTION FLOATIsFinite: BOOL

VAR_INPUT

```
VAR_INPUT
    x        : LREAL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.11.4 FLOATIsNaN

Obsolete function

Please use the LrealIsNaN [▶ 189]() function instead.

FUNCTION FLOATIsNaN: BOOL

VAR_INPUT

```
VAR_INPUT
    x        : LREAL;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.12 ARG_TO_CSVFIELD

The function converts the value of a PLC variable to a data field in CSV format. Single quotation marks in the source file are replaced with double quotation marks. If the bQM parameter is set (QM = quotation marks) the outer quotation marks (around the CSV data field) are also added. If successful the function returns the length of the converted data as the result. The function returns zero if a conversion error occurred or in the event of missing data. The result is written into the byte buffer provided. The application must ensure that the buffer is large enough to contain the result.

The function is usually used together with the function block FB_CSVMemBufferWriter [▶ 50] to generate data sets in the PLC memory in CSV format. In the next step the memory content can be written to the file. In contrast to the STRING_TO_CSVFIELD [▶ 284] function this function can also be used to convert PLC variables with binary data into CSV data fields.

FUNCTION ARG_TO_CSVFIELD: UDINT

VAR_INPUT

```
VAR_INPUT
    in       : T_Arg;
    bQM      : BOOL;
    pOutput  : POINTER TO BYTE;
    cbOutput : UDINT;
END_VAR
```

in: PLC source variable whose value is to be converted into a data field in CSV format (type: T_Arg [▶ 316]).

bQM: If this input is TRUE the converted field data are enclosed in quotation marks.

pOutput: Start address (pointer) for the output buffer. The buffer address can be determined with the ADR operator. The result data are written into this buffer.

cbOutput: The maximum available size of the output buffer in bytes. The length of the output buffer can be determined with the SIZEOF operator.

Example:

The following example illustrates how PLC variables of different types can be converted to CSV format and vice versa. With ARG_TO_CSVFIELD conversion the result is copied into the byte buffer (field1..field6). With CSVFIELD_TO_ARG [▶ 248] conversion the source data are in the byte buffer (field1..field6), and the result is copied into the TwinCAT PLC variable.

```
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 ) )
);
```

The result (byte buffer as hexadecimal string):

cbField1 = 3, field1 = '22 01 22'

cbField2 = 10, field2 = '22 66 66 66 66 66 66 28 40 22'

cbField3 = 6, field3 = '22 07 00 00 00 22'

cbField4 = 25, field4 = '22 4D 6F 64 75 6C 65 3A 20 22 22 58 41 46 22 22 2C 20 04 05 2C 20 32 30 22'

cbField5 = 12, field5 = '22 00 01 02 03 04 05 06 07 08 09 22'

cbField6 = 5, field6 = '22 58 41 46 22'

cbVar1 = 1

cbVar2 = 8

cbVar3 = 4

cbVar4 = 22

cbVar5 = 10

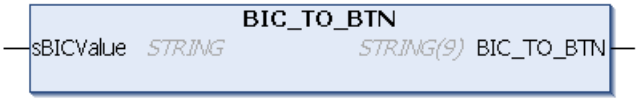
cbVar6 = 3

Further information can be found here: Example: Writing/reading of a CSV file [▶ 344].

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.13 BIC_TO_BTN



The function BIC_TO_BTN extracts the Beckhoff Traceability Number (BTN) from the Beckhoff Identification Code in sBICValue and returns it as return value. Spaces at the end of the BTN are automatically removed. If no BTN is found, then an empty string is returned.

FUNCTION BIC_TO_BTN : STRING(9)

VAR_INPUT

```
VAR_INPUT
    sBICValue   : STRING;
END_VAR
```

Name	Type	Description
sBICValue	**STRING**	This input must contain the Beckhoff Identification Code (BIC) from which the function BIC_TO_BTN extracts the Beckhoff Tracability Number (BTN), e.g. BIC_TO_BTN returns the value "0002agdw" for the BIC "1P193995SBTN0002agdw1KEL7411 Q1 2P112104020018".

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.14 BYTE_TO_BINSTR

This function converts a decimal number into a binary string (base 2).

FUNCTION BYTE_TO_BINSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : BYTE;
    iPrecision : INT;
END_VAR
```

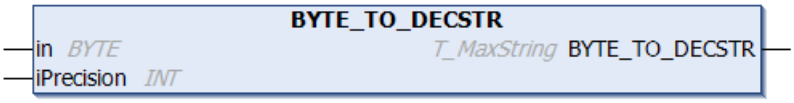
in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.15 BYTE_TO_DECSTR



This function converts a decimal number into a decimal string (base 10).

FUNCTION BYTE_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : BYTE;
    iPrecision : INT;
END_VAR
```

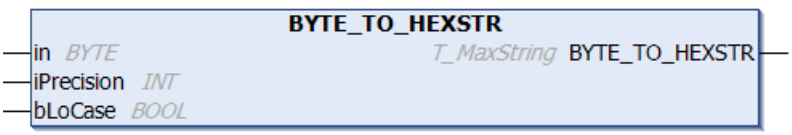
in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.16 BYTE_TO_HEXSTR



This function converts a decimal number into a hexadecimal string (base 16).

FUNCTION BYTE_TO_HEXSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : BYTE;
    iPrecision : INT;
    bLoCase    : BOOL := FALSE;
END_VAR
```

in: The decimal number requiring conversion.

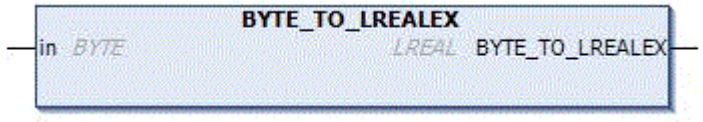
iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

bLoCase: This parameter determines whether lower or upper case letters are used in the conversion. FALSE => "ABCDEF", TRUE => "abcdef".

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.17 BYTE_TO_LREALEX



The conversion of unsigned numbers into floating point numbers of the type LREAL is not supported in TwinCAT 2 on the ARM platform. Unsigned numbers with the highest significant bit set may possibly be implicitly converted into negative floating point numbers. The function described here allows the explicit conversion of the type BYTE into a positive floating point number of the type LREAL in TwinCAT 2 (even if the highest significant bit was set and without compiler warning). You only need this function in order to be able to compile converted TwinCAT 2 projects without changes in TwinCAT 3.

Unsigned numbers of the type BYTE are always (implicitly and explicitly) converted into positive floating point numbers in TwinCAT3. For this reason this function can be dispensed with.

FUNCTION BYTE_TO_LREALEX: LREAL

VAR_INPUT

```
VAR_INPUT
    in : BYTE;
END_VAR
```

Sample:

```
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

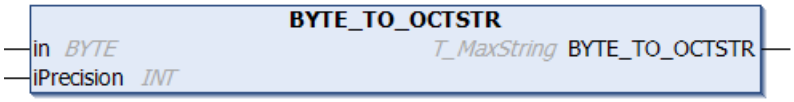
fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
`fLreal := 16#FF`	+255	+255	+255
`fLreal := BYTE_TO_LREALEX( nByte )`	+255	+255	+255
`fLreal := BYTE_TO_LREALEX( BYTE#16#FF )`	+255	+255	+255
`fLreal := BYTE_TO_LREALEX( 16#FF )`	+255	+255	+255

*Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.18 BYTE_TO_OCTSTR



This function converts a decimal number into an octal string (base 8).

FUNCTION BYTE_TO_OCTSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : BYTE;
    iPrecision : INT;
END_VAR
```

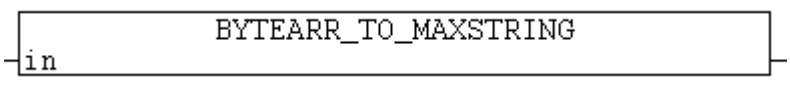
in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.19 BYTEARR_TO_MAXSTRING



Converts the individual ASCII codes of a byte array into a string.

FUNCTION BYTEARR_TO_MAXSTRING: T_MaxString

VAR_INPUT

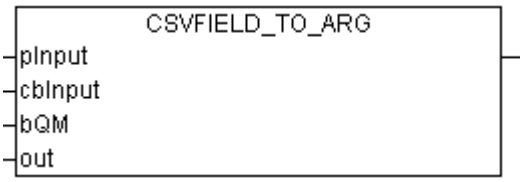
```
VAR_INPUT
    in : ARRAY[0..MAX_STRING_LENGTH] OF BYTE;
END_VAR
```

in: Byte-Array variable (MAX_STRING_LENGTH default value: 255).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.20 CSVFIELD_TO_ARG



The function converts the value from a data field in CSV format which is present as byte buffer into a PLC variable. Double quotation marks in the data field are replaced with simple quotation marks. If the bQM parameter is set (QM = quotation marks) the outer quotation marks (around the data field) are removed from the input data. If successful the function returns the length of the converted data. In the event of an error or if the length of the input data is zero the function returns the value zero. The application must ensure the PLC target variable is large enough to accommodate the value.

The function is usually used together with the function block FB_CSVMemBufferReader [▶ 48] in order to read (parse) data sets that are stored in the PLC memory in CSV format. Before this operation the CSV data sets are usually read from a file into the PLC memory. In contrast to the CSVFIELD_TO_STRING [▶ 249] function this function can also be used to convert CSV data fields with binary data into PLC variables.

FUNCTION CSVFIELD_TO_ARG: UDINT

VAR_INPUT

```
VAR_INPUT
    pInput   : POINTER TO BYTE;
    cbInput  : UDINT;
    bQM      : BOOL;
    out      : T_Arg;
END_VAR
```

pInput: Start address (pointer) to a byte buffer containing the data field to be converted in CSV format. The address can be determined with the ADR operator.

cbInput: Length of the data field to be converted in bytes. The length can be determined with the SIZEOF operator.

bQM: If this input is TRUE the enclosing quotation marks are removed from the field data.

out : PLC target variable into which the value of the data field is to be written (type: T_Arg [▶ 316]).

Example:

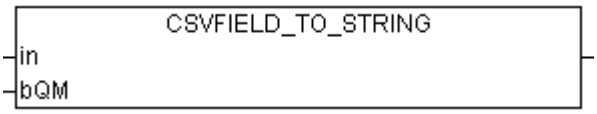
See example in the documentation for the ARG_TO_CSVFIELD [▶ 242] function block.

Further information can be found here: Example: Writing/reading of a CSV file [▶ 344].

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.21 CSVFIELD_TO_STRING



The function converts a data field in CSV data field format that is present as a source string into a value in PLC string format. Double quotation marks in the data field are replaced with simple quotation marks. If the bQM parameter is set (QM = quotation marks) the outer quotation marks (around the data field) are removed from the source string. If successful the function returns the convert string as the result. The function returns an empty string if an error occurred during the conversion, but only if the source string was not an empty string.

The function is usually used together with the function block FB_CSVMemBufferReader [▶ 48] in order to read (interpret) data sets that are stored in the PLC memory in CSV format. Before this operation the CSV data sets are usually read from a file into the PLC memory. The source string must not contain binary data. Binary data with the value zero would terminate and truncate the string in the wrong place. To convert data fields with binary data please use the function: CSVFIELD_TO_ARG [▶ 248].

FUNCTION CSFIELD_TO_STRING: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in  : T_MaxString;
    bQM : BOOL;
END_VAR
```

in: Source string with a data field in CSV format that is to be converted into a value in PLC string format (type: T_MaxString).

bQM: If this input is TRUE the enclosing quotation marks are removed from the source string.

bQM	Description	Source string	Result string	CSV-compliant
FALSE	Source string without enclosing quotation marks should only contain letters and numbers. In this case the source string must not contain any non-printable control character, quotation marks, semicolons, commas (US CSV format) or binary data.	'Module_XA5'	'Module_XA5'	Yes
		'123456'	'123456'	Yes
		''	''	Yes
		'A"""B'	'A""B'	No
		'A""B'	'A"B'	No
		';'	';'	No
		'RN'	'RN'	No
		'AB$00CD'	'AB' (string was truncated)	No

bQM	Description	Source string	Result string	CSV-compliant
TRUE	A source string that is not enclosed in quotation marks should not contain any non-printable control character, quotation mark, semicolon or comma (US CSV format). Binary data are not permitted.	'"Module_XA5"'	'Module_XA5'	Yes
	'"123456"'	'123456'	Yes	
	'""'	''	Yes	
	'"A""""B"'	'A""B'	Yes	
	'"A""B"'	'A"B'	Yes	
	'";"'	';'	Yes	
	'"RN"'	'RN'	Yes	
	'"AB$00CD"'	'AB' (string was truncated)	No	

Example:

```
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 );
```

The result:

s1 = 'ab_$04_$05_cd-"ALFA"_5'

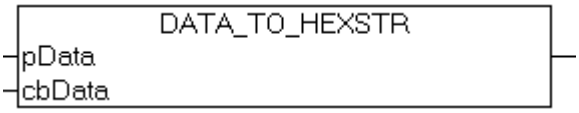
s2 = 'Module_50'

Further information can be found here: Example: Writing/reading of a CSV file [▶ 344].

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.22 DATA_TO_HEXSTR



The function converts binary data into a hexadecimal string. This function can be used for converting simple data types and structure variables. The maximum length of the binary data must not exceed 85 bytes. If the maximum length is exceeded, a dot ('.') is added to the results string and the conversion is aborted. The remaining data bytes are not converted. In the event of faulty function parameters (*pData* = zero or *cbData* = zero) the function returns an empty string.

FUNCTION DATA_TO_HEXSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    pData    : POINTER TO BYTE;
    cbData   : UDINT(0..85);
    bLoCase : BOOL := FALSE;
END_VAR
```

pData: Start address (pointer) for the binary data to be converted. The address can be determined with the ADR operator.

cbData: Max. length of the binary data to be converted. The length may not exceed 85 bytes. The length can be determined with the SIZEOF operator.

bLoCase: This parameter determines whether capital or lower-case letters are to be used in the conversion. TRUE = lower-case letters, FALSE = capital letters.

Example:

Please ensure that that the data size of the *overflow* variables does not exceed 85 bytes. For this reason, a dot is added to the results string *sH5*.

The byte order in the *number* variable is interchanged, since the memory organization of the counter variable is based on the little-endian-format (also referred to as Intel format).

```
PROGRAM MAIN
VAR
    str    : T_MaxString := 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789';
    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 );
```

The result:

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'

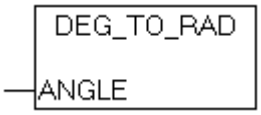
sH4 = '00 00 00 00'

sH5 = '00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00.'

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.23 DEG_TO_RAD



The function converts a degree angle to radian.

FUNCTION DEG_TO_RAD: LREAL

VAR_INPUT

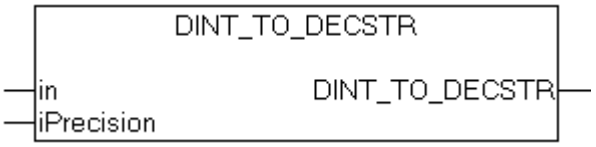
```
VAR_INPUT
    ANGLE  : LREAL;
END_VAR
```

ANGLE: The angle to be converted angle in degrees.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.24 DINT_TO_DECSTR



This function converts a signed decimal number into a decimal string (base 10).

FUNCTION DINT_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : DINT;
    iPrecision : INT;
END_VAR
```

in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty. For negative numbers, the negative sign will appear in the resulting string.

Example:

```
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 );
```

The result:

s1 = '-1234

s2 = '-0000001234'

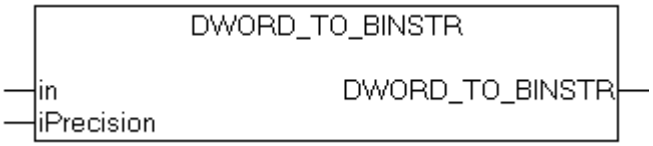
s3 = ''

s4 = '0000001234'

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.25 DWORD_TO_BINSTR



This function converts a decimal number into a binary string (base 2).

FUNCTION DWORD_TO_BINSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : DWORD;
    iPrecision : INT;
END_VAR
```

in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Example:

```
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 );
```

The result:

s1 = '0000000010000001'

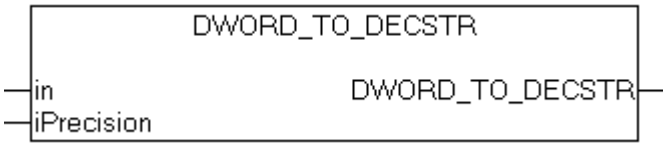
s2 = '1111'

s3 = ''

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.26 DWORD_TO_DECSTR



This function converts a decimal number into a decimal string (base 10).

FUNCTION DWORD_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : DWORD;
    iPrecision  : INT;
END_VAR
```

in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Example:

```
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 );
```

The result:

s1 = '43981'

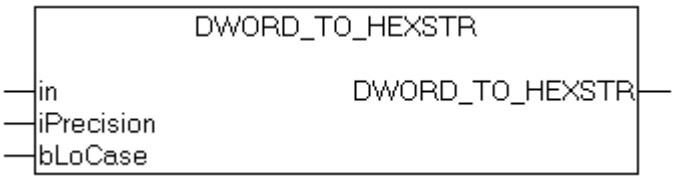
s2 = '0000043981'

s3 = ''

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.27 DWORD_TO_HEXSTR



This function converts a decimal number into a hexadecimal string (base 16).

FUNCTION DWORD_TO_HEXSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : DWORD;
    iPrecision : INT;
    bLoCase    : BOOL;
END_VAR
```

in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

bLoCase: This parameter determines whether lower or upper case letters are used in the conversion. FALSE => "ABCDEF", TRUE => "abcdef".

Example:

```
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 );
```

The result:

s1 = 'ABCD'

s2 = 'abcd'

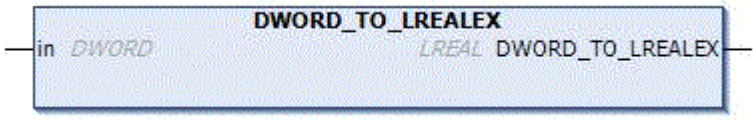
s3 = '000F'

s4 = ''

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.28 DWORD_TO_LREALEX



The conversion of unsigned numbers into floating point numbers of the type LREAL is not supported in TwinCAT 2 on the ARM platform. Unsigned numbers with the highest significant bit set may possibly be implicitly converted into negative floating point numbers. The function described here allows the explicit conversion of the type DWORD into a positive floating point number of the type LREAL in TwinCAT 2 (even if the highest significant bit was set and without compiler warning). You only need this function in order to be able to compile converted TwinCAT 2 projects without changes in TwinCAT 3.

Unsigned numbers of the type DWORD are always (implicitly and explicitly) converted into positive floating point numbers in TwinCAT3. For this reason this function can be dispensed with.

FUNCTION DWORD_TO_LREALEX: LREAL

VAR_INPUT

```
VAR_INPUT
    in : DWORD;
END_VAR
```

Sample:

```
PROGRAM MAIN
VAR
    nDword : DWORD := 16#FFFFFFFF;
    fLreal : LREAL := 0.0;
END_VAR
```

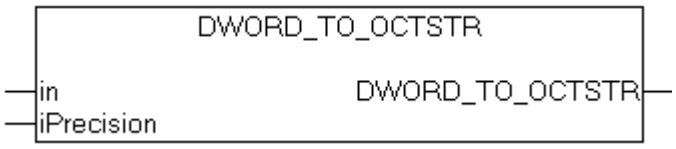
fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := nDword	-*1, Warning 1105**	4294967295	4294967295
fLreal := DWORD_TO_LREAL(nDword)	-*1, Warning 1105**	4294967295	4294967295
fLreal := DWORD#16#FFFFFFFF	-*1, Warning 1105**	4294967295	4294967295
fLreal := 16#FFFFFFFF	4294967295	4294967295	4294967295
fLreal := DWORD_TO_LREALEX(nDword)	4294967295	4294967295	4294967295
fLreal := DWORD_TO_LREALEX(DWORD#16#FFFFFFFF)	4294967295	4294967295	4294967295
fLreal := DWORD_TO_LREALEX(16#FFFFFFFF)	4294967295	4294967295	4294967295

*not supported
**Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.29 DWORD_TO_OCTSTR



This function converts a decimal number into an octal string (base 8).

FUNCTION DWORD_TO_OCTSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : DWORD;
    iPrecision : INT;
END_VAR
```

in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Example:

```
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 );
```

The result:

s1 = '125715'

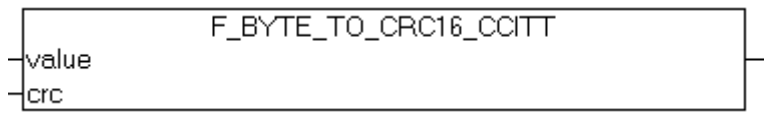
s2 = '0000125715'

s3 = ''

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.30 F_BYTE_TO_CRC16_CCITT



The function "F_BYTE_TO_CRC16_CCITT" can be used to determine a 16-bit CRC CCITT (cyclic redundancy check) for individual data bytes.

The generator polynomial used: Name: CRC-16 CCITT

- Standard: CRC-CCITT
- References: ITU X.25/T.30, ADCCP, SDLC/HDLC, ...
- Polynomial value: 0x1021
- Polynomial: x^16 + x^12 + x^5 + 1

FUNCTION F_BYTE_TO_CRC16_CCITT: WORD

VAR_INPUT

```
VAR_INPUT
    value   : BYTE;(* Data value *)
    crc     : WORD;(* Initial value (16#FFFF or 16#0000) or previous CRC-16 result *)
END_VAR
```

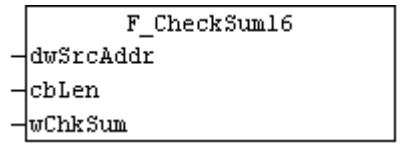
value: The data byte to be converted.

crc: Initial value = 16#FFFF or 16#0000 or the last CRC.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.31 F_CheckSum16



The function "F_CheckSum16" can be used to determine a 16-bit checksum for any data.

FUNCTION F_CheckSum16: WORD

VAR_INPUT

```
VAR_INPUT
    dwSrcAddr  : POINTER TO BYTE;
    cbLen      : UDINT;
    wChkSum    : WORD;
END_VAR
```

dwSrcAddr: Address of the data buffer.

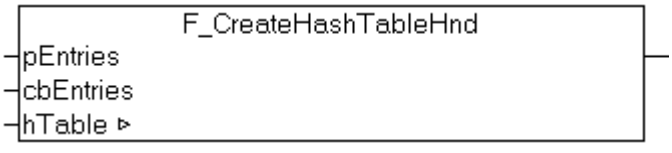
cbLen: Length of the data buffer.

wChkSum: Initial value = 0 or last checksum.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.32 F_CreateHashTableHnd



The function initializes the hash table handle. The table handle must be initialized once by calling the F_CreateHashTableHnd function.

FUNCTION F_CreateHashTableHnd: BOOL

VAR_INPUT

```
VAR_INPUT
    pEntries   : POINTER TO T_HashTableEntry := 0;
    cbEntries  : UDINT := 0;
END_VAR
```

pEntries: Address of the first T_HashTableEntry array element. The address can be determined with the ADR operator (type: T_HashTableEntry [▶ 318]).

cbEntries: T_HashTableEntry byte size. The byte size can be determined with the SIZEOF operator.

VAR_IN_OUT

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

hTable: Hash table handle to be initialized (type: T_HHASHTABLE [▶ 319]). The handle is required for accessing the hash table from the function block FB_HashTableCtrl [▶ 79].

Return parameter	Description
TRUE	No error
FALSE	Error

Example:

See: Example: Hash table (FB_HashTableCtrl). [▶ 334]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.33 F_CreateLinkedListHnd

The function initializes the Linked List Handle. The List Handle must be initialized once by calling the F_CreateLinkedListHnd function.

FUNCTION F_CreateLinkedListHnd: BOOL

VAR_INPUT

```
VAR_INPUT
    pEntries  : POINTER TO T_LinkedListEntry := 0;
    cbEntries : UDINT := 0;
END_VAR
```

pEntries: The address of the first *T_LinkedListEntry* array element. The address can be determined with the ADR operator (type: T_LinkedListEntry [▶ 320]).

cbEntries: The size of the *T_LinkedListEntry* array in bytes. The byte size can be determined with the SIZEOF operator.

VAR_IN_OUT

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

hList: Hash table handle to be initialized (type: T_HLINKEDLIST [▶ 319]). The handle is required for accessing the list from the function block FB_LinkedListCtrl [▶ 87].

Return parameter	Description
TRUE	No error
FALSE	Error

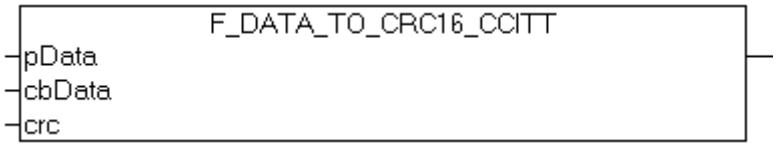
Example:

See: Example: Linked list (FB_LinkedListCtrl). [▶ 338]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.34 F_DATA_TO_CRC16_CCITT



The function "F_DATA_TO_CRC16_CCITT" can be used to determine a 16-bit CRC CCITT (cyclic redundancy check) for any data. Internally the function F_BYTE_TO_CRC16_CCITT [▶ 257] is used.

Further information on the algorithm used can be found in the documentation for the F_BYTE_TO_CRC16_CCITT [▶ 257] function.

FUNCTION F_DATA_TO_CRC16_CCITT: WORD

VAR_INPUT

```
VAR_INPUT
    pData  : POINTER TO BYTE;(* Pointer to first data byte *)
    cbData : UDINT;(* Length of data *)
    crc    : WORD;(* Initial value (16#FFFF or 16#0000) or previous CRC-16 result *)
END_VAR
```

pData: Address of the data buffer.

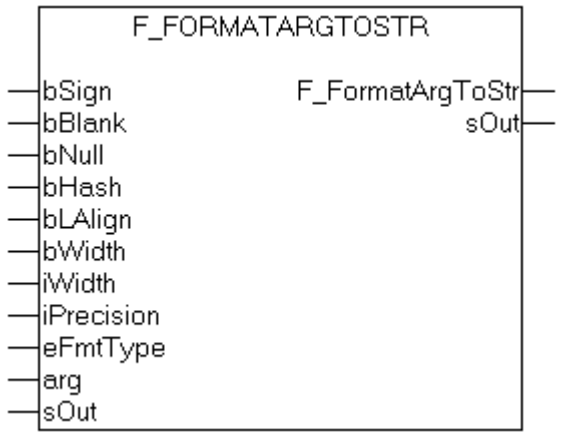
cbData: Length of the data buffer.

crc: Initial value = 16#FFFF or 16#0000 or the last CRC.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.35 F_FormatArgToStr



Auxiliary format function. This function is used internally by the FB_FormatString [▶ 64] function block. The function can be used to convert a variable of type T_Arg [▶ 316] into a formatted string according to the format specification [▶ 349].

FUNCTION F_FormatArgToStr: UDINT

VAR_INPUT

```
VAR_INPUT
   bSign      : BOOL;(* Sign prefix flag *)
   bBlank     : BOOL;(* Blank prefix flag *)
   bNull      : BOOL;(* Null prefix flag *)
   bHash      : BOOL;(* Hash prefix flag *)
   bLAlign    : BOOL;(* FALSE => Right align (default), TRUE => Left align *)
   bWidth     : BOOL;(* FALSE => no width padding, TRUE => blank or zeros padding enabled *)
   iWidth     : INT;(* Width length parameter *)
   iPrecision : INT;(* Precision length parameter *)
   eFmtType   : E_TypeFieldParam;    (* Format type field parameter *)
   arg        : T_Arg;(* Format argument *)
END_VAR
```

bSign: sign flag.

bBlank: blank flag.

bNull: null flag.

bHash: hash prefix flag.

bLAlign: alignment flag (TRUE=left align).

bWidth: If TRUE, the iWidth parameter is interpreted, otherwise not.

iWidth: Width parameter.

iPrecision: Precision parameter.

eFmtType: Type parameter (type: E_TypeFieldParam [▶ 302]).

arg: The argument to be formatted. The following auxiliary functions can be used for converting different types of PLC variables into the required data type T_Arg [▶ 316]: F_BYTE [▶ 228], F_WORD [▶ 235], F_DWORD [▶ 229], F_LWORD [▶ 231], F_SINT [▶ 232], F_INT [▶ 230], F_DINT [▶ 229], F_LINT [▶ 231], F_USINT [▶ 235], F_UINT [▶ 234], F_UDINT [▶ 233], F_ULINT [▶ 235], F_STRING [▶ 232], F_REAL [▶ 232], F_LREAL [▶ 231].

VAR_IN_OUT

```
VAR_IN_OUT
   sOut     : T_MaxString;
END_VAR
```

sOut: If successful, this variable returns the formatted output string (Type:T_MaxString).

Return parameter	Meaning
0	No error
<> 0	Error. For error description see Format error codes [▶ 352]

Examples:

Formatting a BYTE variable as a binary string.

```
PROGRAM MAIN
VAR
    s1      : T_MaxString;
    s2      : T_MaxString;
    s3      : T_MaxString;
    s4      : T_MaxString;
    s5      : T_MaxString;
    errID   : UDINT;
    varByte : BYTE;
    double  : LREAL;
    L1      : INT;
    L2      : INT;
    L3      : INT;
    L4      : INT;
    L5      : INT;
END_VAR
```

```
varByte := 128;
errID   := F_FormatArgToStr(FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, 20, 8, TYPEFIELD_B, F_BYTE( va
rByte ), s1 );
errID   := F_FormatArgToStr(FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, 20, 8, TYPEFIELD_B, F_BYTE( var
Byte ), s2 );
errID   := F_FormatArgToStr(FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, 20, 8, TYPEFIELD_B, F_BYTE( varB
yte ), s3 );
errID   := F_FormatArgToStr(FALSE, FALSE, FALSE, TRUE, TRUE, TRUE, 20, 8, TYPEFIELD_B, F_BYTE( varBy
te ), s4 );
L1 := LEN( s1 );
L2 := LEN( s2 );
L3 := LEN( s3 );
L4 := LEN( s4 );
```

The result:

```
s1 = '10000000'
s2 = '        10000000'
s3 = '10000000        '
s4 = '2#10000000      '
L1 = 8
L2 = 20
L3 = 20
L4 = 20
```

Formatting an LREAL variable.

```
double := 12345.6789;
errID  := F_FormatArgToStr( FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, 20, 8, TYPEFIELD_F, F_LREAL( d
ouble ), s1 );
errID  := F_FormatArgToStr( FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, 20, 8, TYPEFIELD_F, F_LREAL( do
uble ), s2 );
errID  := F_FormatArgToStr( FALSE, FALSE, FALSE, FALSE, TRUE, TRUE, 20, 8, TYPEFIELD_F, F_LREAL( dou
ble ), s3 );
errID  := F_FormatArgToStr( FALSE, FALSE, TRUE, FALSE, FALSE, TRUE, 20, 8, TYPEFIELD_F, F_LREAL( dou
ble ), s4 );
errID  := F_FormatArgToStr( TRUE, FALSE, FALSE, TRUE, TRUE, TRUE, 20, 8, TYPEFIELD_F, F_LREAL( doubl
e ), s5 );
L1 := LEN( s1 );
L2 := LEN( s2 );
L3 := LEN( s3 );
L4 := LEN( s4 );
L5 := LEN( s5 );
```

The result:

```
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
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.36 F_GenerateHashValue

The function can be used to calculate a hash value.

If the calculation is successful, the function returns TRUE.

FUNCTION F_GenerateHashValue : BOOL

VAR_INPUT

```
VAR_INPUT
    hashMode : E_HashMode;
    pData    : PVOID;
    nData    : UDINT;
    pHash    : PVOID;    // destination buffer for generated hash value
    nHash    : UDINT;    // size of destination buffer in bytes. This needs to match the hash mode.
END_VAR
```

hashMode: A hash mode, such as SHA 512, is specified here. See E_HashMode [▶ 297].

pData: The address of the input data is specified here.

nData: The size of the input data in bytes is specified here.

pHash: Here the address of the buffer is specified where the hash value is to be stored.

nHash: The size of the buffer for the hash value in bytes is specified here. The size depends on the hash mode, see also E_HashMode [▶ 297]. The appropriate size must be specified. Otherwise the function will fail.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024.29	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.51.0

4.37 F_GetClassIdVersioned

The function calculates the versioned Class ID based on the Class ID and the Library ID. This is used for versioned C++ projects.

FUNCTION F_GetClassIdVersioned : BOOL

VAR_INPUT

```
VAR_INPUT
    sLibraryId     : STRING(255); // 'vendorName|libraryName|libraryVersion' (e.g. 'C+
+ Module Vendor|IncrementerCpp|0.0.0.1' )
    clsId          : CLSID;
    clsIdVersioned : REFERENCE TO CLSID;
END_VAR
```

sLibraryId: the Library ID is specified here.

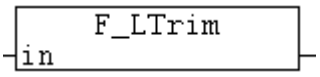
clsId: the Class ID is specified here.

clsIdVersioned: the calculated versioned Class ID is output here.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024.29	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.51.0

4.38 F_LTrim



Removes leading spaces from the character string and returns the reduced character string.

FUNCTION F_LTrim: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in  : T_MaxString;
END_VAR
```

in: The string to be converted (Type:T_MaxString).

Example:

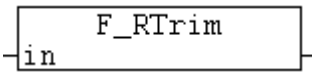
```
PROGRAM MAIN
VAR
    sLTrim : STRING;
END_VAR
```

```
sLTrim := F_LTrim(' <trim ');(* result: '<trim ' *)
sLTrim := F_LTrim(' <trim');(* result: '<trim' *)
sLTrim := F_LTrim('<trim');(* result: '<trim' *)
sLTrim := F_LTrim('');(* result: '' *)
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.39 F_RTrim



Truncates all trailing spaces from the specified value and returns the result.

FUNCTION F_RTrim: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in : T_MaxString;
END_VAR
```

in: The string to be converted (Type: T_MaxString).

Example:

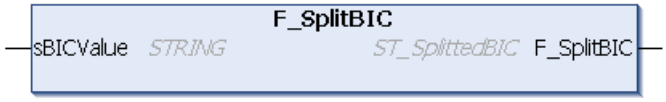
```
PROGRAM MAIN
VAR
    sRTrim  : STRING;
    sLRTrim : STRING;
END_VAR
```

```
sRTrim := F_RTrim(' trim> ');(* result: ' trim>' *)
sRTrim := F_RTrim('trim> ');(* result: 'trim>' *)
sRTrim := F_RTrim('trim>');(* result: 'trim>' *)
sRTrim := F_RTrim('');(* result: '' *)

sLRTrim := F_RTrim( F_LTrim( ' <trim> '));(* result: '<trim>' *)
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.40 F_SplitBIC



The F_SplitBIC function splits the Beckhoff Identification Code (BIC) sBICValue into its components using known identifiers and returns the recognized substrings in a structure ST_SplittedBIC [▶ 312] as return value. Spaces at the end of substrings are automatically removed. Unused substrings are returned as empty strings. If an unknown identifier is found, then the remaining string of the BIC is passed on as sUndefined in the return structure. The BIC can be read e.g. with the function blocks FB_EcCoEReadBIC or FB_EcReadBIC (Tc2_EtherCAT Library) from EtherCAT slaves.

FUNCTION F_SplitBIC : ST_SplittedBIC

VAR_INPUT

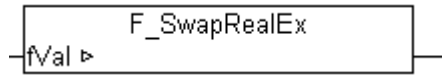
```
VAR_INPUT
    sBICValue   : STRING;
END_VAR
```

Name	Type	Description
sBICValue	**STRING**	This input must contain the Beckhoff Identification Code (BIC), which is split into substrings using the F_SplitBIC function, e.g. "1P193995SBTN0002agdw1KEL7411 Q1 2P112104020018".

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.41 F_SwapRealEx



The memory representation of a REAL number in a Bus Terminal Controller (e.g. BC2000, BC3100, BC9000) differs from the memory representation of a REAL number in an x86/x64/ARM system (IPC or Embedded PC).

To be able to represent a REAL number of a Bus Terminal Controller properly on an IPC, the Hi and Lo words of the REAL number must be interchanged. Under the programming environment this is already done in online or simulation mode. To be able to request the REAL data of a Bus Controller via the network (ADS protocol, ADSDLL, AdsOcx etc.) and represent them properly on an x86/x64/ARM IPC, the REAL data have to be converted into the correct format. This can be done on the Bus Terminal Controller side or the IPC side.

The function F_SwapRealEx can be used to convert the REAL variables (e.g. variables to be read by a VB application or recorded with TwinCAT Scope View) into a suitable format on the PC side. The function changes the memory representation of the transferred *fVal* parameter (VAR_IN_OUT).

FUNCTION F_SwapRealEx: BOOL

VAR_IN_OUT

```
VAR_IN_OUT
    fVal : REAL;
END_VAR
```

fVal: The REAL value to be converted.

Return parameter	Meaning
TRUE	No error
FALSE	Error during function execution

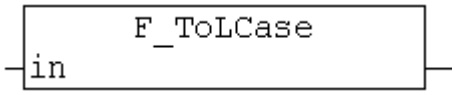
Example:

See: Example: Communication BC/BX<->PC/CX (F_SwapRealEx). [▶ 324]

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.42 F_ToLCase



The F_ToLCase function converts all capital letters of a character string to lower-case letters.

Character set

By default the conversion function uses the character set of the Windows code pager 1252 Latin 1, SBCS (singles byte character set). A different character set can be selected at runtime (currently only Windows code page 1250 Central European) via the global variable GLOBAL_SBCS_TABLE selected (see example).

FUNCTION F_ToLCase: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in : T_MaxString;
END_VAR
```

in: The string to be converted (Type: T_MaxString).

Example:

```
PROGRAM MAIN
VAR
    sLCase : STRING;
END_VAR
```

```
sLCase := F_ToLCase( 'TO LOWER CASE 1234567890 ÄÖÜß' );
```

The result of the conversion is: 'to lower case 1234567890 äöüß'

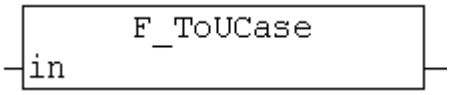
```
GLOBAL_SBCS_TABLE := eSBCS_CentralEuropean;
sLCase := F_ToLCase( 'TO LOWER CASE 1234567890 ĄĘŚĆŻŹŁÓ' );
```

The result of the conversion is: 'to lower case 1234567890 ąęśćżźłó'

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.43 F_ToUCase



The F_ToUCase function converts all lower-case letters of a character string to capital letters.

Character set

By default the conversion function uses the character set of the Windows code pager 1252 Latin 1, SBCS (singles byte character set). A different character set can be selected at runtime (currently only Windows code page 1250 Central European) via the global variable GLOBAL_SBCS_TABLE selected (see example).

FUNCTION F_ToUCase: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in : T_MaxString;
END_VAR
```

in: The string to be converted (Type: T_MaxString).

Example:

```
PROGRAM MAIN
VAR
    sUCase : STRING;
END_VAR
```

```
sUCase := F_ToUCase( 'to upper case 1234567890 äöüß' );
```

The result of the conversion is: 'TO UPPER CASE 1234567890 ÄÖÜß'

```
GLOBAL_SBCS_TABLE := eSBCS_CentralEuropean;
sUCase := F_ToUCase( 'to upper case 1234567890 ąęśćżźłó' );
```

The result of the conversion is: 'TO UPPER CASE 1234567890 ĄĘŚĆŻŹŁÓ'

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.44 GUID_TO_REGSTRING

This function converts a structured GUID [▶ 303] variable into a registry GUID string variable (enclosed in curly brackets).

FUNCTION GUID_TO_REGSTRING: STRING(38)

VAR_INPUT

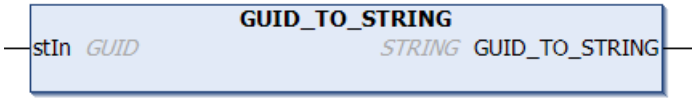
```
VAR_INPUT
    in : GUID;
END_VAR
```

Return value	Meaning
'{xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}'	No error (‘x’ is a hexadecimal half-byte)
'{00000000-0000-0000-0000-000000000000}'	No error, GUID has the initial value (all bytes are zero)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.45 GUID_TO_STRING



This function converts a structured GUID [▶ 303] variable into a GUID string variable (without curly brackets).

FUNCTION GUID_TO_STRING: STRING

VAR_INPUT

```
VAR_INPUT
    stIn : GUID;
END_VAR
```

Return value	Meaning
'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'	No error (‘x’ is a hexadecimal half-byte)
00000000-0000-0000-0000-000000000000'	No error, GUID has the initial value (all bytes are zero)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.46 GuidsEqualByVal

This function compares two GUID values.

FUNCTION GuidsEqualByVal: BOOL

VAR_INPUT

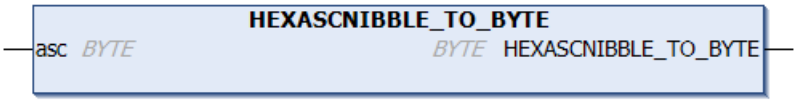
```
VAR_INPUT
    guidA : GUID;
    guidB : GUID;
END_VAR
```

Return value	Meaning
FALSE	guidA <> guidB
TRUE	guidA = guidB

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.47 HEXASCNIBBLE_TO_BYTE



This function converts the ASCII code of a hexadecimal half-byte character into the decimal value.

FUNCTION HEXASCNIBBLE_TO_BYTE: BYTE

VAR_INPUT

```
VAR_INPUT
    asc : BYTE;
END_VAR
```

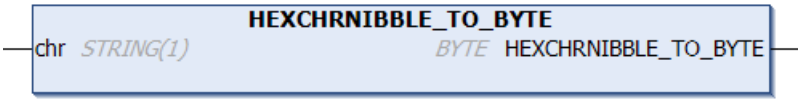
asc: Ascii-code of a hexadecimal half-byte character (Ascii code from: ‘0’ to ‘9’ or ‘a’ to ‘f’ or ‘A’ to ‘F’).

Return value	Meaning
0 to 15	Successful, no error.
255	Error, wrong input parameter value.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.48 HEXCHRNIBBLE_TO_BYTE



This function converts a hexadecimal half-byte character into its decimal value.

FUNCTION HEXCHRNIBBLE_TO_BYTE: BYTE

VAR_INPUT

```
VAR_INPUT
    chr : STRING(1);
END_VAR
```

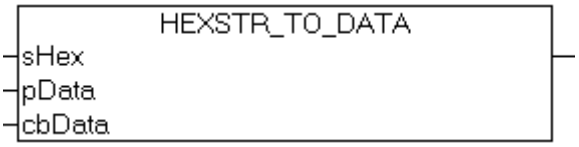
chr: Hexadecimal half-byte character (‘0’ to ‘9’ or ‘a’ to ‘f’ or ‘A’ to ‘F’).

Return value	Meaning
0 to 15	Successful, no error.
255	Error, wrong input parameter value.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.49 HEXSTR_TO_DATA



The function converts a hexadecimal string into binary data and returns the number of successfully converted data bytes as result. Only spaces may be used as separators in the hexadecimal string to be converted. Lower and upper case letters are permitted as hex characters. In the event of an error or an illegal character the conversion is aborted and a zero length is returned as result.

FUNCTION HEXSTR_TO_DATA: UDINT

VAR_INPUT

```
VAR_INPUT
    sHex   : T_MaxString;
    pData  : POINTER TO BYTE;
    cbData : UDINT;
END_VAR
```

sHex: The hexadecimal string to convert (type: T_MaxString, e.g.: 'AB CD 01 23').

pData: Start address (pointer) to the destination buffer into which the converted data bytes are to be written. The address can be determined with the ADR operator.

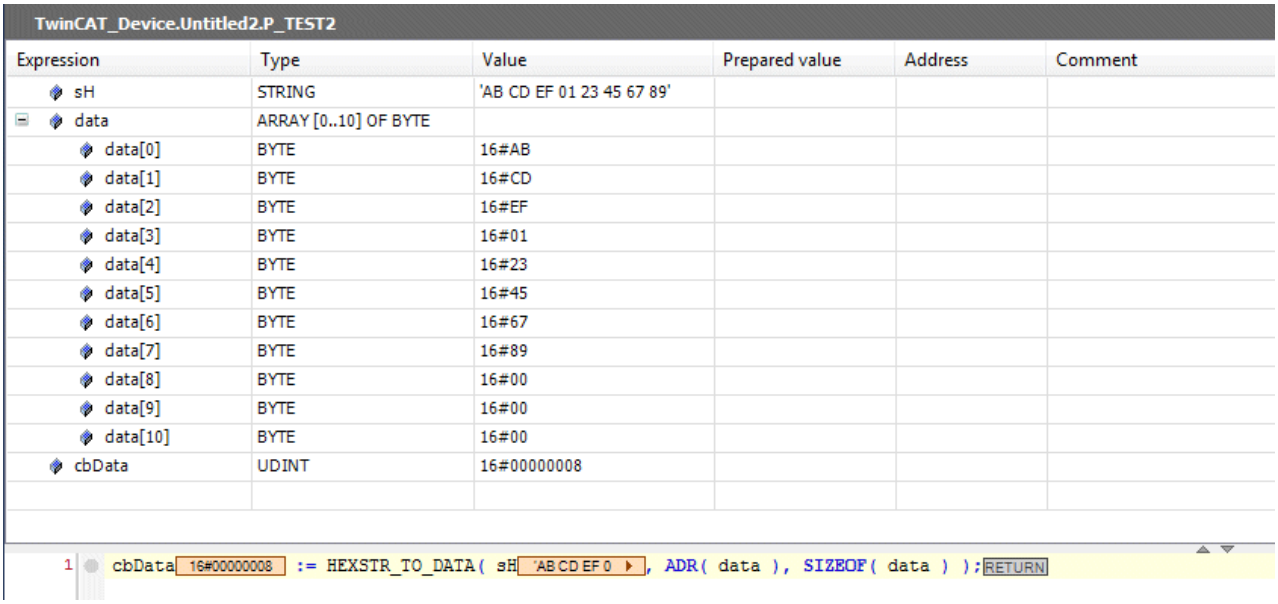
cbData: Max. available length of the destination buffer. The length can be determined with the SIZEOF operator.

Example:

```
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 ) );
```

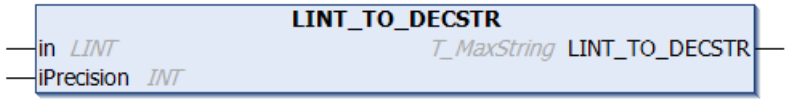
The result (online):



Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.50 LINT_TO_DECSTR



This function converts a signed decimal number into a decimal string (base 10).

FUNCTION LINT_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : LINT;
    iPrecision : INT;
END_VAR
```

in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty. For negative numbers, the negative sign will appear in the resulting string.

Example:

```
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 );
```

The result:

s1 = '-1234

s2 = '-0000001234'

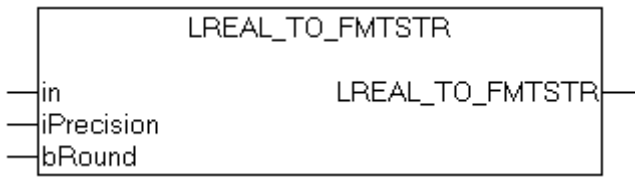
s3 = ''

s4 = '0000001234'

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.51 LREAL_TO_FMTSTR



The function converts and formats a floating-point number into a string variable with the following format: **[-]dddd.dddd** (dddd are decimal numbers). The number of digits before the decimal point depends on the value of the floating-point number. The number of digits after the decimal point depends on the required precision. The sign only appears for negative values. **'#INF'** is returned for infinite positive values, **'-#INF'** for infinite negative values. If the variable transferred has an illegal value (NaN, Not-a-Number), **'#QNAN'** or **'-#QNAN'** is returned. If the length of the formatted string exceeds the maximum permissible length of the resulting string, **'#OVF'** or **'-#OVF'** is returned.

FUNCTION LREAL_TO_FMTSTR: STRING(510)

VAR_INPUT

```
VAR_INPUT
    in         : LREAL;
    iPrecision : INT;
    bRound     : BOOL;
END_VAR
```

in: Floating-point number that is to be converted and formatted.

iPrecision: Precision. The value determines the number of digits after the decimal point. With the minimum value (zero), no decimal places are displayed. The maximum value of *iPrecision* is limited by the number of digits before the decimal point and the maximum permissible length of the resulting string. If *in* = 0 and *iPrecision* = 0 then string '0' is returned.

bRound: If the *bRound* parameter is set, the formatted string is rounded to the respective number of decimal places (*iPrecision*). The following rule applies for rounding: If the decimal number after the last required decimal place is >= 5, the value is rounded up, otherwise not.

Example 1:

0.46523 is to be converted into a string with two decimal places and rounded.

```
sOut := LREAL_TO_FMTSTR( 0.46523, 2, TRUE );
```

The result is: '0.47';

Example: 2

The maximum number of significant digits for LREAL variables is limited to 15.

Due to the internal representation of floating-point numbers and rounding errors during conversion, the resulting string may not precisely correspond to the value of the *in* variable.

```
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 );
```

The result is:

s1 = '0.4999999999999999756000000' This is how *double* variables are represented internally. This number is used as the starting point for the rounding operation.

s2 = '0.49'

Rounding leads to the following results:

s3 = '0'

s4 = '0.50'

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.52 LWORD_TO_BASE36STR

This function converts a decimal number into a Base36 string (base 16). For numbers with base 16, the letters A-Z are used in addition to the digits 0-9.

FUNCTION LWORD_TO_BASE36STR : T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : LWORD;
    iPrecision  : INT;
    bLoCase     : BOOL;
END_VAR
```

in: The decimal number to be converted.

iPrecision: Minimum number of displayed digits. If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

bLoCase: This parameter determines whether lower or upper case letters are used in the conversion. FALSE => "ABCDEFXY", TRUE => "abcdefxy".

Example:

```
PROGRAM MAIN
VAR
    s1   : STRING;
    s2   : STRING;
    s3   : STRING;
    s4   : STRING;
    nCnt : LWORD;
END_VAR
```

```
nCnt := 43981;
s1   := LWORD_TO_BASE36STR( nCnt, 1, FALSE );
s2   := LWORD_TO_BASE36STR( nCnt, 1, TRUE );
nCnt := 15;
s3   := LWORD_TO_BASE36STR( nCnt, 4, FALSE );
nCnt := 0;
s4   := LWORD_TO_BASE36STR( nCnt, 0, FALSE );
```

The result:

s1 = 'XXP'

s2 = 'xxp'

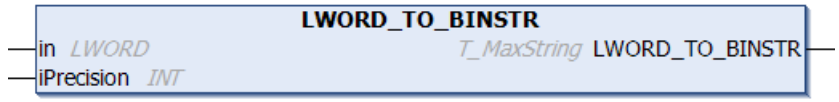
s3 = '000F'

s4 = ''

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.47.0

4.53 LWORD_TO_BINSTR



This function converts a decimal number into a binary string (base 2).

FUNCTION LWORD_TO_BINSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : LWORD;
    iPrecision : INT;
END_VAR
```

in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Example:

```
PROGRAM MAIN
VAR
    s1   : STRING;
    s2   : STRING;
    s3   : STRING;
    nCnt : LWORD;
END_VAR
```

```
s1   := LWORD_TO_BINSTR( 16#81, 16 );
nCnt := 15;
s2   := LWORD_TO_BINSTR( nCnt, 1 );
nCnt := 0;
s3   := LWORD_TO_BINSTR( nCnt, 0 );
```

The result:

s1 = '0000000010000001'

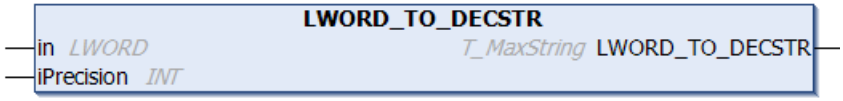
s2 = '1111'

s3 = ''

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.54 LWORD_TO_DECSTR



This function converts a decimal number into a decimal string (base 10).

FUNCTION LWORD_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : LWORD;
    iPrecision : INT;
END_VAR
```

in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Example:

```
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 );
```

The result:

s1 = '43981'

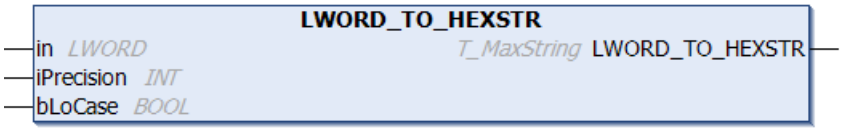
s2 = '0000043981'

s3 = ''

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.55 LWORD_TO_HEXSTR



This function converts a decimal number into a hexadecimal string (base 16).

FUNCTION LWORD_TO_HEXSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : LWORD;
    iPrecision  : INT;
    bLoCase     : BOOL;
END_VAR
```

in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

bLoCase: This parameter determines whether lower or upper case letters are used in the conversion. FALSE => "ABCDEF", TRUE => "abcdef".

Example:

```
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 );
```

The result:

s1 = 'ABCD'

s2 = 'abcd'

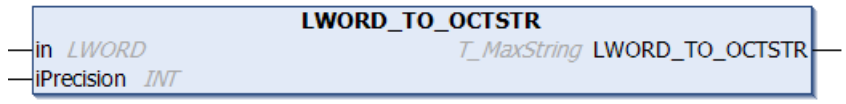
s3 = '000F'

s4 = ''

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.56 LWORD_TO_OCTSTR



This function converts a decimal number into an octal string (base 8).

FUNCTION LWORD_TO_OCTSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : LWORD;
    iPrecision : INT;
END_VAR
```

in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Example:

```
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 );
```

The result:

s1 = '125715'

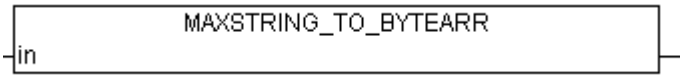
s2 = '0000125715'

s3 = ''

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.57 MAXSTRING_TO_BYTEARR



Converts a string into individual ASCII codes of a byte array.

FUNCTION MAXSTRING_TO_BYTEARR: ARRAY[0..MAX_STRING_LENGTH] OF BYTE

VAR_INPUT

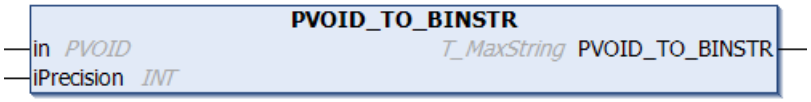
```
VAR_INPUT
    in : T_MaxString;
END_VAR
```

in: String to be converted (type: T_MaxString).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.58 PVOID_TO_BINSTR



The function converts the value of a pointer variable of type PVOID into a binary string (basis 2).

FUNCTION PVOID_TO_BINSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : PVOID;
    iPrecision  : INT;
END_VAR
```

in: The pointer variable to be converted.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Example:

```
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 );
```

The result:

s1 = ''

s2 = '0'

s3 = '00000000000000000000000000000000'

s4 = '10000111110111100000001001010101' (may vary)

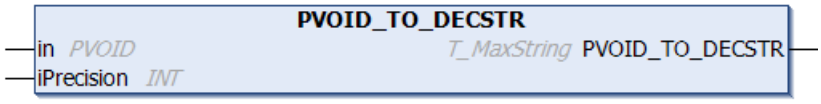
s5 = '10000111110111100000001001010101' (may vary)

s6 = '10000111110111100000001001010101' (may vary)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.59 PVOID_TO_DECSTR



The function converts the value of a pointer variable of type PVOID into a decimal string (basis 10).

FUNCTION PVOID_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : PVOID;
    iPrecision : INT;
END_VAR
```

in: The pointer variable to be converted.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Example:

```
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 );
```

The result:

s1 = ''

s2 = '0'

s3 = '0000000000000000'

s4 = '2279473749' (may vary)

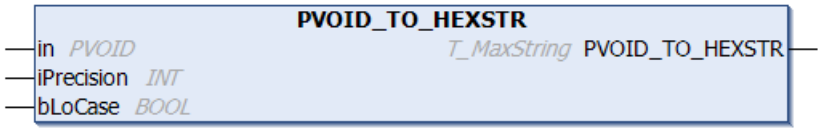
s5 = '2279473749' (may vary)

s6 = '0000002279473749' (may vary)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.60 PVOID_TO_HEXSTR



The function converts the value of a pointer variable of type PVOID into a hexadecimal string (basis 16).

FUNCTION PVOID_TO_HEXSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in          : PVOID;
    iPrecision  : INT;
    bLoCase     : BOOL;
END_VAR
```

in: The pointer variable to be converted.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

bLoCase: This parameter determines whether lower or upper case letters are used in the conversion. FALSE => "ABCDEF", TRUE => "abcdef".

Example:

```
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 );
```

The result:

s1 = ''

s2 = ''

s3 = '0'

s4 = '0'

s5 = '0000000000000000'

s6 = '0000000000000000'

s7 = '87CBC255' (may vary)

s8 = '87cbc255' (may vary)

s9 = '87CBC255' (may vary)

s10 = '87cbc255' (may vary)

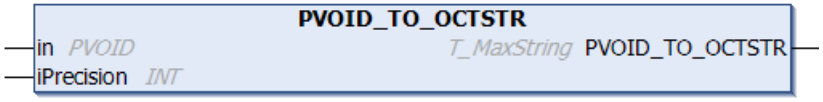
s11 = '0000000087CBC255' (may vary)

s12 = '0000000087cbc255' (may vary)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.61 PVOID_TO_OCTSTR



The function converts the value of a pointer variable of type PVOID into an octal string (basis 8).

FUNCTION PVOID_TO_OCTSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : PVOID;
    iPrecision : INT;
END_VAR
```

in: The pointer variable to be converted.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Example:

```
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 );
```

The result:

s1 = ''

s2 = '0'

s3 = '0000000000000000'

s4 = '20767501125' (may vary)

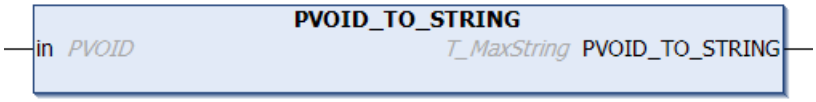
s5 = '20767501125' (may vary)

s6 = '0000020767501125' (may vary)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.62 PVOID_TO_STRING



The function converts the value of a pointer variable of type PVOID into a hexadecimal string (basis 16). The hexadecimal string has the PLC prefix: '16#'. The resolution is fixed at 8 digits in a 32-bit system and 16 digits in 64-bit system.

FUNCTION PVOID_TO_STRING: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in : PVOID;
END_VAR
```

in: The pointer variable to be converted.

Example:

```
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 );
```

The result on a 32-bit system:

s1 = '16#00000000'

s2 = "16#87DE0255' (may vary)

The result on a 64-bit system:

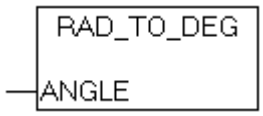
s1 = '16#0000000000000000'

s2 = "16#8734651087DE0255' (may vary)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.63 RAD_TO_DEG



The function converts radian into a degree angle.

FUNCTION RAD_TO_DEG: LREAL

VAR_INPUT

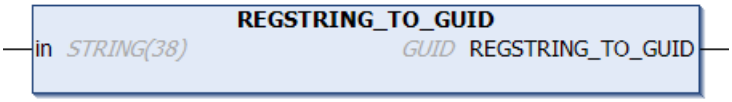
```
VAR_INPUT
    ANGLE : LREAL;
END_VAR
```

ANGLE: Radian to be converted.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.64 REGSTRING_TO_GUID



This function converts a registry GUID string variable (enclosed in curly brackets) into a structured GUID [▶ 303] variable.

FUNCTION REGSTRING_TO_GUID: GUID

VAR_INPUT

```
VAR_INPUT
    in : STRING(38);
END_VAR
```

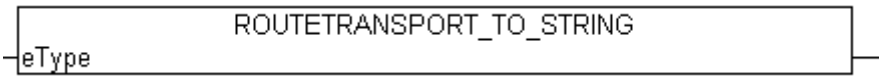
Return value	Meaning
'{xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx}'	No error ('x' is a hexadecimal half-byte)

Return value	Meaning
'{00000000-0000-0000-0000-000000000000}'	No error, GUID has the initial value (all bytes are zero)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.65 ROUTETRANSPORT_TO_STRING



The function converts the AMS message router transport layer ID into a string.

FUNCTION ROUTETRANSPORT_TO_STRING: STRING

VAR_INPUT

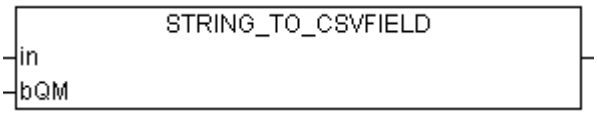
```
VAR_INPUT
    eType : E_RouteTransportType;
END_VAR
```

eType: The transport layer ID to be converted (type: E_RouteTransportType [▶ 301]).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.66 STRING_TO_CSVFIELD



The function converts the value of a PLC string variable to a string data field in CSV format. Single quotation marks in the source string are replaced with double quotation marks. If the bQM parameter is set (QM = quotation marks) the outer quotation marks (around the CSV data field) are also added. If successful the function returns the convert string as the result. The function returns an empty string if an error occurred during the conversion, but only if the source string was not an empty string.

The function is usually used together with the function block FB_CSVMemBufferWriter [▶ 50] to generate data sets in the PLC memory in CSV format. In the next step the memory content can be written to the file.

The source string must not contain binary data. Binary data with the value zero would terminate and truncate the string in the wrong place. To convert binary data please use the function: ARG_TO_CSVFIELD [▶ 242].

FUNCTION STRING_TO_CSVFIELD: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in  : T_MaxString;
    bQM : BOOL;
END_VAR
```

in: Source string whose value is to be converted into a data field in CSV format (type: T_MaxString).

bQM: If this input is TRUE the enclosing quotation marks are added from the result string.

bQM	Description	Source string	Result string	CSV-compliant
FALSE	Source string without enclosing quotation marks should only contain letters and numbers. In this case the source string must not contain any non-printable control character, quotation mark, semicolon, comma (US CSV format) or binary data.			
		'Module_XA5'	'Module_XA5'	Yes
		'123456'	'123456'	Yes
		''	''	Yes
		'A""B'	'A""""B'	No
		'A"B'	'A""B'	No
		';'	';'	No
		'RN'	'RN'	No
		'AB$00CD'	'AB' (string was truncated)	No
TRUE	A source string that is not enclosed in quotation marks should not contain any non-printable control character, quotation mark, semicolon or comma (US CSV format). Binary data are not permitted. '123456' '' 'A""B' 'A"B' ';' 'RN' 'AB$00CD'	'Module_XA5'	'"Module_XA5"'	Yes
		'123456'	'"123456"'	Yes
		''	'""'	Yes
		'A""B'	'"A""""B"'	Yes
		'A"B'	'"A""B"'	Yes
		';'	'";"'	Yes
		'RN'	'"RN"'	Yes
		'AB$00CD'	'"AB"' (String was truncated)	No

Example:

```
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 );
```

The result:

s1 = '"Module_""ALFA_$05""_6"'

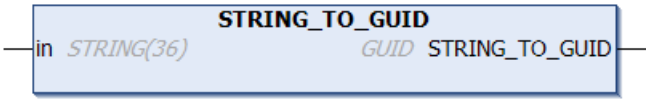
s2 = 'Module_50'

Further information can be found here: Example: Writing/reading of a CSV file [▶ 344].

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.67 STRING_TO_GUID



This function converts a registry GUID string variable (without curly brackets) into a structured GUID [▶ 303] variable.

FUNCTION STRING_TO_GUID: GUID

VAR_INPUT

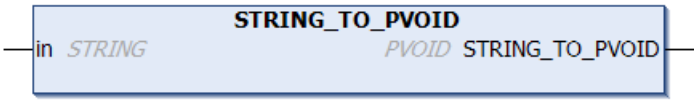
```
VAR_INPUT
    in : STRING(36);
END_VAR
```

Return value	Meaning
'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'	No error (‘x’ is a hexadecimal half-byte)
'00000000-0000-0000-0000-000000000000'	No error, GUID has the initial value (all bytes are zero)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.68 STRING_TO_PVOID



The function converts a string variable into a pointer variable of type PVOID. The function returns zero if the input string contains incorrect characters and cannot be interpreted as an address.

FUNCTION STRING_TO_PVOID: PVOID

VAR_INPUT

```
VAR_INPUT
    in  : STRING;
END_VAR
```

in: String variable to be converted.

Example:

```
PROGRAM MAIN
VAR
    sP1 : STRING := '16#89345678';
    sP2 : STRING := '8#21115053170';
    sP3 : STRING := '2#10001001001101000101011001111000';
    sP4 : STRING := '2301908600';
    sP5 : STRING := '';
    pP1 : PVOID := 0;
    pP2 : PVOID := 0;
    pP3 : PVOID := 0;
    pP4 : PVOID := 0;
    pP5 : PVOID := 0;
END_VAR
```

```
pP1 := STRING_TO_PVOID( sP1 );
pP2 := STRING_TO_PVOID( sP2 );
pP3 := STRING_TO_PVOID( sP3 );
pP4 := STRING_TO_PVOID( sP4 );
pP5 := STRING_TO_PVOID( sP5 );
```

The result:

pP1 = 2301908600

pP2 = 2301908600

pP3 = 2301908600

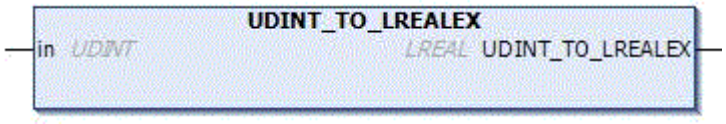
pP4 = 2301908600

pP5 = 0

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.69 UDINT_TO_LREALEX



The conversion of unsigned numbers into floating point numbers of the type LREAL is not supported in TwinCAT 2 on the ARM platform. Unsigned numbers with the highest significant bit set may possibly be implicitly converted into negative floating point numbers. The function described here allows the explicit conversion of the type UDINT into a positive floating point number of the type LREAL in TwinCAT 2 (even if the highest significant bit was set and without compiler warning). You only need this function in order to be able to compile converted TwinCAT 2 projects without changes in TwinCAT 3.

Unsigned numbers of the type UDINT are always (implicitly and explicitly) converted into positive floating point numbers in TwinCAT3. For this reason this function can be dispensed with.

FUNCTION UDINT_TO_LREALEX: LREAL

VAR_INPUT

```
VAR_INPUT
    in : UDINT;
END_VAR
```

Sample:

```
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 value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := UDINT_TO_LREALEX(16#FFFFFFFF)	4294967295	4294967295	4294967295

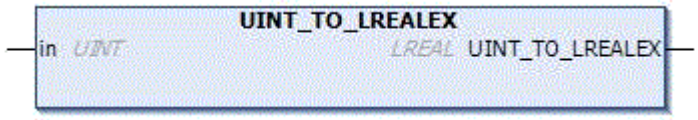
*not supported

**Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.70 UINT_TO_LREALEX



The conversion of unsigned numbers into floating point numbers of the type LREAL is not supported in TwinCAT 2 on the ARM platform. Unsigned numbers with the highest significant bit set may possibly be implicitly converted into negative floating point numbers. The function described here allows the explicit conversion of the type UINT into a positive floating point number of the type LREAL in TwinCAT 2 (even if the highest significant bit was set and without compiler warning). You only need this function in order to be able to compile converted TwinCAT 2 projects without changes in TwinCAT 3.

Unsigned numbers of the type UINT are always (implicitly and explicitly) converted into positive floating point numbers in TwinCAT3. For this reason this function can be dispensed with.

FUNCTION UINT_TO_LREALEX: LREAL

VAR_INPUT

```
VAR_INPUT
    in : UINT;
END_VAR
```

Sample:

```
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_LREALEX(nUint)	+65535	+65535	+65535
fLreal := UINT_TO_LREALEX(UINT#16#FFFF)	+65535	+65535	+65535

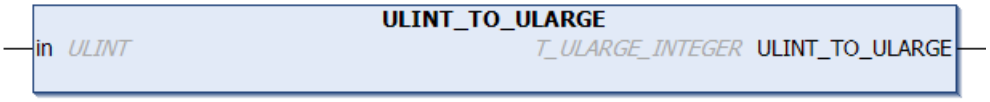
fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := UINT_TO_LREALEX(16#FFFF)	+65535	+65535	+65535

*Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.71 ULINT_TO_ULARGE



The function converts a TwinCAT 3 unsigned 64 bit number (“native” type) into a TwinCAT 2 unsigned 64 bit number (“legacy” type: T_ULARGE_INTEGER [▶ 320]).

FUNCTION ULINT_TO_ULARGE: T_ULARGE_INTEGER

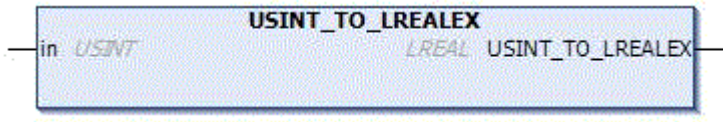
VAR_INPUT

```
VAR_INPUT
    in : ULINT;
END_VAR
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.72 USINT_TO_LREALEX



The conversion of unsigned numbers into floating point numbers of the type LREAL is not supported in TwinCAT 2 on the ARM platform. Unsigned numbers with the highest significant bit set may possibly be implicitly converted into negative floating point numbers. The function described here allows the explicit conversion of the type USINT into a positive floating point number of the type LREAL in TwinCAT 2 (even if the highest significant bit was set and without compiler warning). You only need this function in order to be able to compile converted TwinCAT 2 projects without changes in TwinCAT 3.

Unsigned numbers of the type USINT are always (implicitly and explicitly) converted into positive floating point numbers in TwinCAT3. For this reason this function can be dispensed with.

FUNCTION USINT_TO_LREALEX: LREAL

VAR_INPUT

```
VAR_INPUT
    in : USINT;
END_VAR
```

Sample:

```
PROGRAM MAIN
VAR
    nUsint : USINT := 16#FF;
    fLreal : LREAL := 0.0;
END_VAR
```

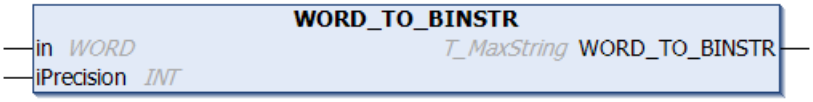
fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := nUsint	+255, Warning 1105*	+255	+255
fLreal := USINT_TO_LREAL(nUsint)	+255, Warning 1105*	+255	+255
fLreal := USINT#16#FF	+255, Warning 1105*	+255	+255
fLreal := 16#FF	+255	+255	+255
fLreal := USINT_TO_LREALEX(nUsint)	+255	+255	+255
fLreal := USINT_TO_LREALEX(USINT#16#FF)	+255	+255	+255
fLreal := USINT_TO_LREALEX(16#FF)	+255	+255	+255

*Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.73 WORD_TO_BINSTR



This function converts a decimal number into a binary string (base 2).

FUNCTION WORD_TO_BINSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : WORD;
    iPrecision : INT;
END_VAR
```

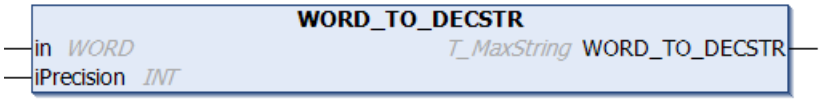
in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.74 WORD_TO_DECSTR



This function converts a decimal number into a decimal string (base 10).

FUNCTION WORD_TO_DECSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : WORD;
    iPrecision : INT;
END_VAR
```

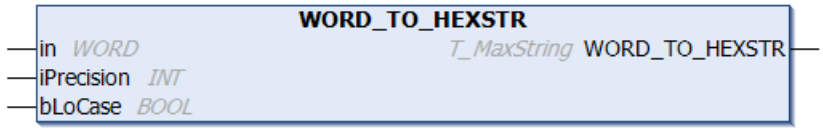
in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.75 WORD_TO_HEXSTR



This function converts a decimal number into a hexadecimal string (base 16).

FUNCTION WORD_TO_HEXSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : WORD;
    iPrecision : INT;
    bLoCase    : BOOL := FALSE;
END_VAR
```

in: The decimal number requiring conversion.

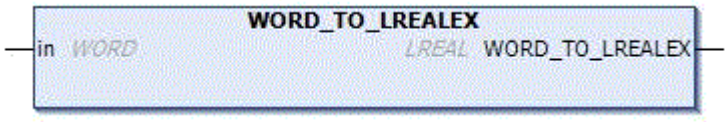
iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

bLoCase: This parameter determines whether lower or upper case letters are used in the conversion. FALSE => "ABCDEF", TRUE => "abcdef".

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.76 WORD_TO_LREALEX



The conversion of unsigned numbers into floating point numbers of the type LREAL is not supported in TwinCAT 2 on the ARM platform. Unsigned numbers with the highest significant bit set may possibly be implicitly converted into negative floating point numbers. The function described here allows the explicit conversion of the type WORD into a positive floating point number of the type LREAL in TwinCAT 2 (even if the highest significant bit was set and without compiler warning). You only need this function in order to be able to compile converted TwinCAT 2 projects without changes in TwinCAT 3.

Unsigned numbers of the type WORD are always (implicitly and explicitly) converted into positive floating point numbers in TwinCAT3. For this reason this function can be dispensed with.

FUNCTION WORD_TO_LREALEX: LREAL

VAR_INPUT

```
VAR_INPUT
    in : WORD;
END_VAR
```

Sample:

```
PROGRAM MAIN
VAR
    nWord  : WORD  := 16#FFFF;
    fLreal : LREAL := 0.0;
END_VAR
```

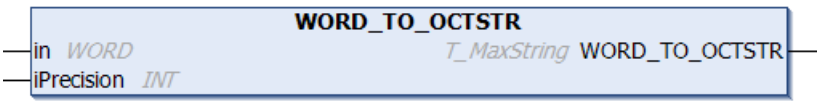
fLreal value	Tc2.x ARM	Tc2.x X86	Tc3.x ARM, X86, X64
fLreal := nWord	+65535, Warning 1105*	+65535	+65535
fLreal := WORD_TO_LREAL(nWord)	+65535, Warning 1105*	+65535	+65535
fLreal := WORD#16#FFFF	+65535, Warning 1105*	+65535	+65535
fLreal := 16#FFFF	+65535	+65535	+65535
fLreal := WORD_TO_LREALEX(nWord)	+65535	+65535	+65535
fLreal := WORD_TO_LREALEX(WORD#16#FFFF)	+65535	+65535	+65535
fLreal := WORD_TO_LREALEX(16#FFFF)	+65535	+65535	+65535

*Conversion of unsigned integer to LREAL is not supported. The value is used as signed instead.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

4.77 WORD_TO_OCTSTR



This function converts a decimal number into an octal string (base 8).

FUNCTION WORD_TO_OCTSTR: T_MaxString

VAR_INPUT

```
VAR_INPUT
    in         : WORD;
    iPrecision : INT;
END_VAR
```

in: The decimal number requiring conversion.

iPrecision: Minimum number of displayed digits (digits). If the actual number of significant digits is less than the *iPrecision* parameter, the resulting string is filled with zeros from the left. If the number of significant digits is greater than the *iPrecision* parameter, the resulting string is not cut off! If the *iPrecision* parameter and the *in* parameter are zero, the resulting string is empty.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5 Data types

5.1 ADSDATATYPEID

ADS data type ID. This data type is used by the function block FB_ReadSymInfoByNameEx [▶ 129], for example.

```
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
```

Value	Meaning
ADST_VOID	Reserved
ADST_INT8	Signed 8 bit integer
ADST_UINT8	Unsigned 8 bit integer
ADST_INT16	Signed 16 bit integer
ADST_UINT16	Unsigned 16 bit integer
ADST_INT32	Signed 32 bit integer
ADST_UINT32	Unsigned 32 bit integer
ADST_INT64	Signed 64 bit integer
ADST_UINT64	Unsigned 64 bit integer
ADST_REAL32	32 bit floating point number
ADST_REAL64	64 bit floating point number
ADST_BIGTYPE	Structured type
ADST_STRING	String type
ADST_WSTRING	Wide character type
ADST_REAL80	Reserved
ADST_BIT	Bit type
ADST_MAXTYPES	Max. available type

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.2 E_AmsLoggerMode

AMS logger control modes. This data type is used by the function block: FB_AmsLogger [▶ 43].

```
TYPE E_AmsLoggerMode :
(
    AMSLOGGER_RUN    := 1,
```

```
    AMSLOGGER_STOP  := 2
);
END_TYPE
```

Value	Meaning
AMSLOGGER_RUN	Starts the AMS Logger
AMSLOGGER_STOP	Stops the AMS Logger

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.3 E_ArgType

Internal argument type ID. This type is used by string format functions/function blocks.

```
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
```

Value	Corresponding PLC data type
ARGTYPE_UNKNOWN	Type is unknown or not initialized
ARGTYPE_BYTE	BYTE (8 bits)
ARGTYPE_WORD	WORD (16 bits)
ARGTYPE_DWORD	DWORD (32 bits)
ARGTYPE_REAL	REAL
ARGTYPE_LREAL	LREAL
ARGTYPE_SINT	SINT
ARGTYPE_INT	INT
ARGTYPE_DINT	DINT
ARGTYPE_USINT	USINT
ARGTYPE_UINT	UINT
ARGTYPE_UDINT	UDINT
ARGTYPE_STRING	String of type: T_MaxString
ARGTYPE_BOOL	BOOL
ARGTYPE_BIGTYPE	Any data structure or byte buffer
ARGTYPE_ULARGE	T_ULARGE_INTEGER or ULINT (unsigned 64-bit integer)
ARGTYPE_UHUGE	T_UHUGE_INTEGER (unsigned 128 bit integer)

Value	Corresponding PLC data type
ARGTYPE_LARGE	T_LARGE_INTEGER or LINT (signed 64 bit integer)
ARGTYPE_HUGE	T_HUGE_INTEGER (signed 128 bit integer)
ARGTYPE_LWORD	LWORD (64 bits)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.4 E_DbgContext

This variable type can be used by protocol blocks. It determines the context of the debug output.

```
TYPE E_DbgContext :
(
    eDbgContext_NONE := 0,(* Not used *)
    eDbgContext_USER := 1,(* Service user *)
    eDbgContext_PROV := 2 (* Service provider *)
);
END_TYPE
```

Value	Meaning
eDbgContext_NONE	Parameter not used
eDbgContext_USER	The debug output was triggered by the service user
eDbgContext_PROV	The debug output was triggered by the service provider

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.5 E_DbgDirection

This variable type can be used by buffer blocks or protocol blocks for configuring the debug output.

```
TYPE E_DbgDirection :
(
    eDbgDirection_OFF := 0,(* Disabled (no debug oputput) *)
    eDbgDirection_IN  := 1,(* Enabled only for incomming data *)
    eDbgDirection_OUT := 2,(* Enabled only for outgoing data *)
    eDbgDirection_ALL := 3(* Enabled for incomming and outgoing data *)
);
END_TYPE
```

Value	Meaning
eDbgDirection_OFF	Disables the debug output
eDbgDirection_IN	Activates output of incoming telegrams
eDbgDirection_OUT	Activates output of outgoing telegrams
eDbgDirection_ALL	Activates output of incoming and outgoing telegrams

Example:

The debug output itself can be realized with the ADSLOGSTR function, for example.

In a ring buffer the debug output could be controlled via the variable in the following way, for example:

- If the value is *eDbgDirection_IN* or *eDbgDirection_ALL* the debug output is triggered if a new value is added to the buffer;

- If the value is *eDbgDirection_out* or *eDbgDirection_ALL* the debug output is triggered if a value is removed from the buffer;

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.6 E_EnumCmdType

Control parameter for enumeration blocks. Not all parameters are used by each enumeration block!

```
TYPE E_EnumCmdType :
(
    eEnumCmd_First := 0,
    eEnumCmd_Next,
    eEnumCmd_Abort
);
END_TYPE
```

Value	Meaning
eEnumCmd_First	Lists the first element
eEnumCmd_Next	Lists the next element
eEnumCmd_Abort	Cancels the listing (closes open handles)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.7 E_HashMode

```
{attribute 'qualified_only'}
{attribute 'strict'}
TYPE E_HashMode :
(
    HASH_MD5 := 1,      // generates 16 bytes hash value
    HASH_SHA1,          // generates 20 bytes hash value
    HASH_SHA256,        // generates 32 bytes hash value
    HASH_SHA384,        // generates 48 bytes hash value
    HASH_SHA512         // generates 64 bytes hash value
) DINT;
END_TYPE
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024.29	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.51.0

5.8 E_LDevType

```
TYPE E_LDevType :
(
    eLDT_Unknown        : UDINT,
    eLDT_Beckhoff       : UDINT,
    eLDT_GenericPC      : UDINT,
    eLDT_TerminalDongle : UDINT,
    eLDT_UsbDongle      : UDINT
);
END_TYPE
```

Value	Meaning
eLDT_Unknown	Device type is unknown.
eLDT_Beckhoff	Beckhoff device (e.g. IPC, CX, …)
eLDT_GenericPC	Third-party PC
eLDT_TerminalDongle	License Key Terminal, e.g. EL6070
eLDT_UsbDongle	License key USB stick, e.g. C9900-L100

5.9 E_LDongleStatus

```
TYPE E_LDongleStatus :
(
    eLDT_Unknown      : UDINT,
    eLDT_OK           : UDINT,
    eLDT_Pending      : UDINT,
    eLDT_Invalid      : UDINT,
    eLDT_NoConnection : UDINT
);
END_TYPE
```

Value	Meaning
eLDT_Unknown	License dongle status unknown
eLDT_OK	License dongle was successfully validated
eLDT_Pending	Validation of the license dongle is running
eLDT_Invalid	License dongle is invalid
eLDT_NoConnection	No connection to the license dongle

5.10 E_LicenseHResult

```
TYPE E_LicenseHResult :
(
    //success
    E_LHR_LicenseOK                : DINT := 0,
    E_LHR_LicenseOK_Pending        : DINT := 16#203,
    E_LHR_LicenseOK_Demo           : DINT := 16#254,
    E_LHR_LicenseOK_OEM            : DINT := 16#255,
    //error
    E_LHR_LicenseNoFound           : 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
```

Value	Meaning
E_LHR_LicenseOK	License is valid
E_LHR_LicenseOK_Pending	Validation of the licensing device (e.g. License Key Terminal) required
E_LHR_LicenseOK_Demo	Trial license is valid
E_LHR_LicenseOK_OEM	OEM license is valid
E_LHR_LicenseNoFound	Missing license
E_LHR_LicenseExpired	License expired
E_LHR_LicenseExceeded	License has too few instances

Value	Meaning
E_LHR_LicenseInvalid	License is invalid
E_LHR_LicenseSystemIdInvalid	Incorrect system ID for the license
E_LHR_LicenseNoTimeLimit	License not limited in time
E_LHR_LicenseTimeInFuture	License problem: Time of issue is in the future
E_LHR_LicenseTimePeriodToLong	License period too long
E_LHR_DeviceException	Exception at system startup
E_LHR_LicenseDuplicated	License data read multiple times
E_LHR_SignatureInvalid	Invalid signature
E_LHR_CertificateInvalid	Invalid certificate
E_LHR_LicenseOemNotFound	OEM license for unknown OEM
E_LHR_LicenseRestricted	License invalid for the system
E_LHR_LicenseDemoDenied	Trial license not allowed
E_LHR_LicensePlatformLevelInv	Invalid platform level for the license

5.11 E_MIB_IF_Type

Management information base interface type.

```
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
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.12 E_NumGroupTypes

Numeric number groups. This data type is used by the function block: FB_EnumStringNumbers [▶ 56], for example.

```
TYPE E_NumGroupTypes :
(
    eNumGroup_Float,
    eNumGroup_Unsigned,
    eNumGroup_Signed
);
END_TYPE
```

Value	Meaning
eNumGroup_Float	Floating-point numbers
eNumGroup_Unsigned	Unsigned numbers
eNumGroup_Signed	Signed numbers

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.13 E_PersistentMode

Mode in which the persistent data should be written. This data type is used by the function block: FB_WritePersistentData [▶ 116].

```
TYPE E_PersistentMode :
(
    SPDM_2PASS     := 0,
    SPDM_VAR_BOOST := 1
);
END_TYPE
```

Value	Meaning
SPDM_2PASS	All data are from the same cycle
SPDM_VAR_BOOST	Data of individual persistent variables are from the same cycle

Example:

See also: Writing of persistent data: system behaviour [▶ 349].

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.14 E_RegValueType

Type IDs for registry values.

```
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
```

Value	Meaning
REG_NONE	No value TYPE
REG_SZ	Unicode null terminated STRING
REG_EXPAND_SZ	Unicode null terminated STRING (with environment variable references)
REG_BINARY	Free form binary
REG_DWORD	32 bit number and REG_DWORD_LITTLE_ENDIAN (same as REG_DWORD)
REG_DWORD_BIG_ENDIAN	32 bit number
REG_LINK	Symbolic Link (unicode)
REG_MULTI_SZ	Multiple Unicode strings
REG_RESOURCE_LIST	Resource list in the resource map
REG_FULL_RESOURCE_DESCRIPTOR	Resource list in the hardware description
REG_RESOURCE_REQUIREMENTS_LIST	-

Value	Meaning
REG_QWORD	64 bit number and REG_QQOERD_LITTLE_ENDIAN (same as REG_QWORD)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.15 E_RouteTransportType

Transport layer used for carrying AMS messages. Currently only TCP/IP is supported as transport layer.

```
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
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.16 E_SBCSType

Windows SBCS (Single Byte Character Set) code page types.

```
TYPE E_SBCSType :
(
    eSBCS_WesternEuropean := 1,
    eSBCS_CentralEuropean := 2
);
END_TYPE
```

Value	Meaning
eSBCS_WesternEuropean	Windows 1252 (default) code page
eSBCS_CentralEuropean	Windows 1251 code page

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.17 E_ScopeServerState

```
TYPE E_ScopeServerState
(
   SCOPE_SERVER_IDLE,
   SCOPE_SERVER_CONNECT,
```

```
   SCOPE_SERVER_START,
   SCOPE_SERVER_STOP,
   SCOPE_SERVER_SAVE,
   SCOPE_SERVER_DISCONNECT,
   SCOPE_SERVER_RESET
);
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.18 E_TimeZoneID

Additional information on the configured time zone of the operating system.

```
TYPE E_TimeZoneID :
(
    eTimeZoneID_Invalid  := -1,
    eTimeZoneID_Unknown  := 0,
    eTimeZoneID_Standard := 1,
    eTimeZoneID_Daylight := 2
);
END_TYPE
```

Value	Meaning
eTimeZoneID_Invalid	The time zone configuration could not be read
eTimeZoneID_Unknown	The time zone configuration could be read, but the standard/summer time information is unknown
eTimeZoneID_Standard	Currently standard time is used
eTimeZoneID_Daylight	Currently summer time is used

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.19 E_TypeFieldParam

String format type field.

```
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: hecadecimal number (upper case characters )*)
    TYPEFIELD_XL, (* x: hecadecimal number (lower case characters )*)
    TYPEFIELD_EU, (* E: float number ( scientific format ) *)
    TYPEFIELD_EL (* e: float number ( scientific format ) *)
);
END_TYPE
```

Value	Meaning
TYPEFIELD_UNKNOWN	Unknown or not initialized
TYPEFIELD_B	b or B: Binary number
TYPEFIELD_O	o or O: octal number
TYPEFIELD_U	u or U: Unsigned decimal number

Value	Meaning
TYPEFIELD_C	c or C: ASCII character
TYPEFIELD_F	f or F: floating-point number (normalized representation)
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: floating-point number (scientific representation)
TYPEFIELD_EL	e: floating-point number (scientific representation)

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.20 FLOAT

LREAL alias type.

```
TYPE FLOAT :LREAL;
END_TYPE
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.21 GUID

System ID.

```
TYPE GUID :
STRUCT
    Data1 : DWORD;
    Data2 : WORD;
    Data3 : WORD;
    Data4 : ARRAY[0..7] OF BYTE;
END_STRUCT
END_TYPE
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.22 OTSTRUCT

Time format for an operating hours counter.

```
TYPE OTSTRUCT :
STRUCT
    wWeek          : WORD;
    wDay           : WORD;
    wHour          : WORD;
    wMinute        : WORD;
    wSecond        : WORD;
```

```
    wMilliseconds : WORD;
END_STRUCT
END_TYPE
```

wWeek: Number of weeks: 0 ~ 65535;

wDay: Number of days: 0 ~ 7;

wHour: Number of hours: 0 ~ 23;

wMinute: Number of minutes: 0 ~ 59;

wSecond: Number of seconds: 0 ~ 59;

wMilliseconds: Number of milliseconds: 0 ~ 999;

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.23 PROFILERSTRUCT

Status information of the profiler function block.

```
TYPE PROFILERSTRUCT :
STRUCT
    LastExecTime     : DWORD;
    MinExecTime      : DWORD;
    MaxExecTime      : DWORD;
    AverageExecTime  : DWORD;
    MeasureCycle     : DWORD;
END_STRUCT
END_TYPE
```

LastExecTime: The most recently measured value for the execution time in [µs].

MinExecTime: The minimum execution time in [µs].

MaxExecTime: The maximum execution time in [µs].

AverageExecTime: The mean execution time for the last 10 measurements in [µs].

MeasureCycle: The number of measurements that have already been carried out.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.24 REMOTEPC

Remote PC configuration entry.

```
TYPE REMOTEPC :
STRUCT
    NetId : T_AmsNetId;
    Name  : STRING(31);
END_STRUCT
ND_TYPE
```

NetId: Network address of the remote PC (type: T_AmsNetID);

Name: The remote PC identifier;

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.25 REMOTEPCINFOSTRUCT

List with several remote PC configuration entries (type: REMOTEPC [▶ 304]).

```
TYPE REMOTEPCINFOSTRUCT : ARRAY[0..99] OF REMOTEPC;
END_TYPE
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.26 ST_AmsRouteEntry

This data type contains information on the configuration of a remote TwinCAT connection.

```
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: Symbolic name of the remote TwinCAT system. This name can be chosen freely. The maximum string length is limited through a constant (default: 31 characters).

sNetID: Network address of the remote TwinCAT system (type: T_AmsNetID).

sAddress: System address in relation to the respective transport layer. If TCP/IP is used as the transport layer the IP address is specified here. The maximum string length is limited through a constant (default: 79 characters).

eTransport: Transport layer used for carrying AMS messages (type: E_RouteTransportType [▶ 301]). Currently only the TCP/IP transport layer is supported.

tTimeout: Timeout time. (currently reserved but not used).

dwFlags: Additional options (currently reserved but not used).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.27 ST_AmsRouteEntryEx

This data type contains information on the configuration of a remote TwinCAT connection.

```
TYPE ST_AmsRouteEntryEx :
STRUCT
    sName          : STRING(MAX_ROUTE_NAME_LEN);
    sNetID         : T_AmsNetId;
    sAddress       : STRING(MAX_ROUTE_ADDR_LEN);
```

```
    sVirtualNetID : T_AmsNetId;
    eTransport    : E_RouteTransportType;
    tTimeout      : TIME;
    dwFlags       : DWORD;
END_STRUCT
END_TYPE
```

sName: Symbolic name of the remote TwinCAT system. This name can be chosen freely. The maximum string length is limited through a constant (default: 31 characters).

sNetID: Network address of the remote TwinCAT system (type: T_AmsNetID).

sAddress: System address in relation to the respective transport layer. If TCP/IP is used as the transport layer the IP address is specified here. The maximum string length is limited through a constant (default: 79 characters).

sVirtualNetID: Virtual network address (type: T_AmsNetID) Saee description ofAmsNAT functionality.

eTransport: Transport layer used for carrying AMS messages (type: E_RouteTransportType [▶ 301]). Currently only the TCP/IP transport layer is supported.

tTimeout: Timeout time. (currently reserved but not used).

dwFlags: Additional options (currently reserved but not used).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (system) >= v3.3.41.0

5.28 ST_CheckLicense

Structure with license information

```
TYPE ST_CheckLicense :
STRUCT
    stLicenseId     : GUID;
    tExpirationTime : TIMESTRUCT;
    sExpirationTime : STRING(80);
    eResult         : E_LicenseHResult;
    nCount          : UDINT;
END_STRUCT
END_TYPE
```

Name	Description
stLicenseId	License ID
tExpirationTime	Expiry date
sExpirationTime	Expiry date
eResult	License status (see E_LicenseHResult [▶ 298])
nCount	Number of instances for this license (0=unlimited)

5.29 ST_DeviceIdentification

```
TYPE ST_DeviceIdentification :
STRUCT
    strTargetType       : STRING(30);
    strHardwareModel    : STRING(10);
    strHardwareSerialNo : STRING(12);
    strHardwareVersion  : STRING(8);
    strHardwareDate     : STRING(12);
    strHardwareCPU      : STRING(20);
    strImageDevice      : STRING(48);
    strImageVersion     : STRING(32);
    strImageLevel       : STRING(32);
    strImageOsName      : STRING(48);
```

```
    strImageOsVersion   : STRING(8);
    strTwinCATVersion   : STRING(4);
    strTwinCATRevision  : STRING(4);
    strTwinCATBuild     : STRING(8);
    strTwinCATLevel     : STRING(20);
    strAmsNetId         : T_AmsNetId;
END_STRUCT
END_TYPE
```

strTargetType: Target system type, e.g. 'CX1000 CE', ...

strHardwareModel: Hardware model, e.g. '1001'.

strHardwareSerialNo: Hardware serial number, e.g. '123'.

strHardwareVersion: Hardware version, e.g. '1.7'.

strHardwareDate: Hardware production date, e.g. '18.8.06'.

strHardwareCPU: Hardware CPU architecture, e.g. 'INTELx86', 'ARM', 'UNKNOWN' or '' (empty string).

strImageDevice: Software platform, e.g. 'CX1000', ...

strImageVersion: Software platform version, e.g. '2.15'.

strImageLevel: Software platform level, e.g. 'HMI'.

strImageOsName: Name of operating system, e.g. 'Windows CE'.

strImageOsVersion: Operating system version, e.g. '5.0'.

strTwinCATVersion: TwinCAT version, e.g. for TwinCAT 2.10.1307: '2'.

strTwinCATRevision: TwinCAT revision, e.g. for TwinCAT 2.10.1307: '10'.

strTwinCATBuild: TwinCAT build, e.g. for TwinCAT 2.10.1307: '1307'.

strTwinCATLevel: Registered TwinCAT level, e.g. 'PLC', 'NC-PTP', 'NC-I', ...

strAmsNetId: TwinCAT AMS-NetID, e.g. '5.0.252.31.1.1'.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.30 ST_DeviceIdentificationEx

```
TYPE ST_DeviceIdentificationEx :
STRUCT
    strTargetType       : STRING(30);
    strHardwareModel    : STRING(16);
    strHardwareSerialNo : STRING(16);
    strHardwareVersion  : STRING(8);
    strHardwareDate     : STRING(12);
    strHardwareCPU      : STRING(20);
    strImageDevice      : STRING(48);
    strImageVersion     : STRING(32);
    strImageLevel       : STRING(32);
    strImageOsName      : STRING(48);
    strImageOsVersion   : STRING(8);
    strTwinCATVersion   : STRING(4);
    strTwinCATRevision  : STRING(4);
    strTwinCATBuild     : STRING(8);
    strTwinCATLevel     : STRING(20);
    strAmsNetId         : T_AmsNetId;
END_STRUCT
END_TYPE
```

strTargetType: Target system type, e.g. 'CX1000 CE', ...

strHardwareModel: Hardware model, e.g. '1001'.

strHardwareSerialNo: Hardware serial number, e.g. '123'.

strHardwareVersion: Hardware version, e.g. '1.7'.

strHardwareDate: Hardware production date, e.g. '18.8.06'.

strHardwareCPU: Hardware CPU architecture, e.g. 'INTELx86', 'ARM', 'UNKNOWN' or '' (empty string).

strImageDevice: Software platform, e.g. 'CX1000', ...

strImageVersion: Software platform version, e.g. '2.15'.

strImageLevel: Software platform level, e.g. 'HMI'.

strImageOsName: Name of operating system, e.g. 'Windows CE'.

strImageOsVersion: Operating system version, e.g. '5.0'.

strTwinCATVersion: TwinCAT version, e.g. for TwinCAT 2.10.1307: '2'.

strTwinCATRevision: TwinCAT revision, e.g. for TwinCAT 2.10.1307: '10'.

strTwinCATBuild: TwinCAT build, e.g. for TwinCAT 2.10.1307: '1307'.

strTwinCATLevel: Registered TwinCAT level, e.g. 'PLC', 'NC-PTP', 'NC-I', ...

strAmsNetId: TwinCAT AMS-NetID, e.g. '5.0.252.31.1.1'.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.31 ST_FileAttributes

File or directory attributes.

```
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: The file or directory only has read access. The file can be read by applications, but it cannot be written to or deleted. In the case of directories, these cannot be deleted by applications.

bHidden: The file or directory is hidden and is not shown in a standard listing.

bSystem: The file or directory is part of the operating system or is exclusively used by the operating system.

bDirectory: This attribute can be used to identify a directory.

bArchive: The file or directory belongs to the archive. Applications use this attribute in order to tag files for backup or deletion.

bDevice: Reserved.

bNormal: The file or directory has no other attributes set. This attribute is only valid if is used exclusively.

bTemporary: The file is only used temporarily for storing data.

bSparseFile: The file is a slimmed-down file.

bReparsePoint: A "reparse point" was associated with the file or directory.

bCompressed: The file or directory is compressed. The file contains compressed data. In the case of a directory, compacting is active by default for new files or subdirectories.

bOffline: The file is not always available.

bNotContentIndexed: The file is not indexed by the indexing service.

bEncrypted: The file or directory is encrypted.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.32 ST_FileRBufferHead

Ring buffer file header status. This structure is used by the function block FB_FileRingBuffer [▶ 59]. This structure is read when the ring buffer file opened and saved in the ring buffer file when it is closed. This structure is always updated when data sets are read/written.

```
TYPE ST_FileRBufferHead :
STRUCT
    status   : DWORD := 0;(* buffer status flags Bit 0 = 1 => Opened, Bit 0 = 0 => Closed, Bit 1 = 1
file corrupted, all other bits are reserved *)
    access   : UDINT := 0;(* access counter, increments every time the buffer is reopened *)
    nID      : UDINT := 0;(* user defined value *)
    cbBuffer : UDINT := 16#100000;(* max. buffer size (1MB) *)
    nCount   : UDINT := 0;(* number of fifo entries *)
    cbSize   : UDINT := 0;(* current (used) file buffer data byte length *)
    ptrFirst : UDINT := 0;(* seek pointer start position of first (oldest) buffer entry *)
    ptrLast  : UDINT := 0;(* seek pointer end position of last (newest) buffer entry *)
    rsrv0    : UDINT := 0;(* reserved *)
    rsrv1    : UDINT := 0;(* reserved *)
    rsrv2    : UDINT := 0;(* reserved *)
    rsrv3    : UDINT := 0;(* reserved *)
END_STRUCT
END_TYPE
```

status: Status flags. Bit 0 = 1 => file is open, bit 0 = 0 => file is closed. Bit 1 = 1 => file is corrupt (was not closed properly, or the maximum buffer size does not match).

acces: Access counter. This counter is incremented whenever the file is opened.

nID: User-defined 32-bit value.

cbBuffer: Maximum ring buffer file size.

nCount: Current number of stored data sets.

cbSize: Current number of stored data bytes.

ptrFirst: File pointer position of the oldest data set.

ptrLast: File pointer position of the latest data set.

rsrv0..rsrv3: Reserved.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.33 ST_FindFileEntry

This data type is used by the file search function blocks: FB_EnumFindFileEntry [▶ 51] and FB_EnumFindFileList [▶ 52].

```
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: Zero-terminated string with the name of the file or directory (type: T_MaxString).

sAlternateFileName: Zero-terminated string with the alternative name of the file or directory in conventional 8.3 format (filename.ext).

fileAttributes: Structure with file/directory attributes (type: ST_FileAttributes [▶ 308]).

fileSize: Byte size of the file (64-bit number, type: T_ULARGE_INTEGER [▶ 320]).

creationTime: The structure variable indicates when the file or directory was created (type: T_FILETIME [▶ 317]).

lastAccessTime: For a file the structure indicates when it was last accessed (read or write) (type: T_FILETIME [▶ 317]). For a directory the structure indicates when it was created.

lastWriteTime: For a file the structure indicates the time of the last write access (type: T_FILETIME [▶ 317]). For a directory the structure indicates when it was created.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.34 ST_IPAdapterHwAddr

Physical address (MAC).

```
TYPE ST_IPAdapterHwAddr :
STRUCT
    length : UDINT := 0;
    b      : ARRAY[0..MAX_ADAPTER_ADDRESS_LENGTH] OF BYTE;
END_STRUCT
END_TYPE
```

length: Byte length of the physical hardware address.

b: MAC address bytes.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.35 ST_IPAdapterInfo

Network adapter information.

```
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: This variable is currently **only used under Windows CE**! If TRUE TwinCAT uses the network adapter as default adapter.

sAdapterName: Adapter name as string.

sDescription: Adapter description as string.

physAddr: Physical hardware address. (Type: ST_IPAdapterHwAddr [▶ 310])

dwIndex: Internal adapter system index.

eType: Adapter type (type: E_MIB_IF_Type [▶ 299]).

sIpAddr: IP address (type: T_Ipv4Addr).

sSubNet: IP network mask (type: T_Ipv4Addr).

sDefGateway: IP address of the default gateway (type: T_Ipv4Addr).

bDhcpEnabled: Indicates whether DHCP was activated for this adapter or not.

sDhcpSrv: IP address of the DHCP server (type: T_Ipv4Addr).

bHaveWins: Indicates whether the Windows Internet Name Service (WINS) is used or not.

sPrimWinsSrv: IP address of the primary WINS server (type: T_Ipv4Addr).

sSecWinsSrv: IP address of the secondary WINS server (type: T_Ipv4Addr).

tLeaseObt: Indicates when the IP address was “hired" by the DHCP server (UTC).

tLeaseExp: Indicates how long the IP address can be “rented” by the DHCP server before an ”extension" has to be requested by the DHCP server (UTC).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.36 ST_LicenseDongle

Structure with identification data of all connected license dongles

```
TYPE ST_LicenseDongle :
STRUCT
    stAmsAddr     : AMSADDR;
    eDevType      : E_LDevType;
    nFlags        : UDINT;
    eDongleStatus : E_LDongleStatus;
    nSerialNo     : UDINT;
    nReserved1    : UDINT;
    nReserved2    : UDINT;
END_STRUCT
END_TYPE
```

Name	Description
stAmsAddr	Network ID (AmsNetId and port) of the license dongle
eDevType	See E_LDevType [▶ 297]
nFlags	0: statically configured 1: dynamic dongle
eDongleStatus	Validation status of the license dongle (see E_LDongleStatus [▶ 298])
nSerialNo	ID number of the dongle / License Key Terminal
nReserved1	Reserved for future use
nReserved2	Reserved for future use

5.37 ST_ReadEvent

Information about a message read by means of FB_AdsReadEvents [▶ 40].

```
TYPE ST_ReadEvent :
STRUCT
    nSourceId       : UDINT;
    nEventId        : UDINT;
    nClass          : DWORD;
    nConfirmState   : DWORD;
    nResetState     : DWORD;
    sSource         : STRING(255);
    sDate           : STRING(23);
    sTime           : STRING(23);
    sComputer       : STRING;
    sMessageText    : STRING(255);
    bQuitMessage    : BOOL;
    bConfirmable    : BOOL;
END_STRUCT
END_TYPE
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.38 ST_SplittedBIC

The structure ST_SplittedBIC can be filled via the function F_SplitBIC [▶ 265] of the Tc2_Utilities using the Beckhoff Identification Conde (BIC). The identifiers and data sizes of the substrings used to decompose the BIC are given as comments at the structure elements.

```
TYPE ST_SplittedBIC :
STRUCT
    sItemNo : STRING(6);  //1, '1P' 8(2+6)
    sBTN : STRING(8); //2, 'SBTN' 12(4+8)
    sDescription : STRING(30); //3, '1K' 32(2+30)
    sQuantity : STRING(5); //4, 'Q' 6(1+5)
    sBatchNo : STRING(12); //5, '2P' 14(2+12)
```

```
    sIdSerialNo : STRING(9); //6, '51S' 12(3+9)
    sVariantNo : STRING(9); //7, '30P' 12(3+9)
    sDataCode : STRING(6); //8, '9D' 8(2+6)
    sOrderBatchNo : STRING(12); //9, '1T' 14(2+12)
    sSerialNo : STRING(20); //10, '52S' 23(3+20)
    sPackUnitQuantity : STRING(5); //11, '4QPU' 9(4+5)
    sDataElement : STRING(11); //12, '6D' 13(2+11)
    sMoistSensLevel : STRING(4); //13, 'Z' 5(1+4)
    sPlatingMaterial : STRING(2); //14, 'E' 3(1+2)
    sManufacturer : STRING(9); //15, '12V' 12(3+9)
    sCountryOfOrigin : STRING(2); //16, '4L' 4(2+2)
    sCustomSpecItemNo : STRING(16); //17, 'P' 17(1+16)
    sUndefined : STRING(819); //remaining undefined BIC string (1023 + /0)
END_STRUCT
END_TYPE
```

Name	Description
sItemNo	Item number, length max. 6 characters, e.g. "072222"
sBTN	Beckhoff Traceability Number (BTN), length max. 8 characters, e.g. "k4p562d7"
sDescription	Description, length max. 30 characters, e.g. "KL9010"
sQuantity	Quantity, length max. 5 characters, e.g. "1"
sBatchNo	Batch number, length max. 12 characters, e.g. "401503180016"
sIdSerialNo	ID/serial number, length max. 9 characters, e.g. "678294104"
sVariantNo	Variant, length max. 9 characters, e.g. "000048443"
sDataCode	Supplier-specific data code, length max. 6 characters, e.g. "G0118"
sOrderBatchNo	Supplier-specific order/batch number, length max. 12 characters, e.g. "FA12345678"
sSerialNo	Supplier-specific serial number, length max. 20 characters, e.g. "2304853-1-004"
sPackUnitQuantity	Packaging unit quantity, length max. 5 characters, e.g. "1000"
sDataElement	Date Code YYYYMMDD, specified, length max. 11 characters, e.g. "20190315146"
sMoistSensLevel	MS dry level, length max. 4 characters, e.g. "MSL3"
sPlatingMaterial	Coating material (according to JEDEC J-STD-609), length max. 2 characters, e.g. "e1"
sManufacturer	Manufacturer (DUNS number), length max. 9 characters, e.g. "313291892"
sCountryOfOrigin	Country of origin, length max. 2 characters, e.g. "CN"
sCustomSpecItemNo	Customer-specific item number, length max. 16 characters, e.g. "0000000107353052"
sUndefined	Unrecognized remaining string of the BIC (length max. 819 characters), e.g. after extension of the BIC by additional elements, if necessary check if new version of the Tc2_Utilities library is available

5.39 ST_TcOnlineLicenseInfoDataEx

```
TYPE ST_TcOnlineLicenseInfoDataEx :
STRUCT
    stLicenseId       : GUID
    stLicenseName     : STRING(80);
    tExpirationTime   : TIMESTRUCT;
    sExpirtaionTime   : STRING(80);
    nMaxCount         : UDINT;
    nUsedCount        : UDINT;
    eResult           : E_LicenseHResult;
    nVolumeNo         : UDINT;
    nOptInfo          : WORD;
    nRestriction      : WORD;
    bOemLicense       : BOOL;
    bBeckhoffLicense  : BOOL;
    bBeckhoffPC       : BOOL;
    bEtherCATDongle   : BOOL;
    bUSBDongle        : BOOL;
    bGenDevTypeLic    : BOOL;
END_STRUCT
END_TYPE
```

Name	Description
stLicenseId	License ID
stLicenseName	License name
tExpirationTime	Expiry date
sExpirationTime	Expiry date
nMaxCount	Maximum number of instances that can be used for this license (0=unlimited)
nUsedCount	Number of license instances used
eResult	License status (see E_LicenseHResult [▶ 298])
nVolumeNo	(only for volume licenses): Volume bundle number 0: no bundle number
nOptInfo	(internal)
nRestriction	License has restrictions on usage
bOemLicense	OEM license
bBeckhoffLicense	Beckhoff license
bBeckhoffPC	License for Beckhoff hardware (IPC, CX etc.)
bEtherCATDongle	License is for EtherCAT License Key Terminal, e.g. EL6070
bUSBDongle	License is for USB license dongle, e.g. C9900-L100
bGenDevTypeLic	License is for the device type

5.40 ST_TcOnlineLicensesInfoData

License information.

```
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: Defines the License ID as GUID [▶ 303].

sLicenseName: Name of the license as a string.

tExpirationTime: Indicates the expiration time of the license (type: TIMESTRUCT [▶ 321]).

sExpirationTime: Indicates the expiration time of the license as a string.

nMaxCount: Indicates the maximum number of permitted instances, if the license concerned is one that contains a limitation of instances (e.g. TC3 NC PTP Axes Pack 25).

nUsedCount: Indicates the number of instances used, if the license concerned is one that contains a limitation of instances.

eResult: Outputs an error code for this license as an HResult enumeration. (An error is represented by a negative value here).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0.4018	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) v3.3.9.0 or higher

5.41 ST_TcRouterStatusInfo

TwinCAT router status information.

```
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
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.42 ST_TimeZoneInformation

Time zone information. The standard time is also referred to as winter time. The bias parameter can also have negative values.

```
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: Defines the current difference between the local time and the UTC time in minutes. UTC = local time + bias.

standardName: Name of the standard time as string.

standardDate: This structure contains information on the transition from summer time to standard time (type: TIMESTRUCT [▶ 321]). The structure parameters *wMonth* is zero if this value is not used. If this parameter is used, the *daylightDate*-parameter must also be used. In order to be able to configure *standardDate,* set the *wYear* parameter to zero, select the required day of the week for *wDayOfWeek* and select a value between 1 and 5 for *wDay* (week of the month, 5 is the last week).

standardBias: Time difference in minutes for calculating the local time during standard time. This value is usually zero.

daylightName: Name of the summer time as string.

daylightDate: This structure contains information on the transition from standard time to summer time (type: TIMESTRUCT [▶ 321]). The structure parameters *wMonth* is zero if this value is not used. If this parameter is used, the *standardDate*-parameter must also be used. In order to be able to configure *daylightDate,* set the *wYear* parameter to zero, select the required day of the week for *wDayOfWeek* and select a value between 1 and 5 for *wDay* (week of the month, 5 is the last week).

daylightBias: Time difference in minutes for calculating the local time during summer time.

Example:

See: FB_SetTimeZoneInformation [▶ 108].

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.43 SYMINFOSTRUCT

TwinCAT PLC symbol information.

```
TYPE SYMINFOSTRUCT :
STRUCT
    symEntryLen  : UDINT;
    idxGroup     : UDINT;
    idxOffset    : UDINT;
    byteSize     : UDINT;
    adsDataType  : ADSDATATYPEID;
    symDataType  : T_MaxString;
    symComment   : T_MaxString;
END_STRUCT
END_TYPE
```

symEntryLen: The actual length in bytes of the symbol entry in the symbol table. The symbols are stored in a symbol table. The length of the individual entries is variable, and depends on the length of the symbol name, the type identifier and the comment.

idxGroup: The index group of the symbolic variables;

idxOffset: The index offset of the symbolic variables;

byteSize: The amount of memory, in bytes, actually occupied by the value of the symbolic variables. A boolean PLC variable, for instance, occupies one byte, while a string with 20 characters in fact occupies 21 bytes (20 bytes for characters plus one byte for the zero marking the end of the string);

adsDataType: The ADS data type ID. (Type: ADSDATATYPEID [▶ 294]) This type identifier is used in ADS access to symbolic variables. All PLC structures and arrays (self-defined data types) have the ADS data type ID: ADST_BIGTYPE and cannot be identified via this data type constant. In order to be able to identify the user-defined data types, use the *symDataType* variable, or read the base type of the individual variables in the structure.

symDataType: The data type identification of the symbolic variable as a string. For instance this might be the type name of a PLC data structure defined by the user (type: T_MaxString, max. 255 characters).

symComment: A comment on the symbolic variable that the user has added to the PLC variable definition line (type: T_MaxString, max. 255 characters).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.44 T_Arg

Argument type for string format functions/function blocks.

```
TYPE T_Arg :
STRUCT
    eType : E_ArgType := ARGTYPE_UNKNOWN;
    cbLen : UDINT     := 0;
    pData : PVOID     := 0;
END_STRUCT
END_TYPE
```

eType: Data type ID (type: E_ArgType [▶ 295]).

cbLen: Number of bytes allocated in the memory.

pData: Address pointer.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.45 T_FILETIME

Variables of this type are 64-bit numbers. The value corresponds to the number of 100-nanosecond intervals since 1 January 1601 (UTC).

```
TYPE T_FILETIME :
STRUCT
    dwLowDateTime  : DWORD;
    dwHighDateTime : DWORD;
END_STRUCT
END_TYPE
```

dwLowDateTime: Lower 32 bits.

dwHighDateTime: Upper 32 bits.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.46 T_FILETIME64

Variables of this type are 64-bit numbers. The value corresponds to the number of 100-nanosecond intervals since 1 January 1601 (typically UTC).

```
TYPE T_FILETIME64 : ULINT; END_TYPE
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.4024	PC or CX (x86, x64, ARM)	Tc2_Utilities (System) >= 3.3.44.0

5.47 T_FIX16

Variables of this type represent a signed 16 bit fixed-point number. This data type is often used by systems that have no FPU unit (e.g.: micro controllers or devices from the area of telecontrol). If for example data in fixed point number format has to be transferred via serial interface, then these data have to be converted to a suitable format.

The number of decimal places is chosen as per the required numerical range and resolution. For 15 decimal places it is possible to display fixed-point numbers in the range: -1..1-2^15. This corresponds approximately to the floating-point number range: -1..0.999969482421875.

Unlike the floating-point numbers the resolution of the fixed-point numbers is constant over the complete numerical range. Unfortunately the fixed-point numbers have a smaller number range for display. Care is to be taken with mathematical operations that can generate a positive or negative overflow.

```
TYPE T_FIX16 :
STRUCT
    value  : INT := 0;
    n      : WORD(0..15);
```

```
    status : DWORD := 0;
END_STRUCT
END_TYPE
```

value: This member variable contains the actual value of the fixed-point number (16 bits before and after the decimal point).

n: Number of decimal places. Permitted range: 0..15. The highest value bit is reserved for the sign bit.

status: Status flags (reserved, not used at the moment).

Example 1:

An A/D-C supplies values as signed 16 bit fixed-point numbers with 15 decimal places. These measurement values are imported into the PLC and should be converted to LREAL data type.

```
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 );
```

Example 2:

The parameters of a micro controller are signed 16 bit fixed-point numbers with 8 decimal places. The LREAL parameters in the PLC should be converted to this format.

```
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 );
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.48 T_HashTableEntry

A hash table entry/element.

```
TYPE T_HashTableEntry :
STRUCT
    key   : DWORD := 0;(* Entry key *)
    value : PVOID := 0;(* Entry value *)
END_STRUCT
END_TYPE
```

key: Key (32-bit unsigned number or 32-bit pointer).

value: Value (may be 32/64-bit unsigned number or pointer).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.49 T_HHASHTABLE

A hash table handle. The hash table handle is used by the function block: FB_HashTableCtrl [▶ 79].

```
TYPE T_HHASHTABLE :
STRUCT
    nCount : UDINT := 0;
    nFree  : UDINT := 0;
END_STRUCT
END_TYPE
```

nCount: Number of elements occupied.

nFree: Number of free elements.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.50 T_HLINKEDLIST

A Linked List Handle. The Linked List Handle is used by the function block FB_LinkedListCtrl [▶ 87].

```
TYPE T_HLINKEDLIST :
STRUCT
    nCount : UDINT := 0;
    nFree  : UDINT := 0;
END_STRUCT
END_TYPE
```

nCount: Number of elements occupied.

nFree: Number of free elements.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.51 T_HUGE_INTEGER

Variables of this type represent a TwinCAT 2 signed 128-bit number (“legacy” type).

```
TYPE T_HUGE_INTEGER :
STRUCT
   qwLowPart  : T_ULARGE_INTEGER;
   qwHighPart : T_ULARGE_INTEGER;
END_STRUCT
END_TYPE
```

qwLowPart: Lower 64 bits.

qwHighPart: Upper 64 bits.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.52 T_LARGE_INTEGER

Variables of this type represent a TwinCAT 2 signed 64-bit number (“legacy” type).

```
TYPE T_LARGE_INTEGER :
STRUCT
    dwLowPart  : DWORD;
    dwHighPart : DWORD;
END_STRUCT
END_TYP
```

dwLowPart: Lower 32 bits.

dwHighPart: Upper 32 bits.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.53 T_LinkedListEntry

Variables of this type represents a node/element of a linked list.

```
TYPE T_LinkedListEntry :
STRUCT
    value : PVOID := 0;
END_STRUCT
END_TYPE
```

value: Value (may be 32/64-bit unsigned number or pointer).

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.54 T_UHUGE_INTEGER

Variables of this type represent a TwinCAT 2 unsigned 128-bit number (“legacy” type).

```
TYPE T_UHUGE_INTEGER :
STRUCT
    qwLowPart  : T_ULARGE_INTEGER;
    qwHighPart : T_ULARGE_INTEGER;
END_STRUCT
END_TYPE
```

qwLowPart: Lower 64 bits.

qwHighPart: Upper 64 bits.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.55 T_ULARGE_INTEGER

Variables of this type represent a TwinCAT 2 unsigned 64-bit number (“legacy” type).

```
TYPE T_ULARGE_INTEGER :
STRUCT
    dwLowPart  : DWORD;
    dwHighPart : DWORD;
END_STRUCT
END_TYPE
```

dwLowPart: Lower 32 bits.

dwHighPart: Upper 32 bits.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

5.56 TIMESTRUCT

Time in system time format.

```
TYPE TIMESTRUCT
STRUCT
   wYear         : WORD;
   wMonth        : WORD;
   wDayOfWeek    : WORD;
   wDay          : WORD;
   wHour         : WORD;
   wMinute       : WORD;
   wSecond       : WORD;
   wMilliseconds : WORD;
END_STRUCT
END_TYPE
```

wYear: the year: 1970 ~ 2106;

wMonth: the month: 1 ~ 12 (January = 1, February = 2, etc.);

wDayOfWeek: the day of the week: 0 ~ 6 (Sunday = 0, Monday = 1 etc.);

wDay: the day of the month: 1 ~ 31;

wHour: hour: 0 ~ 23;

wMinute: minute: 0 ~ 59;

wSecond: second: 0 ~ 59;

wMilliseconds: millisecond: 0 ~ 999;

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

6 Global constants

6.1 Library version

All libraries have a certain version. The version is indicated in the PLC library repository, for example. A global constant contains the information about the library version:

Global_Version

```
VAR_GLOBAL CONSTANT
    stLibVersion_Tc2_Utilities : ST_LibVersion;
END_VAR
```

stLibVersion_Tc2_Utilities: Version number of the Tc2_Utilities library (type: ST_LibVersion).

To check whether the version you have is the version you need, use the function F_CmpLibVersion (defined in Tc2_System library).

All other options for comparing library versions, which you may know from TwinCAT 2, are outdated.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

7 Global variables

VAR_GLOBAL

Name	Type	Value	Use	Meaning
MAX_AVERAGE_MEASURES	INT	10	Profiler [▶ 133]	Number of averaged measurement readings. Possible values: 2..100
GLOBAL_FORMAT_HASH_PREFIX_TYPE	E_HashPrefixTypes	HASHPREFIX_IEC	FB_FormatString [▶ 64], F_FormatArgToStr [▶ 261]	Standard IEC prefix for binary, octal or hexadecimal formatting
GLOBAL_SBCS_TABLE	E_SBCSType [▶ 301]	eSBCS_WesternEuropean	F_ToLCase [▶ 266], F_ToUCase [▶ 267]	Windows SBCS (Single Byte Character Set) Code Page Table
GLOBAL_DCF77_PULSE_SPLIT	TIME	T#140ms	DCF77_TIME [▶ 32]	Pulse length. 0 == pulses < 140ms, 1 == pulses > 140
GLOBAL_DCF77_SEQUENCE_CHECK	BOOL	FALSE	DCF77_TIME [▶ 32]	Plausibility check of two consecutive telegrams: TRUE= Enabled, FALSE = Disabled.
DEFAULT_CSV_FIELD_SEP	BYTE	16#2C	FB_CSVMemBufferWriter [▶ 50], FB_CSVmemBufferReader [▶ 48]	Data field separator. semicolon= 16#3B => german field separator, comma = 16#2C => US field separator

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

8 Samples

8.1 Example: Communication BC/BX<->PC/CX (F_SwapRealEx)

The example demonstrates the application of the F_SwapRealEx [▶ 265] function. This example contains two components: TwinCAT 2.xx BC/BX (Bus Terminal Controller) application and TwinCAT 3.xx PC/CX (x86) application. The PC/CX application reads/writes a structure variable from/into the flag area of the BC/BX. The structure variable contains REAL elements. These have to be converted to the right format before they are used on the PC/CX or transferred to the BC/BX.

Here you can unpack the complete sources:

TwinCAT 2.xx - BC/BX (Bus Terminal Controller) application/project file: https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_Utilities/Resources/803333131/.zip

TwinCAT 3.xx - PC/CX (x86, x64, ARM) application/archive file: https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_Utilities/Resources/803336971/.zip

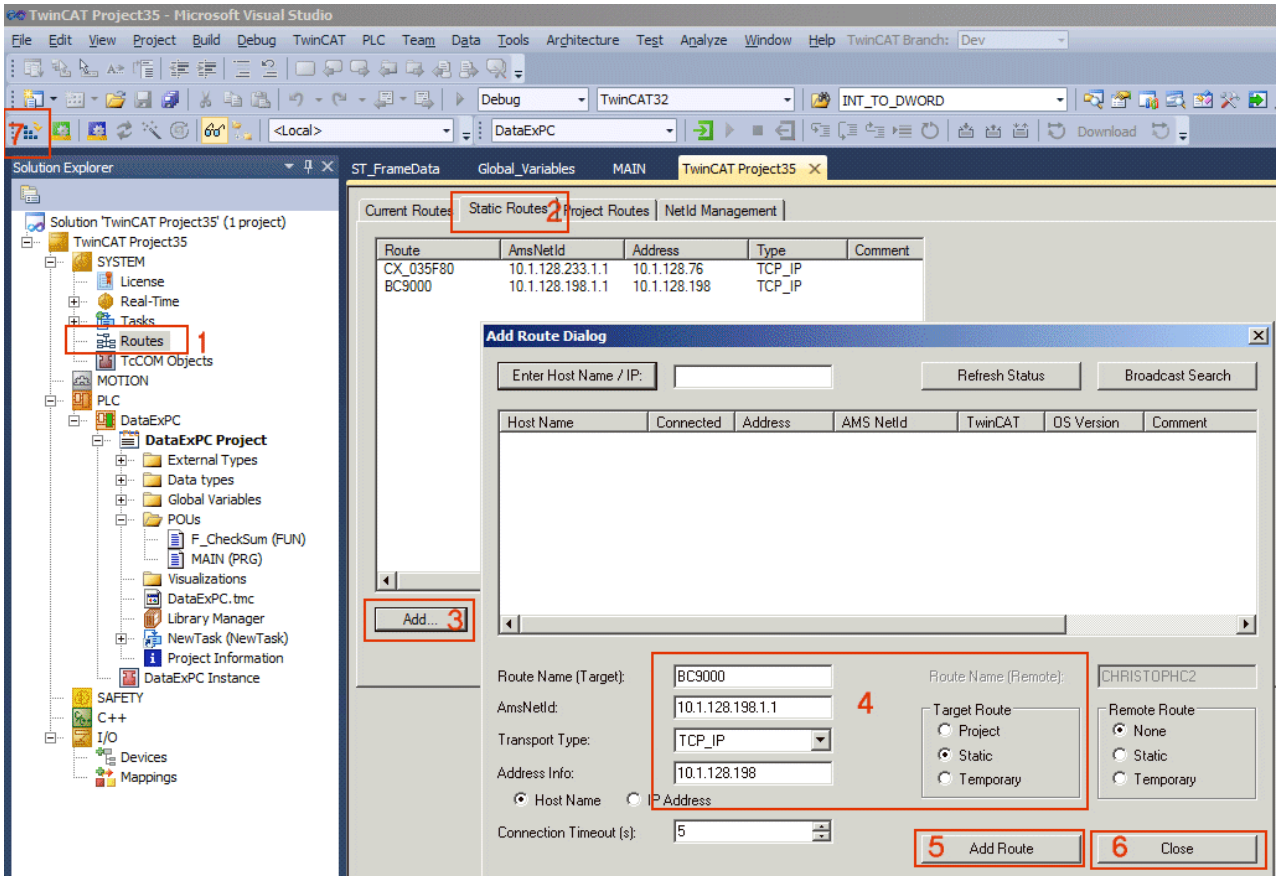
System requirements:

- TwinCAT 2.xx PLC (required for downloading the BC/BX application) + BC/BX hardware (e.g. BC9000);
- TwinCAT 3.xx engineering and runtime system (required for downloading the PC/CX application);

Download a project

Use the TwinCAT 2.xx PLC to download the BC/BX application into the runtime system of a Bus Terminal Controller (e.g. BC9000). Create a boot project and start the PLC. In the next step use TwinCAT 3.xx to create a new XAE project. Import the archive file into TwinCAT XAE with a right mouse click on the *PLC* node, then *Add existing item*.

To enable access to the BC/BX (Bus Terminal Controller) via ADS, it has to be entered as a device in the list of TwinCAT AMS route connections (routes). Create a new static route (follow the steps shown in the diagram). The AmsNetID and IP address of the BC/BX have to be configured accordingly (remember the AmsNetID in the PLC program code).



Important note!

The BC/BX (Bus Terminal Controller) and PC/CX (x86, x64, ARM) have different memory alignment (data alignment). For data exchange BC/BX <-> PC/CX please define structures with 8-byte memory alignment.

- TwinCAT 2.xx + PC/CX (x86) platform => data structures have 1-byte memory alignment;
- TwinCAT 2.xx + CX (ARM) platform => data structures have 4-byte (DWORD) memory alignment;
- TwinCAT 2.xx + BC/BX (Bus Terminal Controller) platform => data structures have 2-byte (WORD) memory alignment;
- TwinCAT 3.xx + PC/CX (x86, x64, ARM) platform => data structures have 8-byte memory alignment;

The structure variable definition used on both systems:

```
(* 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 (Bus Terminal Controller) application

After each write access from the PC/CX the data length and checksum is checked. New random values for read access are then generated, which are also associated with a simple checksum.

```
PROGRAM MAIN
VAR
    stRxFrame AT%MB500   : ST_FrameData;(* Data transported from PC/CX (x86) to BC/
BX (Bus Terminal Controller) *)
    stTxFrame AT%MB0     : ST_FrameData;(* Data transported from BC/
BX (Bus Terminal Controller) to PC/CX (x86) *)
    nReceivedFrame       : UDINT;
    i                    : INT;
    nRxErrors            : UDINT;
END_VAR
```

```
(* New frame from PC/CX received? *)
IF stRxFrame.nTxFrames <> nReceivedFrame THEN
    (* Frame length OK? *)
    IF stRxFrame.nFrameSize = SIZEOF( stRxFrame) THEN
        (* Checksum OK? *)
        IF stRxFrame.nCRC = F_CheckSum( ADR( stRxFrame), SIZEOF( stRxFrame) - 1 ) THEN (* => OK *)
            (* Create/modify the tx data *)
            stTxFrame.nFrameSize  := SIZEOF( stTxFrame);(* Set frame byte size *)
            stTxFrame.nTxFrames   := stTxFrame.nTxFrames + 1;(* Increment the send frame number *)
            stTxFrame.nRxFrames   := stRxFrame.nTxFrames;(* Report the received frame number *)
            stTxFrame.bEnable     := NOT stRxFrame.bEnable;(* Toggle bool flag *)
            stTxFrame.nCounter    := stTxFrame.nCounter + 1;(* Send some counter value *)
            stTxFrame.sMsg        := CONCAT( 'Message from BC/
BX, counter:', DWORD_TO_STRING( stTxFrame.nCounter ) );(* Create any string message *)
            stTxFrame.fU          := stRxFrame.fU + 10.0;(* Modify some floating point values *)
            stTxFrame.fV          := stRxFrame.fV + 100.0;
            stTxFrame.fW          := stRxFrame.fW + 1000.0;
            FOR i:= 0 TO 9 DO
                stTxFrame.aFloats[i] := stTxFrame.aFloats[i] + i + 3.141592;
            END_FOR
            stTxFrame.nCRC        := F_CheckSum( ADR( stTxFrame), SIZEOF( stTxFrame) - 1 );
(* Create checksum *)
        ELSE(* => Checksum error *)
            nRxErrors := nRxErrors + 1;
        END_IF
    ELSE(* => Invalid frame length *)
        nRxErrors := nRxErrors + 1;
    END_IF
    nReceivedFrame := stRxFrame.nTxFrames;
END_IF
```

PC/CX (x86, x64, ARM) application

A rising edge at bWrite starts the write process. The REAL elements are converted to BC/BX format before the write operation. The data length and checksum is determined and set. A rising edge at bRead starts the read process. After a successful read operation the data length is checked, then a simple checksum. The REAL elements are then converted to PC/CX format.

```
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 *)
    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.nTxFrames;(* Report the received frame number *)
            stTxFrame.bEnable    := NOT stTxFrame.bEnable;(* Toggle bool flag *)
            stTxFrame.nCounter      := stTxFrame.nCounter + 1;(* Increment counter value *)
            stTxFrame.sMsg          := CONCAT( 'Message from PC/
CX, counter: ', DWORD_TO_STRING( stTxFrame.nCounter ) );(* Create some string message *)
            stTxFrame.fU            := stTxFrame.fU + 1.2;(* Modify some floating point values *)
            stTxFrame.fV            := stTxFrame.fV + 3.4;
            stTxFrame.fW            := stTxFrame.fW + 5.6;
            FOR i:= 0 TO 9 DO
                stTxFrame.aFloats[i] := stTxFrame.aFloats[i] + i;
            END_FOR
            stTxFrame.nCRC          := 0;

            (* Create temporary copy of tx data *)
            stTxToBC := stTxFrame;

            (* Swap REAL variables to BC/BX (Bus Terminal Controller) format *)
            F_SwapRealEx( stTxToBC .fU );
            F_SwapRealEx( stTxToBC .fV );
            F_SwapRealEx( stTxToBC .fW );
            FOR i:= 0 TO 9 DO
                F_SwapRealEx( stTxToBC .aFloats[i] );
            END_FOR

            (* Create CRC check number *)
            stTxToBC .nCRC      := F_CheckSum( ADR( stTxToBC ), SIZEOF( stTxToBC ) - 1 );

            (* Send *)
            fbWrite( WRITE := FALSE );
            fbWrite( LEN := SIZEOF( stTxToBC ), SRCADDR := ADR( stTxToBC ), WRITE := TRUE );
            nTxState := 1;
        END_IF

    1:(* Wait until ads write command not busy *)
        fbWrite( WRITE := FALSE );
        IF NOT fbWrite.BUSY THEN
            IF NOT fbWrite.ERR THEN
                nTxState := 0;
            ELSE(* Ads error *)
                nTxState := 100;
            END_IF
        END_IF

    100: (* TODO: Error state, add error handling *)
        nTxErrors := nTxErrors + 1;
        nTxState := 0;

END_CASE

(*##########################################################################################*)
CASE nRxState OF
    0:
        IF bRead THEN(* Read BC/BX data *)
            bRead := FALSE;

            fbRead( READ := FALSE );
            fbRead( LEN := SIZEOF( stRxFromBC ), DESTADDR := ADR( stRxFromBC ), READ := TRUE );
            nRxState := 1;
        END_IF

    1:(* Wait until ads read command not busy *)
        fbRead( READ := FALSE );
        IF NOT fbRead.BUSY THEN
            IF NOT fbRead.ERR THEN
                (* Perform simple frame length check *)
                IF stRxFromBC.nFrameSize = SIZEOF( stRxFromBC ) THEN (* Check frame length *)
                    (* Perform simple CRC check *)
                    IF stRxFromBC.nCRC = F_CheckSum( ADR( stRxFromBC ), SIZEOF( stRxFromBC ) -
 1 ) THEN

                        (* Swap REAL variables to PC/CX (x86) format *)
                        F_SwapRealEx( stRxFromBC.fU );
                        F_SwapRealEx( stRxFromBC.fV );
                        F_SwapRealEx( stRxFromBC.fW );
                        FOR i:= 0 TO 9 DO
                            F_SwapRealEx( stRxFromBC.aFloats[i] );
                        END_FOR
```

```
                        stRxFrame := stRxFromBC;
                        nRxState     := 0;

                    ELSE(* => Checksum error *)
                        nRxState := 100;
                    END_IF
                ELSE(* => Invalid frame length *)
                    nRxState := 100;
                END_IF
            ELSE(* => Ads error *)
                nRxState := 100;
            END_IF
        END_IF

    100: (* TODO: Error state, add error handling *)
        nRxErrors := nRxErrors + 1;
        nRxState := 0;

END_CASE
```

Application test

Open the PC/CX application and write TRUE to the bWrite variable. In the next step write TRUE to the bRead variable.

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

8.2 Example: File search (FB_EnumFindFileEntry, FB_EnumFindFileList)

Here you can unpack the complete sources: https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_Utilities/Resources/803340811/.zip

Example: FB_EnumFindFileEntry (ST)

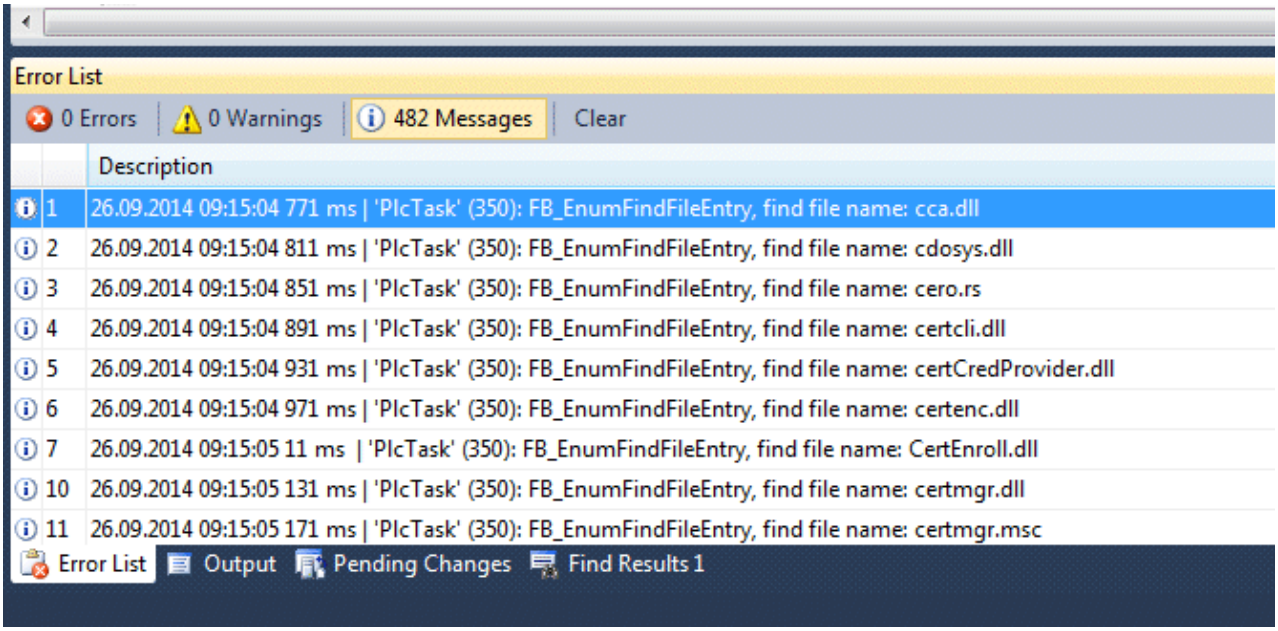
In the local TwinCAT system all files should be listed in the following directory: C:\Windows\system32\. The file names should be written as messages into the TwinCAT XAE error list. It should be possible to cancel this process. A rising edge at the *bEnum* variable starts the listing of found files. A rising edge at the *bAbort* variables aborts the process.

```
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, fin
d file name: %s', fbEnum.stFindFile.sFileName );
                    fbEnum.eCmd := eEnumCmd_Next; (* enum next entry *)
                    nState      := 1;
                ELSE (* no more entries *)
                    nState      := 0;
                END_IF
            ELSE (* log error *)
                ADSLOGSTR( ADSLOG_MSGTYPE_ERROR OR ADSLOG_MSGTYPE_LOG, 'FB_EnumFindFileEntry error:
%s', DWORD_TO_HEXSTR( fbEnum.nErrID, 0, FALSE ) );
                nState     := 0;
            END_IF
        END_IF
END_CASE
```

The log messages written into the TwinCAT XAE error list:

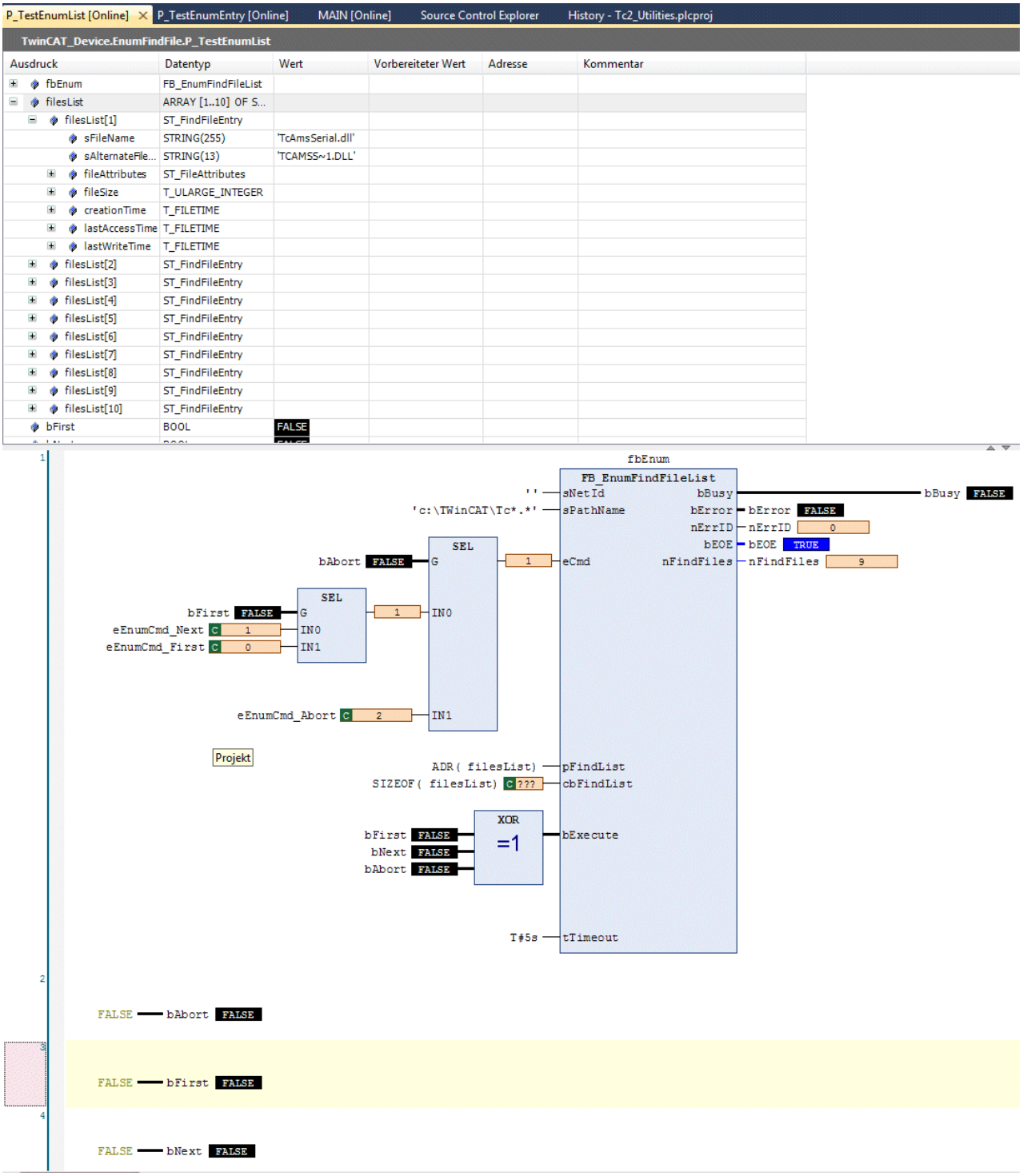


Example: FB_EnuFindFileList (FBD)

A rising edge at the *bFirst* variables activates the process. If successful the file names are entered in the *fileList* array variable.

```
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
```

Online view:



Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

8.3 Example: File ring FIFO (FB_FileRingBuffer)

The complete sources can be found here: https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_Utilities/Resources/803395851/.zip

The following example illustrates a simple application of the function block. The rising edge at *bOpen* opens an existing ring buffer file. If the file does not exist, a new one is created. The rising edge at *bClose* closes an open file. The rising edge at *bCreate* creates a new file. If you set *bAdd* = TRUE a new data record will be written into the ring buffer file, and when *bRemove* =TRUE the oldest data record is removed.

```
PROGRAM MAIN
VAR
    bOpen    : BOOL;
    bClose   : BOOL;
    bCreate  : BOOL;
    bAdd     : BOOL;
    bRemove  : BOOL;
    bGet     : BOOL;
    bReset   : BOOL;

    fbFileBuffer    : FB_FileRingBuffer := ( sNetId     := '',
                                             sPathName  := 'c:\temp\Data.dat',
                                             ePath      := PATH_GENERIC,
                                             nID        := 1,
                                             cbBuffer   := 100, (*cbBuffer := 16#80000000, 2GB*)
                                             bOverwrite := TRUE,
                                             pWriteBuff := 0,
                                             cbWriteLen := 0,
                                             pReadBuff  := 0,
                                             cbReadLen  := 0,
                                             tTimeout   := t#5s );

    storeData : ARRAY[1..10] OF BYTE :=[10(0)];
    cbStore   : UDINT;
    loadData  : ARRAY[1..10] OF BYTE :=[10(0)];
    cbLoad    : UDINT;
    i         : INT;
END_VAR
```

```
fbFileBuffer( cbReturn => cbLoad );

IF NOT fbFileBuffer.bBusy THEN

    IF bOpen THEN
        bOpen := FALSE;
        fbFileBuffer.A_Open();
    END_IF

    IF bClose THEN
        bClose := FALSE;
        fbFileBuffer.A_Close();
    END_IF

    IF bCreate THEN
        bCreate := FALSE;
        fbFileBuffer.A_Create();
    END_IF

    IF bAdd THEN
        bAdd := FALSE;

        (* modify data *)
        FOR i:=1 TO 10 BY 1 DO
            storeData[i] := storeData[i] + 1;
        END_FOR

        cbStore := SEL( cbStore > 1, SIZEOF(storeData), cbStore -
 1 ); (* modify the data chunk length *)
        fbFileBuffer.A_AddTail( pWriteBuff := ADR(storeData), cbWriteLen := cbStore,
                                pReadBuff := 0, cbReadLen:=0 );
    END_IF

    IF bRemove THEN
        bRemove := FALSE;
        fbFileBuffer.A_RemoveHead( pWriteBuff := 0, cbWriteLen := 0,
                                   pReadBuff := ADR(loadData), cbReadLen := SIZEOF(loadData));
    END_IF

    IF bGet THEN
        bGet := FALSE;
        fbFileBuffer.A_GetHead( pWriteBuff := 0, cbWriteLen := 0,
                                pReadBuff := ADR(loadData), cbReadLen := SIZEOF(loadData));
    END_IF
```

```
    IF bReset THEN
        bReset := FALSE;
        fbFileBuffer.A_Reset();
    END_IF

END_IF
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

8.4 Example: Memory ring FiFo (FB_MemRingBuffer)

The complete sources can be found here: https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_Utilities/Resources/803399691/.zip

The following example illustrates a simple application of the function block. Data sets with the same length are to be buffered (although this is not compulsory). The data sets have the following structure:

```
TYPE ST_DataSetEntry :
STRUCT
    bFlag  : BOOL;
    nValue : BYTE;
    sMsg   : STRING(20) := 'Unknown';
END_STRUCT
END_TYPE
```

The interface of the FB_Data setFifo function block:

The application-specific function block FB_Data setFifo used in the project example uses the FB_MemRingBuffer function block internally. This block simplifies adding/removing of data sets. In addition, the new function block provides the current percentage fill status of the buffer and an overwrite option. If the *bOverwrite* input is set and the buffer is full, the oldest entry is removed from the buffer and overwritten with the new one.

```
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
```

The main program:

A rising edge at *bReset* deletes all buffer entries. If you set *bAdd* = TRUE a new data record will be written into the ring buffer, and when *bRemove* =TRUE the oldest data record is removed. A rising edge at *bGet* results in the oldest data set to be read but not removed.

```
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
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

8.5 Example: Memory ring FiFo (FB_MemRingBufferEx)

Here you can unpack the complete sources: https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_Utilities/Resources/803404811/.zip

A rising edge at bAdd causes new data elements (pubObj array) to be stored in the ring buffer. Via a rising edge at bGet the oldest data element can then be copied to the getObj variable.

Data elements that are not required are removed from the buffer via a rising edge at bRelease.

```
PROGRAM MAIN
VAR
    bReset  : BOOL := TRUE;
    bAdd, bGet, bRelease, bGetFree : BOOL;
    putObj  : ARRAY[0..3] OF BYTE :=[ 16#00, 16#AA, 16#BB, 16#CC];
    getObj  : ARRAY[0..3] OF BYTE :=[ 4(0)];
    bOk     : BOOL;
    nCount  : UDINT;
    cbSize  : UDINT;
    cbFree  : UDINT;
    fbBuffer: FB_MemRingBufferEx;
    buffer  : ARRAY[0..30] OF BYTE;
END_VAR
```

```
IF bReset THEN
    bReset := FALSE;
    fbBuffer.A_Reset( pBuffer := ADR( buffer ), cbBuffer := SIZEOF( buffer ),
                    bOk=>bOk,nCount=>nCount, cbSize=>cbSize, cbFree=>cbFree );
END_IF

IF bAdd THEN
    bAdd := FALSE;
    putObj[0] := putObj[0] + 1;(* modify data *)
```

```
    fbBuffer.A_AddTail( pBuffer := ADR( buffer ), cbBuffer := SIZEOF( buffer ),
                        pWrite := ADR( putObj ), cbWrite := SIZEOF( putObj ),
                        bOk=>bOk, nCount=>nCount, cbSize=>cbSize, cbFree=>cbFree );
    IF fbBuffer.bOk THEN
        ;(* Success *)
    ELSE
        ;(* Buffer overflow *)
    END_IF
END_IF

IF bGet THEN
    bGet := FALSE;
    fbBuffer.A_GetHead( pBuffer := ADR( buffer ), cbBuffer := SIZEOF( buffer ),
                        bOk=>bOk, nCount=>nCount, cbSize=>cbSize, cbFree=>cbFree );
    IF fbBuffer.bOk THEN
        (* Success *)
        MEMCPY( ADR( getObj ), fbBuffer.pRead, MIN( SIZEOF( getObj ), fbBuffer.cbRead ) );
    ELSE
        ;(* Buffer empty *)
    END_IF

END_IF


IF bRelease THEN
    bRelease := FALSE;
    fbBuffer.A_FreeHead( pBuffer := ADR( buffer ), cbBuffer := SIZEOF( buffer ),
                        bOk=>bOk, nCount=>nCount, cbSize=>cbSize, cbFree=>cbFree );
    IF fbBuffer.bOk THEN
        ;(* Success *)
    ELSE
        ;(* Buffer empty *)
    END_IF
END_IF

IF bGetFree THEN
    bGetFree := FALSE;
    fbBuffer.A_GetFreeSize( pBuffer := ADR( buffer ), cbBuffer := SIZEOF( buffer ),
                            bOk=>bOk, nCount=>nCount, cbSize=>cbSize, cbFree=>cbFree );
END_IF
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

8.6 Example: Hash table (FB_HashTableCtrl)

Here you can unpack the complete sources: https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_Utilities/Resources/803421707/.zip

The example project has two parts to the program:

- P_TABLE_OF_UDINT is a simple example program that only processes 32-bit values in the hash table.
- P_TABLE_OF_STRUCTDATA illustrates how other data types (e.g. structured data types) can be processed in the hash table.

The maximum number of table elements cannot be changed at runtime, and is limited in the example project by MAX_DATA_ELEMENTS. If more elements are required, the table array can be enlarged accordingly (i.e. increase the value of the constant).

```
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

In the first PLC cycle the article number and article name are stored in the table. The article number serves as key and the array index of the article name as value.
Via a rising edge at bLookup the article name can be found via the article number.

```
PROGRAM P_TABLE_OF_UDINT
VAR
    sInfo    : T_MaxString := '';
    bAdd     : BOOL := TRUE;
    bLookup  : BOOL := TRUE;
    bRemove  : BOOL := TRUE;
    bEnum    : BOOL := TRUE;
    bCount   : BOOL := TRUE;

    search   : UDINT := 11111;(* article number *)

    fbTable  : FB_HashTableCtrl;(* basic hash table control function block *)
    hTable   : T_HHASHTABLE;(* hash table handle *)
    table    : ARRAY[0..MAX_DATA_ELEMENTS] OF T_HashTableEntry;
(* Max. number of hash table entries. The value of hash table entry = 32 bit integer *)
    names    : ARRAY[0..MAX_DATA_ELEMENTS] OF STRING(MAX_NAME_LENGTH);
    bInit    : BOOL := TRUE;
END_VAR
```

```
IF bInit THEN
    bInit := FALSE;
    F_CreateHashTableHnd( ADR( table ), SIZEOF( table ), hTable );(* Intialize table handle *)
END_IF

IF bAdd THEN
    bAdd := FALSE;

    (* Fill table. Article number is the key. Array index number is the value (article name) *)
    names[0] := 'Chair';
    fbTable.A_Add( key := 12345, putValue := 0(* array index*), hTable := hTable );
    IF NOT fbTable.bOk THEN
        ;(* Table overflow *)
    END_IF

    names[1] := 'Table';
    fbTable.A_Add( key := 67890, putValue := 1, hTable := hTable );
    IF NOT fbTable.bOk THEN
        ;(* Table overflow *)
    END_IF

    names[2] := 'Couch';
    fbTable.A_Add( key := 11111, putValue := 2, hTable := hTable );
    IF NOT fbTable.bOk THEN
        ;(* Table overflow *)
    END_IF

    names[3] := 'TV set';
    fbTable.A_Add( key := 22222, putValue := 3, hTable := hTable );
    IF NOT fbTable.bOk THEN
        ;(* Table overflow *)
    END_IF
END_IF


IF bLookup THEN    (* search for the article name by article number *)
    bLookup := FALSE;
    sInfo := '';
    fbTable.A_Lookup( key := search, hTable := hTable );
    IF fbTable.bOk THEN
        sInfo := names[fbTable.getValue];
    ELSE
        ;(* Entry not found *)
    END_IF
END_IF

IF bRemove THEN(* remove one entry from the table *)
    bRemove := FALSE;
    sInfo := '';
    fbTable.A_Remove( key := search, hTable := hTable );
    IF fbTable.bOk THEN
        sInfo := names[fbTable.getValue];
    ELSE
        ;(* Entry not found *)
    END_IF
```

```
END_IF


IF bEnum THEN(* enumerate table entries *)
    bEnum := FALSE;
    sInfo := '';

    fbTable.A_GetFirst( putPosPtr := 0, hTable := hTable );
    IF fbTable.bOk THEN
        sInfo := names[fbTable.getValue];

        REPEAT
            fbTable.A_GetNext( putPosPtr := fbTable.getPosPtr , hTable := hTable );
            IF fbTable.bOk THEN
                sInfo := names[fbTable.getValue];
            END_IF
        UNTIL NOT fbTable.bOk
        END_REPEAT

    END_IF
END_IF

IF bCount THEN(* count entries in the table *)
    bCount := FALSE;
    sInfo := UDINT_TO_STRING( hTable.nCount );
END_IF
```

PROGRAM P_TABLE_OF_STRUCTDATA

This section of the program illustrates how structured data sets can be manipulated in the table in place of simply 32-bit numbers. The 32-bit element value is only used as reference pointer to the actual element value. The reference pointer is able to point to instances of structured variables or other data types. The functionality is encapsulated in a function block. The function block *FB_SpecialHashTableCtrl* can be regarded as a specialized version of the *FB_HashTableCtrl* function block. The *FB_HashTableCtrl*-block is also used internally by the FB the specialized FB.

The *DATAELEMENT_TO_STRING* function is only used to permit visual output of the value of the node.

A structured variable of type *ST_DataElement* is used as an example. The highlight: You can add further member variables to the data type declaration of ST_DataElement without having to make any changes to the program or to the FB_SpecialHashTableCtrl function block.

The type declaration for ST_DataElement:

```
TYPE ST_DataElement :(* Structured application data entry *)
STRUCT
    (* Adapt this structure to match your application needs *)
    number        : UDINT := 0;
    name        : STRING(MAX_NAME_LENGTH) := '';
    price        : REAL := 0.0;
END_STRUCT
END_TYPE
```

How do the 32-bit element values become reference pointers to the instances of the ST_DataElement array?

The maximum size of the table is limited by the constant MAX_DATA_ELEMENTS. It follows that no more than MAX_DATA_ELEMENTS reference pointers can be stored in the table. Internally the *FB_SpecialHashTableCtrl* block has a ST_DataElement array variable with the same array size as the T_HashTableEntry array variable. To simplify matters, the array indices are the same for both arrays!

Each T_HashTableEntry array element can only be used once in the table. The *FB_HashTableCtrl* function block searches for a free/unused T_HashTableEntry array element. If successful, the element is added to the table. The action *A_GetIndexAtPosPtr* can be used to determine the index of the T_HashTableEntry array. In the next step, the 32-bit node value that has just been added is assigned the address of the same array element in the ST_DataElement array. In the project example through the second action call: *A_Add*.

nodes[index].value := ADR(dataPool[index])

The allocation is realized in the *FB_SpecialHashTableCtrl->A_Add* action, for example:

```
(* Adds entry to the table *)
MEMSET( ADR( getValue ), 0, SIZEOF( getValue ) );
getPosPtr := 0;

fbTable.A_Add( hTable := hTable, key := key, putValue := 16#00000000(* we will set this value later
*), getPosPtr=>getPosPtr, bOk=>bOk );
(* Add new element to the table, getPosPtr points to the new entry *)
IF fbTable.bOk THEN(* Success *)
    fbTable.A_GetIndexAtPosPtr( hTable := hTable, putPosPtr := getPosPtr, getValue =>indexOfElem, bO
k=>bOk );(* Get array index of getPosPtr entry *)
    IF fbTable.bOk THEN(* Success *)
        pRefPtr     := ADR( dataPool[indexOfElem] );(* Get pointer to the data element *)

        pRefPtr^ := putValue;(* copy application value *)

        fbTable.A_Add( hTable := hTable, key := key, putValue := pRefPtr, bOk=>bOk );
(* Assign the entry value = pointer to the data element *)
        IF fbTable.bOk THEN(* Success *)
            getValue := putValue;
        END_IF
    END_IF
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
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

8.7 Example: Linked list (FB_LinkedListCtrl)

Here you can unpack the complete sources: https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_Utilities/Resources/803425547/.zip

The example project has two parts to the program:

- P_LIST_OF_UDINT is a simple example program that simply edits 32-bit values in the linked list.
- P_LIST_OF_STRUCTDATA illustrates how other data types (e.g. structured data types) can be managed in the context of the linked list.

The maximum number of node elements cannot be changed at runtime, and is limited in the example project by MAX_DATA_ELEMENTS. If you need more nodes, then you must increase the size of the node array accordingly (i.e. increase the value of the constant).

```
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

The handle of the linked list is initialized in the first PLC cycle. This handle is then passed as the VAR_IN_OUT variable to the FB_LinkedListCtrl function block when accessing the list. The linked list is manipulated through the action calls of the function block. This allows node elements to be added, removed, searched. The desired action is executed in response to a rising edge of the associated boolean variable. When you run the program, all the operations are carried out once.

```
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=>get
Value );
    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

This section of the program illustrates how structured data sets can be manipulated in the list in place of simply 32-bit numbers. In this case, the 32-bit node value is only used as a reference pointer to the actual value of the node. The reference pointer is able to point to instances of structured variables or other data types. The functionality is encapsulated in a function block. The function block *FB_SpecialLinkedListCtrl* can be regarded as a specialized version of the FB_*LinkedListCtrl* function block. The *FB_LinkedListCtrl* block is also used internally by the specialized FB.

The *DATAELEMENT_TO_STRING* function is only used to permit visual output of the value of the node.

A structured variable of type *ST_DataElement* is used as an example. The highlight: You can add further member variables to the data type declaration of ST_DataElement without having to make any changes to the program or to the FB_SpecialLinkedListCtrl function block.

The type declaration for ST_DataElement:

```
(* Structured application data entry *)
TYPE ST_DataElement :
STRUCT
    (* Adapt this structure to match your application needs *)
    number   : UDINT := 0;
    name     : STRING(MAX_NAME_LENGTH) := '';
    price    : REAL := 0.0;
END_STRUCT
END_TYPE
```

A simple search function is implemented. You can search for nodes having a particular *name*, *number* or *price*.

How do the 32-bit node values become reference pointers to the instances of the ST_DataElement array?

The maximum size of the list is limited by the constant MAX_DATA_ELEMENTS. It follows that no more than MAX_DATA_ELEMENTS reference pointers can be stored in the list. The *FB_SpecialLinkedListCtrl* function block has an internal ST_DataElement array variable with the same size as the T_LinkedListEntry array variable. To simplify matters, the array indices are the same for both arrays!

Each T_LinkedListEntry array element can only be inserted into the list once. The *FB_LinkedListCtrl* function block therefore searches for a free/unused T_LinkedListEntry array element, and inserts it into the list if successful. The index of the T_LinkedListEntry being used can be determined through the action *A_GetIndexAtPosPtr*. In the next step, the 32-bit node value that has just been added is assigned the address of the same array element in the ST_DataElement array. In the project example through the action call: *A_SetValueAtPosPtr*.

nodes[index].value := ADR(dataPool[index])

The assignment is, for instance, carried out in the *FB_SpecialLinkedListCtrl->A_AddHeadValue* action:

```
(* Adds head to the node list *)
MEMSET( ADR( getValue ), 0, SIZEOF( getValue ) );
getPosPtr := 0;

fbList.A_AddHeadValue( hList := hList, putValue := 16#00000000(* we will set this value later *), ge
tPosPtr=>getPosPtr, bOk=>bOk );
(* Add new element to the list, getPosPtr points to the new list node *)
IF fbList.bOk THEN(* Success *)
    fbList.A_GetIndexAtPosPtr( hList := hList, putPosPtr := getPosPtr, getValue =>indexOfElem, bOk=>
bOk );(* Get array index of getPosPtr *)
    IF fbList.bOk THEN(* Success *)
        pRefPtr     := ADR( dataPool[indexOfElem] );(* Get pointer to the data element *)
        pRefPtr^     := putValue;(* set element value *)

        fbList.A_SetValueAtPosPtr( hList := hList, putPosPtr := getPosPtr, putValue := pRefPtr, bOk=
>bOk );(* Assign the node value = pointer to the data element *)
        IF fbList.bOk THEN(* Success *)
            getValue := putValue;
```

```
        END_IF
    END_IF
END_IF
```

```
PROGRAM P_LIST_OF_STRUCTDATA
VAR
    sInfo             : T_MaxString := '';
    bAddTailValue     : BOOL := TRUE;
    bAddHeadValue     : BOOL := TRUE;
    bGetTail          : BOOL := TRUE;
    bGetHead          : BOOL := TRUE;
    bFind             : BOOL := TRUE;
    bRemoveHeadValue  : BOOL := TRUE;
    bRemoveTailValue  : BOOL := TRUE;
    bCount            : BOOL := TRUE;

    search            : ST_DataElement := ( name := 'Couch', price := 99.98, number := 12345 );
(* search value ( by name, by price or by number ) *)
    eSearch           : E_SEARCH_CRITERIA := eSEARCH_BY_NAME;
(* / eSEARCH_BY_PRICE / eSEARCH_BY_NUMBER *)

    fbList            : FB_SpecialLinkedListCtrl;
(* Specialized linked list control function block *)
    putValue          : ST_DataElement;
    getValue          : ST_DataElement;
    getPosPtr         : POINTER TO T_LinkedListEntry := 0;
    bInit             : BOOL := TRUE;
END_VAR
```

```
IF bInit THEN
    bInit     := FALSE;
    fbList.A_Reset();(* reset / initialize list *)
END_IF


IF bAddTailValue THEN(* add some nodes to the list *)
    bAddTailValue := FALSE;

    putValue.number := 22222;
    putValue.name := 'TV set';
    putValue.price :=  99.98;
    fbList.A_AddTailValue( putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbList.bOk THEN
        ;(* List overflow *)
    END_IF


    putValue.number := 11111;
    putValue.name := 'Couch';
    putValue.price :=  99.98;
    fbList.A_AddTailValue( putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbList.bOk THEN
        ;(* List overflow *)
    END_IF

    putValue.number := 12345;
    putValue.name := 'Chair';
    putValue.price := 44.98;
    fbList.A_AddTailValue( putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbList.bOk THEN
        ;(* List overflow *)
    END_IF

    putValue.number := 67890;
    putValue.name := 'Table';
    putValue.price := 99.98;
    fbList.A_AddTailValue( putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbList.bOk THEN
        ;(* List overflow *)
    END_IF
END_IF

IF bAddHeadValue THEN
    bAddHeadValue := FALSE;

    putValue.number := 33333;
    putValue.name := 'Couch';
    putValue.price := 199.98;
    fbList.A_AddHeadValue( putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbList.bOk THEN
        ;(* List overflow *)
```

```
    END_IF

    putValue.number := 44444;
    putValue.name := 'Couch';
    putValue.price := 299.98;
    fbList.A_AddHeadValue( putValue := putValue, getPosPtr=>getPosPtr, getValue=>getValue );
    IF NOT fbList.bOk THEN
        ;(* List overflow *)
    END_IF
END_IF



IF bGetTail THEN(* enumerate all nodes in list (start at tail node) *)
    bGetTail := FALSE;
    sInfo := '';

    fbList.A_GetTail( getValue=>getValue, getPosPtr=>getPosPtr );
    IF fbList.bOk THEN
        sInfo := DATAELEMENT_TO_STRING( getValue );

        REPEAT
            fbList.A_GetPrev( putPosPtr := getPosPtr, getValue=>getValue, getPosPtr=>getPosPtr );
            IF fbList.bOk THEN
                sInfo := DATAELEMENT_TO_STRING( getValue );
            ELSE
                EXIT;
            END_IF
        UNTIL NOT fbList.bOk
        END_REPEAT
    END_IF
END_IF

IF bGetHead THEN(* enumerate all nodes in list (start at head node) *)
    bGetHead := FALSE;
    sInfo := '';

    fbList.A_GetHead( getValue=>getValue, getPosPtr=>getPosPtr );
    IF fbList.bOk THEN
        sInfo := DATAELEMENT_TO_STRING( getValue );

        REPEAT
            fbList.A_GetNext( putPosPtr := getPosPtr, getValue=>getValue, getPosPtr=>getPosPtr );
            IF fbList.bOk THEN
                sInfo := DATAELEMENT_TO_STRING( getValue );
            ELSE
                EXIT;
            END_IF
        UNTIL NOT fbList.bOk
        END_REPEAT
    END_IF
END_IF


IF bFind  THEN(* search for node in the list by node value (name, price, number... )*)
    bFind     := FALSE;
    getPosPtr := 0;(* start from first node element *)
    sInfo := '';

    REPEAT
        fbList.A_Find( eSearch := eSearch, putPosPtr := getPosPtr, putValue := search, getValue=>get
Value, getPosPtr=>getPosPtr );
        IF fbList.bOk THEN
            sInfo := DATAELEMENT_TO_STRING( getValue );
        ELSE
            EXIT;
        END_IF
    UNTIL NOT fbList.bOk
    END_REPEAT

END_IF


IF bRemoveTailValue THEN(* remove tail node from node list *)
    bRemoveTailValue := FALSE;
    sInfo := '';
    fbList.A_RemoveTailValue( getValue=>getValue, getPosPtr=>getPosPtr );
    IF fbList.bOk THEN
        sInfo := DATAELEMENT_TO_STRING( getValue );
    END_IF
```

```
END_IF


IF bRemoveHeadValue THEN(* remove head node from node list *)
    bRemoveHeadValue := FALSE;
    sInfo := '';
    fbList.A_RemoveHeadValue( getValue=>getValue, getPosPtr=>getPosPtr );
    IF fbList.bOk THEN
        sInfo := DATAELEMENT_TO_STRING( getValue );
    END_IF
END_IF

IF bCount THEN(* count nodes in list *)
    bCount := FALSE;
    sInfo := '';
    fbList.A_Count( );
    IF fbList.bOk THEN
        sInfo := UDINT_TO_STRING( fbList.nCount );
    END_IF
END_IF
```

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

8.8 Example: Writing/reading of CSV file

Here you can unpack the complete sources to the sample project: https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_Utilities/Resources/803601163/.zip

CSV files generated with the project example:

Data fields without binary data: https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_Utilities/Resources/803605003/.zip
Data fields contain binary data: https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_Utilities/Resources/803430923/.zip (please note that this file requires special software for correct interpretation)

CSV stands for comma-separated values. The following documentation describes how CSV files can be written and read with the aid of auxiliary PLC CSV functions. CSV files, which are basically text files, can store simply structured data sets that can be used for data exchange between two systems. This format enables storage of tables or lists of different lengths. A table row corresponds to a data set (or row) in the CSV file. A table cell corresponds to a data field in the CSV file.

General information on the supported CSV format

- Files in CSV format should have the extension **.csv**.
- The CRLF character (CR = Carriage Return, LF= Line Feed) is used to separate the individual data sets (rows) (Windows operating systems). I.e. each data set must be followed by a CRLF.
- The CSV file must end with a CRLF character.
- Binary data must be enclosed in single quotation marks. If no single quotation marks are used the data field may only contain numbers and/or letters.
- Data fields containing special characters/control characters are enclosed in double quotation marks. If the data field contains a double quotation mark a second double quotation mark is added.
- A special character is used for separating data fields (columns). The standard separator for the individual data fields used by the auxiliary functions is a semicolon. In Germany and Europe a semicolon used as a data field separator, in the USA a comma tends to be used. The separator can be configured from semicolon to comma via the global PLC variable **DEFAULT_CSV_FIELD_SEP**.
- Each data set should have the same number of data fields (columns).

Basic configuration of a CSV file with m columns and n rows (the CRLF characters are usually not visible and are indicated in the diagram with the letters **CRLF**)

```
"Field1Record1";"Field2Record1"; ... ;"Field(m)Record1"CRLF
"Field1Record2";"Field2Record2"; ... ;"Field(m)Record2"CRLF

...

"Field1Record(n)";"Field2Record(n)";    ... ;"Field(m)Record(n)"CRLF
```

Available function blocks and functions

- STRING_TO_CSVFIELD [▶ 284], ARG_TO_CSVFIELD [▶ 242]: Converts PLC data into a data field in CSV format;
- CSVFIELD_TO_STRING [▶ 249], CSVFIELD_TO_ARG [▶ 248]: Converts data field in CSV format to PLC data;
- FB_CSVMemBufferWriter [▶ 50]: Generates data sets in a byte buffer from several data fields;
- FB_CSVMemBufferReader [▶ 48]: Splits data sets in a byte buffer into individual data fields;

Write/read CSV file in text mode or binary mode

A CSV file can be read or written in text or binary mode with the aid of the PLC function blocks for file access. Depending on the selected mode there are differences with advantages and disadvantages.

In 99% of cases the CSV files can be read/written in text mode. Binary mode is only required in rare cases.

	Text mode	Binary mode
Function block for the file read access	**FB_FileGets** (Special feature: The CR character at the end of the last data set is automatically removed from the file by this function block during read access. The character has to be restored/re-inserted to ensure that the FB_CSVMemBufferReader function block can interpret such a data set)	**FB_FileRead**
Function block for file write access	**FB_FilePuts** (Special feature: During write access this function block automatically adds a CR character at the end of the last data set. However, the FB_CSVMemBufferWriter also generates the CR characters. In order to avoid duplication of the character in the CSV file it must be removed from the buffer before the write access)	**FB_FileWrite**
Programming effort	Smaller	Greater
Special characters, non-printable control characters in the data field	Not permitted	Permitted
Maximum data set length that can be written/read	Limited to 253 characters (data set + CRLF). I.e. the data set length must not exceed 253 characters.	The maximum data set length is theoretically unlimited. However, the function blocks (FB_CSVMemBufferReader and FB_CSVMemBufferWriter) limit a CSV field to the maximum size specified in the global parameter cMaxCSVFieldValueSize.
A complete data set can be written with the function block for write access	Yes	Yes

	Text mode	Binary mode
A complete data set can be read with the function block for read access	Yes. A data set in a pure text file ends with CRLF. In such a file CRLF indicates the end of a line. The FB_FileGets function block reads the data up to CRLF.	No
Binary data in a data field	Not permitted	Permitted
Auxiliary functions for conversion of PLC data into CSV format and vice versa	CSVFIELD_TO_STRING [▶ 249] STRING_TO_CSVFIELD [▶ 284]	CSVFIELD_TO_ARG [▶ 248] ARG_TO_CSVFIELD [▶ 242]
Supported PLC variable types that can be written/ read directly	T_MaxString (STRING with 255 characters), other data types must first be converted to string and then read/written as a data field in string format.	Any data types can be written/read
Sample code	P_TextModeRead() P_TextModeWrite()	P_BinaryModeRead() P_BinaryModeWrite()

Example project

The project example actually contains 4 examples: 2 for write/read access in text mode (preferred) and 2 for write/read access in binary mode (rare):

P_TextModeRead();

P_TextModeWrite();

P_BinaryModeRead();

P_BinaryModeWrite();

Basic program sequence for reading a CSV file in text mode:

Step 1: Open the CSV file in text mode (FB_FileOpen). If successful go to step 2.

Step 2: Read a row with the function block FB_FileGets. Append a CR character (see notes in the table). If successful go to step 3 go, otherwise go to step 4 (the end of the file was reached or an error has occurred).

Step 3: Parse the read row with the function block FB_CSVMemBufferReader. The individual data fields are read. Then go to step 2 and read the next row. Repeat steps 2 and 3 until the end of the file is reached or an error occurs.

Step 4: Close the CSV file (FB_FileClose).

Basic program sequence for writing a CSV file in text mode.

Step 1: Open the CSV file in text mode (FB_FileOpen). If successful go to step 2.

Step 2: Use the function block FB_CSVMemBufferWriter to generate a new data set. The individual data fields are written into a buffer. This buffer may be a larger string. Remove the CR character at the end of the data set and go to step 3.

Step 3: Write a row with the function block FB_FilePuts. Repeat steps 2 and 3 until all data sets have been written. Then go to step 4.

Step 4: Close the file (FB_FileClose).

Requirements

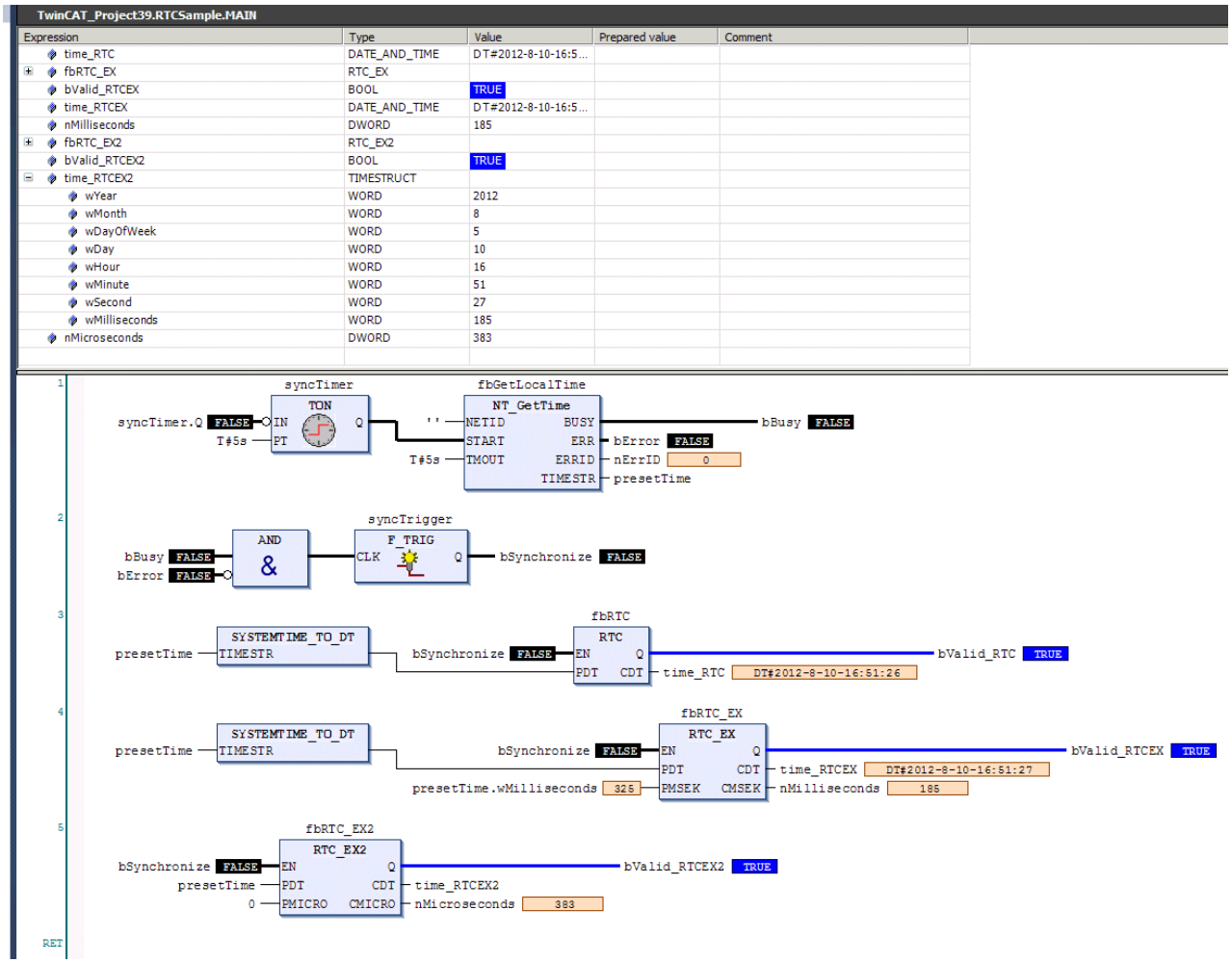
Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

8.9 Example: Software clocks (RTC, RTC_EX, RTC_EX2)

Here you can unpack the complete sources: https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc2_Utilities/Resources/803608843/.zip

In the following example the three software clocks are synchronized with the local Windows system time every 5 seconds (the local Windows system time is shown in the taskbar).

```
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
```



Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

9 Appendix

9.1 System behavior when writing persistent data

Write trigger	Internal optimization of persistent data access	Persistent data consistency	Plc cycle time exceedance
Function block WritePersistentData [▶ 142]	None	All data is from same plc cycle.	Yes, if writing of all data takes more than plc cycle time.
Function block FB_WritePersistentData [▶ 116] and SPDM_2PASS [▶ 300]	Yes	All data is from same plc cycle.	Yes, if writing of all data takes more than plc cycle time.
Function block FB_WritePersistentData [▶ 116] and SPDM_VAR_BOOST [▶ 300]	Yes	The data of each variable is from same plc cycle.	Rare, if writing of biggest pers. variable takes more than plc cycle time.
TwinCAT system stop (all persistent data is written automatically on TwinCAT system stop).	Yes	All data is from same plc cycle.	None

9.2 Format specification

This format specification is used by the FB_FormatString [▶ 64] and FB_FormatString2 [▶ 65] function blocks as well as the F_FormatArgToStr [▶ 261] function. Whereas the format specification is transferred to the function blocks via a string input variable, it is transferred to the function via the individual function parameters.

The format specification includes various required and optional parameter fields:

- Type [▶ 349]
- Flags [▶ 350]
- Width [▶ 351]
- Precision [▶ 351]

The simplest format specification contains only the percentage sign and the type field (e.g. %s). All characters that follow the percentage sign are evaluated as parameter fields up to the type field. Characters before the percentage sign and after the type field are copied into the output string. Formatting is aborted with an error in the event of unidentifiable or illegal characters. In order to output the percentage sign in the output string, use two consecutive percentage signs (%%).

Type

Required parameter field. The type field contains an ASCII character, which specifies whether the associated argument is interpreted as a string, an integer or a floating point number. Note that some type field parameters are case-sensitive.

Type	Argument	Output
b, B	BYTE, WORD, DWORD, REAL[1],SINT[2], INT[2], DINT[2], USINT, UINT, UDINT	Binary string (e.g. '101010111000')

Type	Argument	Output
o, O	BYTE, WORD, DWORD, REAL[1],SINT[2], INT[2], DINT[2], USINT, UINT, UDINT	Octal string
u, U	BYTE, WORD, DWORD, SINT[2], INT[2], DINT[2], USINT, UINT, UDINT	Decimal string without sign
c, C	BYTE, USINT	Single (ASCII) byte character
f, F	REAL[3], LREAL	Floating point number The string has the form **[–]dddd.dddd**, (dddd are decimal numbers). The number of digits before the decimal point depends on the value of the floating point number. The number of digits after the decimal point depends on the required precision. The sign only appears for negative values. **'#INF'** is returned for an infinite positive value and **'-#INF'** for an infinite negative value. If the variable transferred has an illegal value (NaN, Not-a-Number), **'#QNAN'** or **'-#QNAN'** is returned. If the length of the formatted string exceeds the maximum permissible length of the resulting string, **'#OVF'** or **'-#OVF'** is returned.
d, D	BYTE, WORD, DWORD, SINT, INT, DINT, USINT, UINT, UDINT	Decimal string The sign only appears for negative values.
s, S	STRING	Single byte character string Characters are output until the final zero or the precision field parameter has been reached.
X	BYTE, WORD, DWORD, REAL[1],SINT[2], INT[2], DINT[2], USINT, UINT, UDINT	Hexadecimal string Upper case letters ('ABCDEF') are used for formatting.
x	BYTE, WORD, DWORD, REAL[1],SINT[2], INT[2], DINT[2], USINT, UINT, UDINT	Hexadecimal string Lower case letters ('abcdef') are used for formatting.
E	Not implemented. Reserved for future use.	Floating point numbers in scientific notation
e	Not implemented. Reserved for future use.	Floating point numbers in scientific notation

[1] The content of the REAL variable is returned as a binary, octal, hexadecimal or decimal string.

[2] The content of the signed types is returned as a binary, octal, hexadecimal or decimal string.

[3] The REAL variable is converted to the LREAL type and then formatted.

Flags

Optional parameter field. One or more flags can be specified in any desired order in the flag field. The flag field parameters specify the alignment of the formatted value, the output of signs, spaces and the binary/ octal/hex prefixes.

Flag	Meaning	Type	Standard
-	Left alignment flag. The formatted value is left aligned within the width aligned in parameter Width.	Can be used in conjunction with all types.	Right alignment.

Flag	Meaning	Type	Standard
+	Sign flag. Forces output of the positive sign for signed positive numbers.	Only in conjunction with **e**, **E**, **f**, **F**, **d,D**, otherwise the flag is ignored.	The negative sign only appears for negative values.
0	Zero flag. If this flag precedes the Width parameter, the resulting string is filled with zeros from the left until the required width is reached.	Only in conjunction with **e**, **E**, **f**, **F**, **s**, **S**, otherwise the flag is ignored. The zero flag will also be ignored if the left-alignment flag (-) is additionally set.	No filling with zeros.
Space (' ')	Space flag A positive value is preceded by a blank.	Only in conjunction with **e**, **E**, **f**, **F**, **d**, **D**, otherwise the flag is ignored. The blank flag is also ignored if the left sign flag (+) was set at the same time	No blank.
#	Prefix flag. An IEC or standard C prefix is placed before the formatted value. IEC prefixes: **2#**, **8#**, **16#** (default) Standard C prefixes: **0**, **0x**, **0X**	Only in conjunction with **b**, **B**, **o**, **O**, **x**, **X**, otherwise the flag is ignored. The standard C prefix type can be activated by setting the global variable GLOBAL_FORMAT_HASH_PREFIX_TYPE in the program: GLOBAL_FORMAT_HASH_PREFIX_TYPE := HASHPREFIX_STDC;	No prefix.

Width

Optional parameter field. The Width field contains a positive decimal value that specifies the minimum number of characters output in the output string.

Depending on the alignment flag, spaces will be appended to the left or right of the output string until the desired width is reached. If the zero flag is placed before the width field parameter, the resulting string will be filled with zeros from the left until the desired width is reached. The output string will never be truncated to the desired length by the width field parameter.

An asterisk (*) can also be entered for the width field parameter. The required value is then provided by an argument (permissible types: BYTE, WORD, DWORD, USINT, UINT, UDINT). The argument for the width field parameter then follows the argument for the value to be formatted.

Precision

Optional parameter field. The precision field follows the dot (.) and contains a positive decimal value. If the dot is not followed by a value, the default precision value is used (see table).

Type	Meaning	Standard
b, B, o, O, u, U, x, X, d, D	The precision field parameter specifies how many decimal characters (digits) are output in the output string. If there are not enough digits, the string is filled with zeros from the left. The output string is never cut.	Standard: 1
c, C	Has no meaning and is ignored.	One character is output.
f, F	The precision field parameter specifies the number of decimal places. The argument value is always rounded to the respective number of decimal places.	Standard: 6 decimal places
s, S	The precision field parameter specifies how many characters from the argument string are output.	All characters up to the final zero are output.

Type	Meaning	Standard
	Characters exceeding the precision value are not output.	

An asterisk (*) can also be entered for the precision field parameter. The required value is then provided by an argument (permissible types: BYTE, WORD, DWORD, USINT, UINT, UDINT). The argument for the precision field parameter then follows the argument for the value to be formatted.

9.3 Format error codes

The following error codes are returned by the FB_FormatString [▶ 64] and FB_FormatString2 [▶ 65] function blocks as well as the F_FormatArgToStr [▶ 261] function. If several arguments are used, the argument number (1..9) is returned in addition to the error code. The argument number provides information about where exactly an error was detected during formatting.

Error code	Meaning
16#00000000	No error
16#00000010 + Argument number (1..9)	Percent sign (%) at invalid position
16#00000020 + Argument number (1..9)	Asterisk parameter at invalid position
16#00000040 + Argument number (1..9)	Invalid width field value
16#00000080 + Argument number (1..9)	Invalid precision field value
16#00000100 + Argument number (1..9)	One of the flags at invalid position
16#00000200 + Argument number (1..9)	The width or precision field value at invalid position
16#00000400 + Argument number (1..9)	Dot "." sign of precision field at invalid position
16#00000800 + Argument number (1..9)	Invalid (unsupported) type field value
16#00001000 + Argument number (1..9)	Different type field and argument parameter
16#00002000 + Argument number (1..9)	Invalid format string parameters
16#00004000 + Argument number (1..9)	Too much arguments in format string
16#00008000 + Argument number (1..9)	Destination string buffer overflow (formatted string is to long)
16#00010000 + Argument number (1..9)	Invalid pointer input

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

9.4 Scope Server error codes

The following error codes are returned by the FB_ScopeServerControl [▶ 106] function block.

```
TYPE E_UTILITIES_ERRORCODES :
(
   eUtilError_NoError                := 0,
   eUtilError_ScopeServerNotAvailable := 16#8001,
   eUtilError_ScopeServerStateChange  := 16#8002
);
END_TYPE
```

Error code	Enum	Meaning
0x0000	eUtilError_NoError	No error
0x8001	eUtilError_ScopeServerNotAvailable	TwinCAT Scope Server is not available. It may not be installed.

Error code	Enum	Meaning
0x8002	eUtilError_ScopeServerStateChange	The requested change of status is not permitted. For list of valid changes of status see B_ScopeServerControl state diagram [▶ 106].

Requirements

Development environment	Target platform	PLC libraries to be integrated (category group)
TwinCAT v3.1.0	PC or CX (x86, x64, ARM)	Tc2_Utilities (System)

9.5 ADS Return Codes

Grouping of error codes:
Global error codes: ADS Return Codes [▶ 353]... (0x9811_0000 ...)
Router error codes: ADS Return Codes [▶ 353]... (0x9811_0500 ...)
General ADS errors: ADS Return Codes [▶ 354]... (0x9811_0700 ...)
RTime error codes: ADS Return Codes [▶ 355]... (0x9811_1000 ...)

Global error codes

Hex	Dec	HRESULT	Name	Description
0x0	0	0x98110000	ERR_NOERROR	No error.
0x1	1	0x98110001	ERR_INTERNAL	Internal error.
0x2	2	0x98110002	ERR_NORTIME	No real time.
0x3	3	0x98110003	ERR_ALLOCLOCKEDMEM	Allocation locked – memory error.
0x4	4	0x98110004	ERR_INSERTMAILBOX	Mailbox full – the ADS message could not be sent. Reducing the number of ADS messages per cycle will help.
0x5	5	0x98110005	ERR_WRONGRECEIVEHMSG	Wrong HMSG.
0x6	6	0x98110006	ERR_TARGETPORTNOTFOUND	Target port not found – ADS server is not started or is not reachable.
0x7	7	0x98110007	ERR_TARGETMACHINENOTFOUND	Target computer not found – AMS route was not found.
0x8	8	0x98110008	ERR_UNKNOWNCMDID	Unknown command ID.
0x9	9	0x98110009	ERR_BADTASKID	Invalid task ID.
0xA	10	0x9811000A	ERR_NOIO	No IO.
0xB	11	0x9811000B	ERR_UNKNOWNAMSCMD	Unknown AMS command.
0xC	12	0x9811000C	ERR_WIN32ERROR	Win32 error.
0xD	13	0x9811000D	ERR_PORTNOTCONNECTED	Port not connected.
0xE	14	0x9811000E	ERR_INVALIDAMSLENGTH	Invalid AMS length.
0xF	15	0x9811000F	ERR_INVALIDAMSNETID	Invalid AMS Net ID.
0x10	16	0x98110010	ERR_LOWINSTLEVEL	Installation level is too low –TwinCAT 2 license error.
0x11	17	0x98110011	ERR_NODEBUGINTAVAILABLE	No debugging available.
0x12	18	0x98110012	ERR_PORTDISABLED	Port disabled – TwinCAT system service not started.
0x13	19	0x98110013	ERR_PORTALREADYCONNECTED	Port already connected.
0x14	20	0x98110014	ERR_AMSSYNC_W32ERROR	AMS Sync Win32 error.
0x15	21	0x98110015	ERR_AMSSYNC_TIMEOUT	AMS Sync Timeout.
0x16	22	0x98110016	ERR_AMSSYNC_AMSERROR	AMS Sync error.
0x17	23	0x98110017	ERR_AMSSYNC_NOINDEXINMAP	No index map for AMS Sync available.
0x18	24	0x98110018	ERR_INVALIDAMSPORT	Invalid AMS port.
0x19	25	0x98110019	ERR_NOMEMORY	No memory.
0x1A	26	0x9811001A	ERR_TCPSEND	TCP send error.
0x1B	27	0x9811001B	ERR_HOSTUNREACHABLE	Host unreachable.
0x1C	28	0x9811001C	ERR_INVALIDAMSFRAGMENT	Invalid AMS fragment.
0x1D	29	0x9811001D	ERR_TLSSEND	TLS send error – secure ADS connection failed.
0x1E	30	0x9811001E	ERR_ACCESSDENIED	Access denied – secure ADS access denied.

Router error codes

Hex	Dec	HRESULT	Name	Description
0x500	1280	0x98110500	ROUTERERR_NOLOCKEDMEMORY	Locked memory cannot be allocated.
0x501	1281	0x98110501	ROUTERERR_RESIZEMEMORY	The router memory size could not be changed.
0x502	1282	0x98110502	ROUTERERR_MAILBOXFULL	The mailbox has reached the maximum number of possible messages.
0x503	1283	0x98110503	ROUTERERR_DEBUGBOXFULL	The Debug mailbox has reached the maximum number of possible messages.
0x504	1284	0x98110504	ROUTERERR_UNKNOWNPORTTYPE	The port type is unknown.
0x505	1285	0x98110505	ROUTERERR_NOTINITIALIZED	The router is not initialized.
0x506	1286	0x98110506	ROUTERERR_PORTALREADYINUSE	The port number is already assigned.
0x507	1287	0x98110507	ROUTERERR_NOTREGISTERED	The port is not registered.
0x508	1288	0x98110508	ROUTERERR_NOMOREQUEUES	The maximum number of ports has been reached.
0x509	1289	0x98110509	ROUTERERR_INVALIDPORT	The port is invalid.
0x50A	1290	0x9811050A	ROUTERERR_NOTACTIVATED	The router is not active.
0x50B	1291	0x9811050B	ROUTERERR_FRAGMENTBOXFULL	The mailbox has reached the maximum number for fragmented messages.
0x50C	1292	0x9811050C	ROUTERERR_FRAGMENTTIMEOUT	A fragment timeout has occurred.
0x50D	1293	0x9811050D	ROUTERERR_TOBEREMOVED	The port is removed.

General ADS error codes

Hex	Dec	HRESULT	Name	Description
0x700	1792	0x98110700	ADSERR_DEVICE_ERROR	General device error.
0x701	1793	0x98110701	ADSERR_DEVICE_SRVNOTSUPP	Service is not supported by the server.
0x702	1794	0x98110702	ADSERR_DEVICE_INVALIDGRP	Invalid index group.
0x703	1795	0x98110703	ADSERR_DEVICE_INVALIDOFFSET	Invalid index offset.
0x704	1796	0x98110704	ADSERR_DEVICE_INVALIDACCESS	Reading or writing not permitted.
0x705	1797	0x98110705	ADSERR_DEVICE_INVALIDSIZE	Parameter size not correct.
0x706	1798	0x98110706	ADSERR_DEVICE_INVALIDDATA	Invalid data values.
0x707	1799	0x98110707	ADSERR_DEVICE_NOTREADY	Device is not ready to operate.
0x708	1800	0x98110708	ADSERR_DEVICE_BUSY	Device is busy.
0x709	1801	0x98110709	ADSERR_DEVICE_INVALIDCONTEXT	Invalid operating system context. This can result from use of ADS blocks in different tasks. It may be possible to resolve this through multitasking synchronization in the PLC.
0x70A	1802	0x9811070A	ADSERR_DEVICE_NOMEMORY	Insufficient memory.
0x70B	1803	0x9811070B	ADSERR_DEVICE_INVALIDPARM	Invalid parameter values.
0x70C	1804	0x9811070C	ADSERR_DEVICE_NOTFOUND	Not found (files, ...).
0x70D	1805	0x9811070D	ADSERR_DEVICE_SYNTAX	Syntax error in file or command.
0x70E	1806	0x9811070E	ADSERR_DEVICE_INCOMPATIBLE	Objects do not match.
0x70F	1807	0x9811070F	ADSERR_DEVICE_EXISTS	Object already exists.
0x710	1808	0x98110710	ADSERR_DEVICE_SYMBOLNOTFOUND	Symbol not found.
0x711	1809	0x98110711	ADSERR_DEVICE_SYMBOLVERSIONINVALID	Invalid symbol version. This can occur due to an online change. Create a new handle.
0x712	1810	0x98110712	ADSERR_DEVICE_INVALIDSTATE	Device (server) is in invalid state.
0x713	1811	0x98110713	ADSERR_DEVICE_TRANSMODENOTSUPP	AdsTransMode not supported.
0x714	1812	0x98110714	ADSERR_DEVICE_NOTIFYHNDINVALID	Notification handle is invalid.
0x715	1813	0x98110715	ADSERR_DEVICE_CLIENTUNKNOWN	Notification client not registered.
0x716	1814	0x98110716	ADSERR_DEVICE_NOMOREHDLS	No further handle available.
0x717	1815	0x98110717	ADSERR_DEVICE_INVALIDWATCHSIZE	Notification size too large.
0x718	1816	0x98110718	ADSERR_DEVICE_NOTINIT	Device not initialized.
0x719	1817	0x98110719	ADSERR_DEVICE_TIMEOUT	Device has a timeout.
0x71A	1818	0x9811071A	ADSERR_DEVICE_NOINTERFACE	Interface query failed.
0x71B	1819	0x9811071B	ADSERR_DEVICE_INVALIDINTERFACE	Wrong interface requested.
0x71C	1820	0x9811071C	ADSERR_DEVICE_INVALIDCLSID	Class ID is invalid.
0x71D	1821	0x9811071D	ADSERR_DEVICE_INVALIDOBJID	Object ID is invalid.
0x71E	1822	0x9811071E	ADSERR_DEVICE_PENDING	Request pending.
0x71F	1823	0x9811071F	ADSERR_DEVICE_ABORTED	Request is aborted.

Hex	Dec	HRESULT	Name	Description
0x720	1824	0x98110720	ADSERR_DEVICE_WARNING	Signal warning.
0x721	1825	0x98110721	ADSERR_DEVICE_INVALIDARRAYIDX	Invalid array index.
0x722	1826	0x98110722	ADSERR_DEVICE_SYMBOLNOTACTIVE	Symbol not active.
0x723	1827	0x98110723	ADSERR_DEVICE_ACCESSDENIED	Access denied.
0x724	1828	0x98110724	ADSERR_DEVICE_LICENSENOTFOUND	Missing license.
0x725	1829	0x98110725	ADSERR_DEVICE_LICENSEEXPIRED	License expired.
0x726	1830	0x98110726	ADSERR_DEVICE_LICENSEEXCEEDED	License exceeded.
0x727	1831	0x98110727	ADSERR_DEVICE_LICENSEINVALID	Invalid license.
0x728	1832	0x98110728	ADSERR_DEVICE_LICENSESYSTEMID	License problem: System ID is invalid.
0x729	1833	0x98110729	ADSERR_DEVICE_LICENSENOTIMELIMIT	License not limited in time.
0x72A	1834	0x9811072A	ADSERR_DEVICE_LICENSEFUTUREISSUE	Licensing problem: time in the future.
0x72B	1835	0x9811072B	ADSERR_DEVICE_LICENSETIMETOLONG	License period too long.
0x72C	1836	0x9811072C	ADSERR_DEVICE_EXCEPTION	Exception at system startup.
0x72D	1837	0x9811072D	ADSERR_DEVICE_LICENSEDUPLICATED	License file read twice.
0x72E	1838	0x9811072E	ADSERR_DEVICE_SIGNATUREINVALID	Invalid signature.
0x72F	1839	0x9811072F	ADSERR_DEVICE_CERTIFICATEINVALID	Invalid certificate.
0x730	1840	0x98110730	ADSERR_DEVICE_LICENSEOEMNOTFOUND	Public key not known from OEM.
0x731	1841	0x98110731	ADSERR_DEVICE_LICENSERESTRICTED	License not valid for this system ID.
0x732	1842	0x98110732	ADSERR_DEVICE_LICENSEDEMODENIED	Demo license prohibited.
0x733	1843	0x98110733	ADSERR_DEVICE_INVALIDFNCID	Invalid function ID.
0x734	1844	0x98110734	ADSERR_DEVICE_OUTOFRANGE	Outside the valid range.
0x735	1845	0x98110735	ADSERR_DEVICE_INVALIDALIGNMENT	Invalid alignment.
0x736	1846	0x98110736	ADSERR_DEVICE_LICENSEPLATFORM	Invalid platform level.
0x737	1847	0x98110737	ADSERR_DEVICE_FORWARD_PL	Context – forward to passive level.
0x738	1848	0x98110738	ADSERR_DEVICE_FORWARD_DL	Context – forward to dispatch level.
0x739	1849	0x98110739	ADSERR_DEVICE_FORWARD_RT	Context – forward to real time.
0x740	1856	0x98110740	ADSERR_CLIENT_ERROR	Client error.
0x741	1857	0x98110741	ADSERR_CLIENT_INVALIDPARM	Service contains an invalid parameter.
0x742	1858	0x98110742	ADSERR_CLIENT_LISTEMPTY	Polling list is empty.
0x743	1859	0x98110743	ADSERR_CLIENT_VARUSED	Var connection already in use.
0x744	1860	0x98110744	ADSERR_CLIENT_DUPLINVOKEID	The called ID is already in use.
0x745	1861	0x98110745	ADSERR_CLIENT_SYNCTIMEOUT	Timeout has occurred – the remote terminal is not responding in the specified ADS timeout. The route setting of the remote terminal may be configured incorrectly.
0x746	1862	0x98110746	ADSERR_CLIENT_W32ERROR	Error in Win32 subsystem.
0x747	1863	0x98110747	ADSERR_CLIENT_TIMEOUTINVALID	Invalid client timeout value.
0x748	1864	0x98110748	ADSERR_CLIENT_PORTNOTOPEN	Port not open.
0x749	1865	0x98110749	ADSERR_CLIENT_NOAMSADDR	No AMS address.
0x750	1872	0x98110750	ADSERR_CLIENT_SYNCINTERNAL	Internal error in Ads sync.
0x751	1873	0x98110751	ADSERR_CLIENT_ADDHASH	Hash table overflow.
0x752	1874	0x98110752	ADSERR_CLIENT_REMOVEHASH	Key not found in the table.
0x753	1875	0x98110753	ADSERR_CLIENT_NOMORESYM	No symbols in the cache.
0x754	1876	0x98110754	ADSERR_CLIENT_SYNCRESINVALID	Invalid response received.
0x755	1877	0x98110755	ADSERR_CLIENT_SYNCPORTLOCKED	Sync Port is locked.
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	The request was cancelled.

RTime error codes

Hex	Dec	HRESULT	Name	Description
0x1000	4096	0x98111000	RTERR_INTERNAL	Internal error in the real-time system.
0x1001	4097	0x98111001	RTERR_BADTIMERPERIODS	Timer value is not valid.
0x1002	4098	0x98111002	RTERR_INVALIDTASKPTR	Task pointer has the invalid value 0 (zero).
0x1003	4099	0x98111003	RTERR_INVALIDSTACKPTR	Stack pointer has the invalid value 0 (zero).
0x1004	4100	0x98111004	RTERR_PRIOEXISTS	The request task priority is already assigned.
0x1005	4101	0x98111005	RTERR_NOMORETCB	No free TCB (Task Control Block) available. The maximum number of TCBs is 64.
0x1006	4102	0x98111006	RTERR_NOMORESEMAS	No free semaphores available. The maximum number of semaphores is 64.

Hex	Dec	HRESULT	Name	Description
0x1007	4103	0x98111007	RTERR_NOMOREQUEUES	No free space available in the queue. The maximum number of positions in the queue is 64.
0x100D	4109	0x9811100D	RTERR_EXTIRQALREADYDEF	An external synchronization interrupt is already applied.
0x100E	4110	0x9811100E	RTERR_EXTIRQNOTDEF	No external sync interrupt applied.
0x100F	4111	0x9811100F	RTERR_EXTIRQINSTALLFAILED	Application of the external synchronization interrupt has failed.
0x1010	4112	0x98111010	RTERR_IRQLNOTLESSOREQUAL	Call of a service function in the wrong context
0x1017	4119	0x98111017	RTERR_VMXNOTSUPPORTED	Intel VT-x extension is not supported.
0x1018	4120	0x98111018	RTERR_VMXDISABLED	Intel VT-x extension is not enabled in the BIOS.
0x1019	4121	0x98111019	RTERR_VMXCONTROLSMISSING	Missing function in Intel VT-x extension.
0x101A	4122	0x9811101A	RTERR_VMXENABLEFAILS	Activation of Intel VT-x fails.

Specific positive HRESULT Return Codes:

HRESULT	Name	Description
0x0000_0000	S_OK	No error.
0x0000_0001	S_FALSE	No error. Example: successful processing, but with a negative or incomplete result.
0x0000_0203	S_PENDING	No error. Example: successful processing, but no result is available yet.
0x0000_0256	S_WATCHDOG_TIMEOUT	No error. Example: successful processing, but a timeout occurred.

TCP Winsock error codes

Hex	Dec	Name	Description
0x274C	10060	WSAETIMEDOUT	A connection timeout has occurred - error while establishing the connection, because the remote terminal did not respond properly after a certain period of time, or the established connection could not be maintained because the connected host did not respond.
0x274D	10061	WSAECONNREFUSED	Connection refused - no connection could be established because the target computer has explicitly rejected it. This error usually results from an attempt to connect to a service that is inactive on the external host, that is, a service for which no server application is running.
0x2751	10065	WSAEHOSTUNREACH	No route to host - a socket operation referred to an unavailable host.
More Winsock error codes: Win32 error codes [▶ 356]			

9.6 Win32 Error Codes

The following table provides a list of Win32 error codes.

0 [▶ 356], 100 [▶ 358], 200 [▶ 360], 1001 [▶ 361], 1100 [▶ 363], 1200 [▶ 365], 1400 [▶ 370], 1600 [▶ 371], 1800 [▶ 375], 2000 [▶ 376], 3000 [▶ 377], 5000 [▶ 379], 6000 [▶ 382], 8000 [▶ 384], 8500 [▶ 392], 9001 [▶ 396], 10004 [▶ 397], 12000 [▶ 400],

Error			Description
decimal	Hexadecimal	Name	
0	0x00000000	ERROR_SUCCESS	The operation completed successfully.
1	0x00000001	ERROR_INVALID_FUNCTION	Incorrect function.
2	0x00000002	ERROR_FILE_NOT_FOUND	The system cannot find the file specified.
3	0x00000003	ERROR_PATH_NOT_FOUND	The system cannot find the path specified.
4	0x00000004	ERROR_TOO_MANY_OPEN_FILES	The system cannot open the file.
5	0x00000005	ERROR_ACCESS_DENIED	Access is denied.
6	0x00000006	ERROR_INVALID_HANDLE	The handle is invalid.
7	0x00000007	ERROR_ARENA_TRASHED	The storage control blocks were destroyed.
8	0x00000008	ERROR_NOT_ENOUGH_MEMORY	Not enough storage is available to process this command.
9	0x00000009	ERROR_INVALID_BLOCK	The storage control block address is invalid.
10	0x0000000A	ERROR_BAD_ENVIRONMENT	The environment is incorrect.

Error			Description
decimal	Hexadecimal	Name	
11	0x0000000B	ERROR_BAD_FORMAT	An attempt was made to load a program with an incorrect format.
12	0x0000000C	ERROR_INVALID_ACCESS	The access code is invalid.
13	0x0000000D	ERROR_INVALID_DATA	The data is invalid.
14	0x0000000E	ERROR_OUTOFMEMORY	Not enough storage is available to complete this operation.
15	0x0000000F	ERROR_INVALID_DRIVE	The system cannot find the drive specified.
16	0x00000010	ERROR_CURRENT_DIRECTORY	The directory cannot be removed.
17	0x00000011	ERROR_NOT_SAME_DEVICE	The system cannot move the file to a different disk drive.
18	0x00000012	ERROR_NO_MORE_FILES	There are no more files.
19	0x00000013	ERROR_WRITE_PROTECT	The media is write protected.
20	0x00000014	ERROR_BAD_UNIT	The system cannot find the device specified.
21	0x00000015	ERROR_NOT_READY	The device is not ready.
22	0x00000016	ERROR_BAD_COMMAND	The device does not recognize the command.
23	0x00000017	ERROR_CRC	Data error (cyclic redundancy check).
24	0x00000018	ERROR_BAD_LENGTH	The program issued a command but the command length is incorrect.
25	0x00000019	ERROR_SEEK	The drive cannot locate a specific area or track on the disk.
26	0x0000001A	ERROR_NOT_DOS_DISK	The specified disk or diskette cannot be accessed.
27	0x0000001B	ERROR_SECTOR_NOT_FOUND	The drive cannot find the sector requested.
28	0x0000001C	ERROR_OUT_OF_PAPER	The printer is out of paper.
29	0x0000001D	ERROR_WRITE_FAULT	The system cannot write to the specified device.
30	0x0000001E	ERROR_READ_FAULT	The system cannot read from the specified device.
31	0x0000001F	ERROR_GEN_FAILURE	A device attached to the system is not functioning.
32	0x00000020	ERROR_SHARING_VIOLATION	The process cannot access the file because it is being used by another process.
33	0x00000021	ERROR_LOCK_VIOLATION	The process cannot access the file because another process has locked a portion of the file.
34	0x00000022	ERROR_WRONG_DISK	The wrong diskette is in the drive. Insert %2 (Volume Serial Number: %3) into drive %1.
36	0x00000024	ERROR_SHARING_BUFFER_EXCEEDED	Too many files opened for sharing.
38	0x00000026	ERROR_HANDLE_EOF	Reached the end of the file.
39	0x00000027	ERROR_HANDLE_DISK_FULL	The disk is full.
50	0x00000032	ERROR_NOT_SUPPORTED	The request is not supported.
51	0x00000033	ERROR_REM_NOT_LIST	The remote computer is not available.
52	0x00000034	ERROR_DUP_NAME	A duplicate name exists on the network.
53	0x00000035	ERROR_BAD_NETPATH	The network path was not found.
54	0x00000036	ERROR_NETWORK_BUSY	The network is busy.
55	0x00000037	ERROR_DEV_NOT_EXIST	The specified network resource or device is no longer available.
56	0x00000038	ERROR_TOO_MANY_CMDS	The network BIOS command limit has been reached.
57	0x00000039	ERROR_ADAP_HDW_ERR	A network adapter hardware error occurred.
58	0x0000003A	ERROR_BAD_NET_RESP	The specified server cannot perform the requested operation.
59	0x0000003B	ERROR_UNEXP_NET_ERR	An unexpected network error occurred.
60	0x0000003C	ERROR_BAD_REM_ADAP	The remote adapter is not compatible.
61	0x0000003D	ERROR_PRINTQ_FULL	The printer queue is full.
62	0x0000003E	ERROR_NO_SPOOL_SPACE	Space to store the file waiting to be printed is not available on the server.
63	0x0000003F	ERROR_PRINT_CANCELLED	Your file waiting to be printed was deleted.
64	0x00000040	ERROR_NETNAME_DELETED	The specified network name is no longer available.
65	0x00000041	ERROR_NETWORK_ACCESS_DENIED	Network access is denied.
66	0x00000042	ERROR_BAD_DEV_TYPE	The network resource type is not correct.
67	0x00000043	ERROR_BAD_NET_NAME	The network name cannot be found.

Error			Description
decimal	Hexadecimal	Name	
68	0x00000044	ERROR_TOO_MANY_NAMES	The name limit for the local computer network adapter card was exceeded.
69	0x00000045	ERROR_TOO_MANY_SESS	The network BIOS session limit was exceeded.
70	0x00000046	ERROR_SHARING_PAUSED	The remote server has been paused or is in the process of being started.
71	0x00000047	ERROR_REQ_NOT_ACCEP	No more connections can be made to this remote computer at this time because there are already as many connections as the computer can accept.
72	0x00000048	ERROR_REDIR_PAUSED	The specified printer or disk device has been paused.
80	0x00000050	ERROR_FILE_EXISTS	The file exists.
82	0x00000052	ERROR_CANNOT_MAKE	The directory or file cannot be created.
83	0x00000053	ERROR_FAIL_I24	Fail on INT 24.
84	0x00000054	ERROR_OUT_OF_STRUCTURES	Storage to process this request is not available.
85	0x00000055	ERROR_ALREADY_ASSIGNED	The local device name is already in use.
86	0x00000056	ERROR_INVALID_PASSWORD	The specified network password is not correct.
87	0x00000057	ERROR_INVALID_PARAMETER	The parameter is incorrect.
88	0x00000058	ERROR_NET_WRITE_FAULT	A write fault occurred on the network.
89	0x00000059	ERROR_NO_PROC_SLOTS	The system cannot start another process at this time.

Error			Description
decimal	Hexadecimal	Name	
100	0x00000064	ERROR_TOO_MANY_SEMAPHORES	Cannot create another system semaphore.
101	0x00000065	ERROR_EXCL_SEM_ALREADY_OWNED	The exclusive semaphore is owned by another process.
102	0x00000066	ERROR_SEM_IS_SET	The semaphore is set and cannot be closed.
103	0x00000067	ERROR_TOO_MANY_SEM_REQUESTS	The semaphore cannot be set again.
104	0x00000068	ERROR_INVALID_AT_INTERRUPT_TIME	Cannot request exclusive semaphores at interrupt time.
105	0x00000069	ERROR_SEM_OWNER_DIED	The previous ownership of this semaphore has ended.
106	0x0000006A	ERROR_SEM_USER_LIMIT	Insert the diskette for drive %1.
107	0x0000006B	ERROR_DISK_CHANGE	The program stopped because an alternate diskette was not inserted.
108	0x0000006C	ERROR_DRIVE_LOCKED	The disk is in use or locked by another process.
109	0x0000006D	ERROR_BROKEN_PIPE	The pipe has been ended.
110	0x0000006E	ERROR_OPEN_FAILED	The system cannot open the device or file specified.
111	0x0000006F	ERROR_BUFFER_OVERFLOW	The file name is too long.
112	0x00000070	ERROR_DISK_FULL	There is not enough space on the disk.
113	0x00000071	ERROR_NO_MORE_SEARCH_HANDLES	No more internal file identifiers available.
114	0x00000072	ERROR_INVALID_TARGET_HANDLE	The target internal file identifier is incorrect.
117	0x00000075	ERROR_INVALID_CATEGORY	The IOCTL call made by the application program is not correct.
118	0x00000076	ERROR_INVALID_VERIFY_SWITCH	The verify-on-write switch parameter value is not correct.
119	0x00000077	ERROR_BAD_DRIVER_LEVEL	The system does not support the command requested.
120	0x00000078	ERROR_CALL_NOT_IMPLEMENTED	This function is not supported on this system.
121	0x00000079	ERROR_SEM_TIMEOUT	The semaphore timeout period has expired.
122	0x0000007A	ERROR_INSUFFICIENT_BUFFER	The data area passed to a system call is too small.
123	0x0000007B	ERROR_INVALID_NAME	The filename, directory name, or volume label syntax is incorrect.
124	0x0000007C	ERROR_INVALID_LEVEL	The system call level is not correct.
125	0x0000007D	ERROR_NO_VOLUME_LABEL	The disk has no volume label.
126	0x0000007E	ERROR_MOD_NOT_FOUND	The specified module could not be found.
127	0x0000007F	ERROR_PROC_NOT_FOUND	The specified procedure could not be found.
128	0x00000080	ERROR_WAIT_NO_CHILDREN	There are no child processes to wait for.

Error			Description
decimal	Hexadecimal	Name	
129	0x00000081	ERROR_CHILD_NOT_COMPLETE	The %1 application cannot be run in Win32 mode.
130	0x00000082	ERROR_DIRECT_ACCESS_HANDLE	Attempt to use a file handle to an open disk partition for an operation other than raw disk I/O.
131	0x00000083	ERROR_NEGATIVE_SEEK	An attempt was made to move the file pointer before the beginning of the file.
132	0x00000084	ERROR_SEEK_ON_DEVICE	The file pointer cannot be set on the specified device or file.
133	0x00000085	ERROR_IS_JOIN_TARGET	A JOIN or SUBST command cannot be used for a drive that contains previously joined drives.
134	0x00000086	ERROR_IS_JOINED	An attempt was made to use a JOIN or SUBST command on a drive that has already been joined.
135	0x00000087	ERROR_IS_SUBSTED	An attempt was made to use a JOIN or SUBST command on a drive that has already been substituted.
136	0x00000088	ERROR_NOT_JOINED	The system tried to delete the JOIN of a drive that is not joined.
137	0x00000089	ERROR_NOT_SUBSTED	The system tried to delete the substitution of a drive that is not substituted.
138	0x0000008A	ERROR_JOIN_TO_JOIN	The system tried to join a drive to a directory on a joined drive.
139	0x0000008B	ERROR_SUBST_TO_SUBST	The system tried to substitute a drive to a directory on a substituted drive.
140	0x0000008C	ERROR_JOIN_TO_SUBST	The system tried to join a drive to a directory on a substituted drive.
141	0x0000008D	ERROR_SUBST_TO_JOIN	The system tried to SUBST a drive to a directory on a joined drive.
142	0x0000008E	ERROR_BUSY_DRIVE	The system cannot perform a JOIN or SUBST at this time.
143	0x0000008F	ERROR_SAME_DRIVE	The system cannot join or substitute a drive to or for a directory on the same drive.
144	0x00000090	ERROR_DIR_NOT_ROOT	The directory is not a subdirectory of the root directory.
145	0x00000091	ERROR_DIR_NOT_EMPTY	The directory is not empty.
146	0x00000092	ERROR_IS_SUBST_PATH	The path specified is being used in a substitute.
147	0x00000093	ERROR_IS_JOIN_PATH	Not enough resources are available to process this command.
148	0x00000094	ERROR_PATH_BUSY	The path specified cannot be used at this time.
149	0x00000095	ERROR_IS_SUBST_TARGET	An attempt was made to join or substitute a drive for which a directory on the drive is the target of a previous substitute.
150	0x00000096	ERROR_SYSTEM_TRACE	System trace information was not specified in your CONFIG.SYS file, or tracing is disallowed.
151	0x00000097	ERROR_INVALID_EVENT_COUNT	The number of specified semaphore events for DosMuxSemWait is not correct.
152	0x00000098	ERROR_TOO_MANY_MUXWAITERS	DosMuxSemWait did not execute; too many semaphores are already set.
153	0x00000099	ERROR_INVALID_LIST_FORMAT	The DosMuxSemWait list is not correct.
154	0x0000009A	ERROR_LABEL_TOO_LONG	The volume label you entered exceeds the label character limit of the target file system.
155	0x0000009B	ERROR_TOO_MANY_TCBS	Cannot create another thread.
156	0x0000009C	ERROR_SIGNAL_REFUSED	The recipient process has refused the signal.
157	0x0000009D	ERROR_DISCARDED	The segment is already discarded and cannot be locked.
158	0x0000009E	ERROR_NOT_LOCKED	The segment is already unlocked.
159	0x0000009F	ERROR_BAD_THREADID_ADDR	The address for the thread ID is not correct.
160	0x000000A0	ERROR_BAD_ARGUMENTS	The argument string passed to DosExecPgm is not correct.
161	0x000000A1	ERROR_BAD_PATHNAME	The specified path is invalid.
162	0x000000A2	ERROR_SIGNAL_PENDING	A signal is already pending.
164	0x000000A4	ERROR_MAX_THRDS_REACHED	No more threads can be created in the system.
167	0x000000A7	ERROR_LOCK_FAILED	Unable to lock a region of a file.
170	0x000000AA	ERROR_BUSY	The requested resource is in use.

Error			Description
decimal	Hexadecimal	Name	
173	0x000000AD	ERROR_CANCEL_VIOLATION	A lock request was not outstanding for the supplied cancel region.
174	0x000000AE	ERROR_ATOMIC_LOCKS_NOT_SUPPORTED	The file system does not support atomic changes to the lock type.
180	0x000000B4	ERROR_INVALID_SEGMENT_NUMBER	The system detected a segment number that was not correct.
182	0x000000B6	ERROR_INVALID_ORDINAL	The operating system cannot run %1.
183	0x000000B7	ERROR_ALREADY_EXISTS	Cannot create a file when that file already exists.
186	0x000000BA	ERROR_INVALID_FLAG_NUMBER	The flag passed is not correct.
187	0x000000BB	ERROR_SEM_NOT_FOUND	The specified system semaphore name was not found.
188	0x000000BC	ERROR_INVALID_STARTING_CODESEG	The operating system cannot run %1.
189	0x000000BD	ERROR_INVALID_STACKSEG	The operating system cannot run %1.
190	0x000000BE	ERROR_INVALID_MODULETYPE	The operating system cannot run %1.
191	0x000000BF	ERROR_INVALID_EXE_SIGNATURE	Cannot run %1 in Win32 mode.
192	0x000000C0	ERROR_EXE_MARKED_INVALID	The operating system cannot run %1.
193	0x000000C1	ERROR_BAD_EXE_FORMAT	%1 is not a valid Win32 application.
194	0x000000C2	ERROR_ITERATED_DATA_EXCEEDS_64k	The operating system cannot run %1.
195	0x000000C3	ERROR_INVALID_MINALLOCSIZE	The operating system cannot run %1.
196	0x000000C4	ERROR_DYNLINK_FROM_INVALID_RING	The operating system cannot run this application program.
197	0x000000C5	ERROR_IOPL_NOT_ENABLED	The operating system is not presently configured to run this application.
198	0x000000C6	ERROR_INVALID_SEGDPL	The operating system cannot run %1.
199	0x000000C7	ERROR_AUTODATASEG_EXCEEDS_64k	The operating system cannot run this application program.

Error			Description
decimal	Hexadecimal	Name	
200	0x000000C8	ERROR_RING2SEG_MUST_BE_MOVABLE	The code segment cannot be greater than or equal to 64K.
201	0x000000C9	ERROR_RELOC_CHAIN_XEEDS_SEGLIM	The operating system cannot run %1.
202	0x000000CA	ERROR_INFLOOP_IN_RELOC_CHAIN	The operating system cannot run %1.
203	0x000000CB	ERROR_ENVVAR_NOT_FOUND	The system could not find the environment option that was entered.
205	0x000000CD	ERROR_NO_SIGNAL_SENT	No process in the command subtree has a signal handler.
206	0x000000CE	ERROR_FILENAME_EXCED_RANGE	The filename or extension is too long.
207	0x000000CF	ERROR_RING2_STACK_IN_USE	The ring 2 stack is in use.
208	0x000000D0	ERROR_META_EXPANSION_TOO_LONG	The global filename characters, * or ?, are entered incorrectly or too many global filename characters are specified.
209	0x000000D1	ERROR_INVALID_SIGNAL_NUMBER	The signal being posted is not correct.
210	0x000000D2	ERROR_THREAD_1_INACTIVE	The signal handler cannot be set.
212	0x000000D4	ERROR_LOCKED	The segment is locked and cannot be reallocated.
214	0x000000D6	ERROR_TOO_MANY_MODULES	Too many dynamic-link modules are attached to this program or dynamic-link module.
215	0x000000D7	ERROR_NESTING_NOT_ALLOWED	Cannot nest calls to LoadModule.
216	0x000000D8	ERROR_EXE_MACHINE_TYPE_MISMATCH	The image file %1 is valid, but is for a machine type other than the current machine.
230	0x000000E6	ERROR_BAD_PIPE	The pipe state is invalid.
231	0x000000E7	ERROR_PIPE_BUSY	All pipe instances are busy.
232	0x000000E8	ERROR_NO_DATA	The pipe is being closed.
233	0x000000E9	ERROR_PIPE_NOT_CONNECTED	No process is on the other end of the pipe.
234	0x000000EA	ERROR_MORE_DATA	More data is available.
240	0x000000F0	ERROR_VC_DISCONNECTED	The session was canceled.
254	0x000000FE	ERROR_INVALID_EA_NAME	The specified extended attribute name was invalid.
255	0x000000FF	ERROR_EA_LIST_INCONSISTENT	The extended attributes are inconsistent.
258	0x00000102	WAIT_TIMEOUT	The wait operation timed out.
259	0x00000103	ERROR_NO_MORE_ITEMS	No more data is available.

Error			Description
decimal	Hexadecimal	Name	
266	0x0000010A	ERROR_CANNOT_COPY	The copy functions cannot be used.
267	0x0000010B	ERROR_DIRECTORY	The directory name is invalid.
275	0x00000113	ERROR_EAS_DIDNT_FIT	The extended attributes did not fit in the buffer.
276	0x00000114	ERROR_EA_FILE_CORRUPT	The extended attribute file on the mounted file system is corrupt.
277	0x00000115	ERROR_EA_TABLE_FULL	The extended attribute table file is full.
278	0x00000116	ERROR_INVALID_EA_HANDLE	The specified extended attribute handle is invalid.
282	0x0000011A	ERROR_EAS_NOT_SUPPORTED	The mounted file system does not support extended attributes.
288	0x00000120	ERROR_NOT_OWNER	Attempt to release mutex not owned by caller.
298	0x0000012A	ERROR_TOO_MANY_POSTS	Too many posts were made to a semaphore.
299	0x0000012B	ERROR_PARTIAL_COPY	Only part of a ReadProcessMemory or WriteProcessMemory request was completed.
300	0x0000012C	ERROR_OPLOCK_NOT_GRANTED	The oplock request is denied.
301	0x0000012D	ERROR_INVALID_OPLOCK_PROTOCOL	An invalid oplock acknowledgment was received by the system.
302	0x0000012E	ERROR_DISK_TOO_FRAGMENTED	The volume is too fragmented to complete this operation.
303	0x0000012F	ERROR_DELETE_PENDING	The file cannot be opened because it is in the process of being deleted.
317	0x0000013D	ERROR_MR_MID_NOT_FOUND	The system cannot find message text for message number 0x%1 in the message file for %2.
487	0x000001E7	ERROR_INVALID_ADDRESS	Attempt to access invalid address.
534	0x00000216	ERROR_ARITHMETIC_OVERFLOW	Arithmetic result exceeded 32 bits.
535	0x00000217	ERROR_PIPE_CONNECTED	There is a process on other end of the pipe.
536	0x00000218	ERROR_PIPE_LISTENING	Waiting for a process to open the other end of the pipe.
994	0x000003E2	ERROR_EA_ACCESS_DENIED	Access to the extended attribute was denied.
995	0x000003E3	ERROR_OPERATION_ABORTED	The I/O operation has been aborted because of either a thread exit or an application request.
996	0x000003E4	ERROR_IO_INCOMPLETE	Overlapped I/O event is not in a signaled state.
997	0x000003E5	ERROR_IO_PENDING	Overlapped I/O operation is in progress.
998	0x000003E6	ERROR_NOACCESS	Invalid access to memory location.
999	0x000003E7	ERROR_SWAPERROR	Error performing inpage operation.

Error			Description
decimal	Hexadecimal	Name	
1001	0x000003E9	ERROR_STACK_OVERFLOW	Recursion too deep; the stack overflowed.
1002	0x000003EA	ERROR_INVALID_MESSAGE	The window cannot act on the sent message.
1003	0x000003EB	ERROR_CAN_NOT_COMPLETE	Cannot complete this function.
1004	0x000003EC	ERROR_INVALID_FLAGS	Invalid flags.
1005	0x000003ED	ERROR_UNRECOGNIZED_VOLUME	The volume does not contain a recognized file system. Please make sure that all required file system drivers are loaded and that the volume is not corrupted.
1006	0x000003EE	ERROR_FILE_INVALID	The volume for a file has been externally altered so that the opened file is no longer valid.
1007	0x000003EF	ERROR_FULLSCREEN_MODE	The requested operation cannot be performed in full-screen mode.
1008	0x000003F0	ERROR_NO_TOKEN	An attempt was made to reference a token that does not exist.
1009	0x000003F1	ERROR_BADDB	The configuration registry database is corrupt.
1010	0x000003F2	ERROR_BADKEY	The configuration registry key is invalid.
1011	0x000003F3	ERROR_CANTOPEN	The configuration registry key could not be opened.
1012	0x000003F4	ERROR_CANTREAD	The configuration registry key could not be read.
1013	0x000003F5	ERROR_CANTWRITE	The configuration registry key could not be written.
1014	0x000003F6	ERROR_REGISTRY_RECOVERED	One of the files in the registry database had to be recovered by use of a log or alternate copy. The recovery was successful.

Error			Description
decimal	Hexadecimal	Name	
1015	0x000003F7	ERROR_REGISTRY_CORRUPT	The registry is corrupted. The structure of one of the files containing registry data is corrupted, or the system's memory image of the file is corrupted, or the file could not be recovered because the alternate copy or log was absent or corrupted.
1016	0x000003F8	ERROR_REGISTRY_IO_FAILED	An I/O operation initiated by the registry failed unrecoverably. The registry could not read in, or write out, or flush, one of the files that contain the system's image of the registry.
1017	0x000003F9	ERROR_NOT_REGISTRY_FILE	The system has attempted to load or restore a file into the registry, but the specified file is not in a registry file format.
1018	0x000003FA	ERROR_KEY_DELETED	Illegal operation attempted on a registry key that has been marked for deletion.
1019	0x000003FB	ERROR_NO_LOG_SPACE	System could not allocate the required space in a registry log.
1020	0x000003FC	ERROR_KEY_HAS_CHILDREN	Cannot create a symbolic link in a registry key that already has subkeys or values.
1021	0x000003FD	ERROR_CHILD_MUST_BE_VOLATILE	Cannot create a stable subkey under a volatile parent key.
1022	0x000003FE	ERROR_NOTIFY_ENUM_DIR	A notify change request is being completed and the information is not being returned in the caller's buffer. The caller now needs to enumerate the files to find the changes.
1051	0x0000041B	ERROR_DEPENDENT_SERVICES_RUNNING	A stop control has been sent to a service that other running services are dependent on.
1052	0x0000041C	ERROR_INVALID_SERVICE_CONTROL	The requested control is not valid for this service.
1053	0x0000041D	ERROR_SERVICE_REQUEST_TIMEOUT	The service did not respond to the start or control request in a timely fashion.
1054	0x0000041E	ERROR_SERVICE_NO_THREAD	A thread could not be created for the service.
1055	0x0000041F	ERROR_SERVICE_DATABASE_LOCKED	The service database is locked.
1056	0x00000420	ERROR_SERVICE_ALREADY_RUNNING	An instance of the service is already running.
1057	0x00000421	ERROR_INVALID_SERVICE_ACCOUNT	The account name is invalid or does not exist, or the password is invalid for the account name specified.
1058	0x00000422	ERROR_SERVICE_DISABLED	The service cannot be started, either because it is disabled or because it has no enabled devices associated with it.
1059	0x00000423	ERROR_CIRCULAR_DEPENDENCY	Circular service dependency was specified.
1060	0x00000424	ERROR_SERVICE_DOES_NOT_EXIST	The specified service does not exist as an installed service.
1061	0x00000425	ERROR_SERVICE_CANNOT_ACCEPT_CTRL	The service cannot accept control messages at this time.
1062	0x00000426	ERROR_SERVICE_NOT_ACTIVE	The service has not been started.
1063	0x00000427	ERROR_FAILED_SERVICE_CONTROLLER_CONNECT	The service process could not connect to the service controller.
1064	0x00000428	ERROR_EXCEPTION_IN_SERVICE	An exception occurred in the service when handling the control request.
1065	0x00000429	ERROR_DATABASE_DOES_NOT_EXIST	The database specified does not exist.
1066	0x0000042A	ERROR_SERVICE_SPECIFIC_ERROR	The service has returned a service-specific error code.
1067	0x0000042B	ERROR_PROCESS_ABORTED	The process terminated unexpectedly.
1068	0x0000042C	ERROR_SERVICE_DEPENDENCY_FAIL	The dependency service or group failed to start.
1069	0x0000042D	ERROR_SERVICE_LOGON_FAILED	The service did not start due to a logon failure.
1070	0x0000042E	ERROR_SERVICE_START_HANG	After starting, the service hung in a start-pending state.
1071	0x0000042F	ERROR_INVALID_SERVICE_LOCK	The specified service database lock is invalid.
1072	0x00000430	ERROR_SERVICE_MARKED_FOR_DELETE	The specified service has been marked for deletion.
1073	0x00000431	ERROR_SERVICE_EXISTS	The specified service already exists.
1074	0x00000432	ERROR_ALREADY_RUNNING_LKG	The system is currently running with the last-known-good configuration.
1075	0x00000433	ERROR_SERVICE_DEPENDENCY_DELETED	The dependency service does not exist or has been marked for deletion.

Error			Description
decimal	Hexadecimal	Name	
1076	0x00000434	ERROR_BOOT_ALREADY_ACCEPTED	The current boot has already been accepted for use as the last-known-good control set.
1077	0x00000435	ERROR_SERVICE_NEVER_STARTED	No attempts to start the service have been made since the last boot.
1078	0x00000436	ERROR_DUPLICATE_SERVICE_NAME	The name is already in use as either a service name or a service display name.
1079	0x00000437	ERROR_DIFFERENT_SERVICE_ACCOUNT	The account specified for this service is different from the account specified for other services running in the same process.
1080	0x00000438	ERROR_CANNOT_DETECT_DRIVER_FAILURE	Failure actions can only be set for Win32 services, not for drivers.
1081	0x00000439	ERROR_CANNOT_DETECT_PROCESS_ABORT	This service runs in the same process as the service control manager. Therefore, the service control manager cannot take action if this service's process terminates unexpectedly.
1082	0x0000043A	ERROR_NO_RECOVERY_PROGRAM	No recovery program has been configured for this service.
1083	0x0000043B	ERROR_SERVICE_NOT_IN_EXE	The executable program that this service is configured to run in does not implement the service.
1084	0x0000043C	ERROR_NOT_SAFEBOOT_SERVICE	This service cannot be started in Safe Mode.

Error			Description
decimal	Hexadecimal	Name	
1100	0x0000044C	ERROR_END_OF_MEDIA	The physical end of the tape has been reached.
1101	0x0000044D	ERROR_FILEMARK_DETECTED	A tape access reached a filemark.
1102	0x0000044E	ERROR_BEGINNING_OF_MEDIA	The beginning of the tape or a partition was encountered.
1103	0x0000044F	ERROR_SETMARK_DETECTED	A tape access reached the end of a set of files.
1104	0x00000450	ERROR_NO_DATA_DETECTED	No more data is on the tape.
1105	0x00000451	ERROR_PARTITION_FAILURE	Tape could not be partitioned.
1106	0x00000452	ERROR_INVALID_BLOCK_LENGTH	When accessing a new tape of a multivolume partition, the current block size is incorrect.
1107	0x00000453	ERROR_DEVICE_NOT_PARTITIONED	Tape partition information could not be found when loading a tape.
1108	0x00000454	ERROR_UNABLE_TO_LOCK_MEDIA	Unable to lock the media eject mechanism.
1109	0x00000455	ERROR_UNABLE_TO_UNLOAD_MEDIA	Unable to unload the media.
1110	0x00000456	ERROR_MEDIA_CHANGED	The media in the drive may have changed.
1111	0x00000457	ERROR_BUS_RESET	The I/O bus was reset.
1112	0x00000458	ERROR_NO_MEDIA_IN_DRIVE	No media in drive.
1113	0x00000459	ERROR_NO_UNICODE_TRANSLATION	No mapping for the Unicode character exists in the target multi-byte code page.
1114	0x0000045A	ERROR_DLL_INIT_FAILED	A dynamic link library (DLL) initialization routine failed.
1115	0x0000045B	ERROR_SHUTDOWN_IN_PROGRESS	A system shutdown is in progress.
1116	0x0000045C	ERROR_NO_SHUTDOWN_IN_PROGRESS	Unable to abort the system shutdown because no shutdown was in progress.
1117	0x0000045D	ERROR_IO_DEVICE	The request could not be performed because of an I/O device error.
1118	0x0000045E	ERROR_SERIAL_NO_DEVICE	No serial device was successfully initialized. The serial driver will unload.
1119	0x0000045F	ERROR_IRQ_BUSY	Unable to open a device that was sharing an interrupt request (IRQ) with other devices. At least one other device that uses that IRQ was already opened.
1120	0x00000460	ERROR_MORE_WRITES	A serial I/O operation was completed by another write to the serial port. (The IOCTL_SERIAL_XOFF_COUNTER reached zero.)
1121	0x00000461	ERROR_COUNTER_TIMEOUT	A serial I/O operation completed because the timeout period expired. (The IOCTL_SERIAL_XOFF_COUNTER did not reach zero.)
1122	0x00000462	ERROR_FLOPPY_ID_MARK_NOT_FOUND	No ID address mark was found on the floppy disk.

Error			Description
decimal	Hexadecimal	Name	
1123	0x00000463	ERROR_FLOPPY_WRONG_CYLINDER	Mismatch between the floppy disk sector ID field and the floppy disk controller track address.
1124	0x00000464	ERROR_FLOPPY_UNKNOWN_ERROR	The floppy disk controller reported an error that is not recognized by the floppy disk driver.
1125	0x00000465	ERROR_FLOPPY_BAD_REGISTERS	The floppy disk controller returned inconsistent results in its registers.
1126	0x00000466	ERROR_DISK_RECALIBRATE_FAILED	While accessing the hard disk, a recalibrate operation failed, even after retries.
1127	0x00000467	ERROR_DISK_OPERATION_FAILED	While accessing the hard disk, a disk operation failed even after retries.
1128	0x00000468	ERROR_DISK_RESET_FAILED	While accessing the hard disk, a disk controller reset was needed, but even that failed.
1129	0x00000469	ERROR_EOM_OVERFLOW	Physical end of tape encountered.
1130	0x0000046A	ERROR_NOT_ENOUGH_SERVER_MEMORY	Not enough server storage is available to process this command.
1131	0x0000046B	ERROR_POSSIBLE_DEADLOCK	A potential deadlock condition has been detected.
1132	0x0000046C	ERROR_MAPPED_ALIGNMENT	The base address or the file offset specified does not have the proper alignment.
1140	0x00000474	ERROR_SET_POWER_STATE_VETOED	An attempt to change the system power state was vetoed by another application or driver.
1141	0x00000475	ERROR_SET_POWER_STATE_FAILED	The system BIOS failed an attempt to change the system power state.
1142	0x00000476	ERROR_TOO_MANY_LINKS	An attempt was made to create more links on a file than the file system supports.
1150	0x0000047E	ERROR_OLD_WIN_VERSION	The specified program requires a newer version of Windows.
1151	0x0000047F	ERROR_APP_WRONG_OS	The specified program is not a Windows or MS-DOS program.
1152	0x00000480	ERROR_SINGLE_INSTANCE_APP	Cannot start more than one instance of the specified program.
1153	0x00000481	ERROR_RMODE_APP	The specified program was written for an earlier version of Windows.
1154	0x00000482	ERROR_INVALID_DLL	One of the library files needed to run this application is damaged.
1155	0x00000483	ERROR_NO_ASSOCIATION	No application is associated with the specified file for this operation.
1156	0x00000484	ERROR_DDE_FAIL	An error occurred in sending the command to the application.
1157	0x00000485	ERROR_DLL_NOT_FOUND	One of the library files needed to run this application cannot be found.
1158	0x00000486	ERROR_NO_MORE_USER_HANDLES	The current process has used all of its system allowance of handles for Window Manager objects.
1159	0x00000487	ERROR_MESSAGE_SYNC_ONLY	The message can be used only with synchronous operations.
1160	0x00000488	ERROR_SOURCE_ELEMENT_EMPTY	The indicated source element has no media.
1161	0x00000489	ERROR_DESTINATION_ELEMENT_FULL	The indicated destination element already contains media.
1162	0x0000048A	ERROR_ILLEGAL_ELEMENT_ADDRESS	The indicated element does not exist.
1163	0x0000048B	ERROR_MAGAZINE_NOT_PRESENT	The indicated element is part of a magazine that is not present.
1164	0x0000048C	ERROR_DEVICE_REINITIALIZATION_NEEDED	The indicated device requires reinitialization due to hardware errors.
1165	0x0000048D	ERROR_DEVICE_REQUIRES_CLEANING	The device has indicated that cleaning is required before further operations are attempted.
1166	0x0000048E	ERROR_DEVICE_DOOR_OPEN	The device has indicated that its door is open.
1167	0x0000048F	ERROR_DEVICE_NOT_CONNECTED	The device is not connected.
1168	0x00000490	ERROR_NOT_FOUND	Element not found.
1169	0x00000491	ERROR_NO_MATCH	There was no match for the specified key in the index.
1170	0x00000492	ERROR_SET_NOT_FOUND	The property set specified does not exist on the object.
1171	0x00000493	ERROR_POINT_NOT_FOUND	The point passed to GetMouseMovePointsEx is not in the buffer.

Error			Description
decimal	Hexadecimal	Name	
1172	0x00000494	ERROR_NO_TRACKING_SERVICE	The tracking (workstation) service is not running.
1173	0x00000495	ERROR_NO_VOLUME_ID	The Volume ID could not be found.
1175	0x00000497	ERROR_UNABLE_TO_REMOVE_REPLACED	Unable to remove the file to be replaced.
1176	0x00000498	ERROR_UNABLE_TO_MOVE_REPLACEMENT	Unable to move the replacement file to the file to be replaced. The file to be replaced has retained its original name.
1177	0x00000499	ERROR_UNABLE_TO_MOVE_REPLACEMENT_2	Unable to move the replacement file to the file to be replaced. The file to be replaced has been renamed using the backup name.
1178	0x0000049A	ERROR_JOURNAL_DELETE_IN_PROGRESS	The volume change journal is being deleted.
1179	0x0000049B	ERROR_JOURNAL_NOT_ACTIVE	The volume change journal is not active.
1180	0x0000049C	ERROR_POTENTIAL_FILE_FOUND	A file was found, but it may not be the correct file.
1181	0x0000049D	ERROR_JOURNAL_ENTRY_DELETED	The journal entry has been deleted from the journal.

Error			Description
decimal	Hexadecimal	Name	
1200	0x000004B0	ERROR_BAD_DEVICE	The specified device name is invalid.
1201	0x000004B1	ERROR_CONNECTION_UNAVAIL	The device is not currently connected but it is a remembered connection.
1202	0x000004B2	ERROR_DEVICE_ALREADY_REMEMBERED	The local device name has a remembered connection to another network resource.
1203	0x000004B3	ERROR_NO_NET_OR_BAD_PATH	No network provider accepted the given network path.
1204	0x000004B4	ERROR_BAD_PROVIDER	The specified network provider name is invalid.
1205	0x000004B5	ERROR_CANNOT_OPEN_PROFILE	Unable to open the network connection profile.
1206	0x000004B6	ERROR_BAD_PROFILE	The network connection profile is corrupted.
1207	0x000004B7	ERROR_NOT_CONTAINER	Cannot enumerate a noncontainer.
1208	0x000004B8	ERROR_EXTENDED_ERROR	An extended error has occurred.
1209	0x000004B9	ERROR_INVALID_GROUPNAME	The format of the specified group name is invalid.
1210	0x000004BA	ERROR_INVALID_COMPUTERNAME	The format of the specified computer name is invalid.
1211	0x000004BB	ERROR_INVALID_EVENTNAME	The format of the specified event name is invalid.
1212	0x000004BC	ERROR_INVALID_DOMAINNAME	The format of the specified domain name is invalid.
1213	0x000004BD	ERROR_INVALID_SERVICENAME	The format of the specified service name is invalid.
1214	0x000004BE	ERROR_INVALID_NETNAME	The format of the specified network name is invalid.
1215	0x000004BF	ERROR_INVALID_SHARENAME	The format of the specified share name is invalid.
1216	0x000004C0	ERROR_INVALID_PASSWORDNAME	The format of the specified password is invalid.
1217	0x000004C1	ERROR_INVALID_MESSAGENAME	The format of the specified message name is invalid.
1218	0x000004C2	ERROR_INVALID_MESSAGEDEST	The format of the specified message destination is invalid.
1219	0x000004C3	ERROR_SESSION_CREDENTIAL_CONFLICT	The credentials supplied conflict with an existing set of credentials.
1220	0x000004C4	ERROR_REMOTE_SESSION_LIMIT_EXCEEDED	An attempt was made to establish a session to a network server, but there are already too many sessions established to that server.
1221	0x000004C5	ERROR_DUP_DOMAINNAME	The workgroup or domain name is already in use by another computer on the network.
1222	0x000004C6	ERROR_NO_NETWORK	The network is not present or not started.
1223	0x000004C7	ERROR_CANCELLED	The operation was canceled by the user.
1224	0x000004C8	ERROR_USER_MAPPED_FILE	The requested operation cannot be performed on a file with a user-mapped section open.
1225	0x000004C9	ERROR_CONNECTION_REFUSED	The remote system refused the network connection.
1226	0x000004CA	ERROR_GRACEFUL_DISCONNECT	The network connection was gracefully closed.
1227	0x000004CB	ERROR_ADDRESS_ALREADY_ASSOCIATED	The network transport endpoint already has an address associated with it.
1228	0x000004CC	ERROR_ADDRESS_NOT_ASSOCIATED	An address has not yet been associated with the network endpoint.

Error			Description
decimal	Hexadecimal	Name	
1229	0x000004CD	ERROR_CONNECTION_INVALID	An operation was attempted on a nonexistent network connection.
1230	0x000004CE	ERROR_CONNECTION_ACTIVE	An invalid operation was attempted on an active network connection.
1231	0x000004CF	ERROR_NETWORK_UNREACHABLE	The network location cannot be reached. For information about network troubleshooting, see Windows Help.
1232	0x000004D0	ERROR_HOST_UNREACHABLE	The network location cannot be reached. For information about network troubleshooting, see Windows Help.
1233	0x000004D1	ERROR_PROTOCOL_UNREACHABLE	The network location cannot be reached. For information about network troubleshooting, see Windows Help.
1234	0x000004D2	ERROR_PORT_UNREACHABLE	No service is operating at the destination network endpoint on the remote system.
1235	0x000004D3	ERROR_REQUEST_ABORTED	The request was aborted.
1236	0x000004D4	ERROR_CONNECTION_ABORTED	The network connection was aborted by the local system.
1237	0x000004D5	ERROR_RETRY	The operation could not be completed. A retry should be performed.
1238	0x000004D6	ERROR_CONNECTION_COUNT_LIMIT	A connection to the server could not be made because the limit on the number of concurrent connections for this account has been reached.
1239	0x000004D7	ERROR_LOGIN_TIME_RESTRICTION	Attempting to log in during an unauthorized time of day for this account.
1240	0x000004D8	ERROR_LOGIN_WKSTA_RESTRICTION	The account is not authorized to log in from this station.
1241	0x000004D9	ERROR_INCORRECT_ADDRESS	The network address could not be used for the operation requested.
1242	0x000004DA	ERROR_ALREADY_REGISTERED	The service is already registered.
1243	0x000004DB	ERROR_SERVICE_NOT_FOUND	The specified service does not exist.
1244	0x000004DC	ERROR_NOT_AUTHENTICATED	The operation being requested was not performed because the user has not been authenticated.
1245	0x000004DD	ERROR_NOT_LOGGED_ON	The operation being requested was not performed because the user has not logged on to the network. The specified service does not exist.
1246	0x000004DE	ERROR_CONTINUE	Continue with work in progress.
1247	0x000004DF	ERROR_ALREADY_INITIALIZED	An attempt was made to perform an initialization operation when initialization has already been completed.
1248	0x000004E0	ERROR_NO_MORE_DEVICES	No more local devices.
1249	0x000004E1	ERROR_NO_SUCH_SITE	The specified site does not exist.
1250	0x000004E2	ERROR_DOMAIN_CONTROLLER_EXISTS	A domain controller with the specified name already exists.
1251	0x000004E3	ERROR_ONLY_IF_CONNECTED	This operation is supported only when you are connected to the server.
1252	0x000004E4	ERROR_OVERRIDE_NOCHANGES	The group policy framework should call the extension even if there are no changes.
1253	0x000004E5	ERROR_BAD_USER_PROFILE	The specified user does not have a valid profile.
1254	0x000004E6	ERROR_NOT_SUPPORTED_ON_SBS	This operation is not supported on a Microsoft Small Business Server.
1255	0x000004E7	ERROR_SERVER_SHUTDOWN_IN_PROGRESS	The server machine is shutting down.
1256	0x000004E8	ERROR_HOST_DOWN	The remote system is not available. For information about network troubleshooting, see Windows Help.
1257	0x000004E9	ERROR_NON_ACCOUNT_SID	The security identifier provided is not from an account domain.
1258	0x000004EA	ERROR_NON_DOMAIN_SID	The security identifier provided does not have a domain component.
1259	0x000004EB	ERROR_APPHELP_BLOCK	AppHelp dialog canceled thus preventing the application from starting.
1260	0x000004EC	ERROR_ACCESS_DISABLED_BY_POLICY	Access to the requested resource has been disabled by your administrator.

Error			Description
decimal	Hexadecimal	Name	
1261	0x000004ED	ERROR_REG_NAT_CONSUMPTION	A program attempt to use an invalid register value. Normally caused by an uninitialized register. This error is Itanium specific.
1262	0x000004EE	ERROR_CSCSHARE_OFFLINE	The share is currently offline or does not exist.
1300	0x00000514	ERROR_NOT_ALL_ASSIGNED	Not all privileges referenced are assigned to the caller.
1301	0x00000515	ERROR_SOME_NOT_MAPPED	Some mapping between account names and security IDs was not done.
1302	0x00000516	ERROR_NO_QUOTAS_FOR_ACCOUNT	No system quota limits are specifically set for this account.
1303	0x00000517	ERROR_LOCAL_USER_SESSION_KEY	No encryption key is available. A well-known encryption key was returned.
1304	0x00000518	ERROR_NULL_LM_PASSWORD	The password is too complex to be converted to a LAN Manager password. The LAN Manager password returned is a NULL string.
1305	0x00000519	ERROR_UNKNOWN_REVISION	The revision level is unknown.
1306	0x0000051A	ERROR_REVISION_MISMATCH	Indicates two revision levels are incompatible.
1307	0x0000051B	ERROR_INVALID_OWNER	This security ID may not be assigned as the owner of this object.
1308	0x0000051C	ERROR_INVALID_PRIMARY_GROUP	This security ID may not be assigned as the primary group of an object.
1309	0x0000051D	ERROR_NO_IMPERSONATION_TOKEN	An attempt has been made to operate on an impersonation token by a thread that is not currently impersonating a client.
1310	0x0000051E	ERROR_CANT_DISABLE_MANDATORY	The group may not be disabled.
1311	0x0000051F	ERROR_NO_LOGON_SERVERS	There are currently no logon servers available to service the logon request.
1312	0x00000520	ERROR_NO_SUCH_LOGON_SESSION	A specified logon session does not exist. It may already have been terminated.
1313	0x00000521	ERROR_NO_SUCH_PRIVILEGE	A specified privilege does not exist.
1314	0x00000522	ERROR_PRIVILEGE_NOT_HELD	A required privilege is not held by the client.
1315	0x00000523	ERROR_INVALID_ACCOUNT_NAME	The name provided is not a properly formed account name.
1316	0x00000524	ERROR_USER_EXISTS	The specified user already exists.
1317	0x00000525	ERROR_NO_SUCH_USER	The specified user does not exist.
1318	0x00000526	ERROR_GROUP_EXISTS	The specified group already exists.
1319	0x00000527	ERROR_NO_SUCH_GROUP	The specified group does not exist.
1320	0x00000528	ERROR_MEMBER_IN_GROUP	Either the specified user account is already a member of the specified group, or the specified group cannot be deleted because it contains a member.
1321	0x00000529	ERROR_MEMBER_NOT_IN_GROUP	The specified user account is not a member of the specified group account.
1322	0x0000052A	ERROR_LAST_ADMIN	The last remaining administration account cannot be disabled or deleted.
1323	0x0000052B	ERROR_WRONG_PASSWORD	Unable to update the password. The value provided as the current password is incorrect.
1324	0x0000052C	ERROR_ILL_FORMED_PASSWORD	Unable to update the password. The value provided for the new password contains values that are not allowed in passwords.
1325	0x0000052D	ERROR_PASSWORD_RESTRICTION	Unable to update the password. The value provided for the new password does not meet the length, complexity, or history requirement of the domain.
1326	0x0000052E	ERROR_LOGON_FAILURE	Logon failure: unknown user name or bad password.
1327	0x0000052F	ERROR_ACCOUNT_RESTRICTION	Logon failure: user account restriction.
1328	0x00000530	ERROR_INVALID_LOGON_HOURS	Logon failure: account logon time restriction violation.
1329	0x00000531	ERROR_INVALID_WORKSTATION	Logon failure: user not allowed to log on to this computer.
1330	0x00000532	ERROR_PASSWORD_EXPIRED	Logon failure: the specified account password has expired.
1331	0x00000533	ERROR_ACCOUNT_DISABLED	Logon failure: account currently disabled.

Error			Description
decimal	Hexadecimal	Name	
1332	0x00000534	ERROR_NONE_MAPPED	No mapping between account names and security IDs was done.
1333	0x00000535	ERROR_TOO_MANY_LUIDS_REQUESTED	Too many local user identifiers (LUIDs) were requested at one time.
1334	0x00000536	ERROR_LUIDS_EXHAUSTED	No more local user identifiers (LUIDs) are available.
1335	0x00000537	ERROR_INVALID_SUB_AUTHORITY	The subauthority part of a security ID is invalid for this particular use.
1336	0x00000538	ERROR_INVALID_ACL	The access control list (ACL) structure is invalid.
1337	0x00000539	ERROR_INVALID_SID	The security ID structure is invalid.
1338	0x0000053A	ERROR_INVALID_SECURITY_DESCR	The security descriptor structure is invalid.
1340	0x0000053C	ERROR_BAD_INHERITANCE_ACL	The inherited access control list (ACL) or access control entry (ACE) could not be built.
1341	0x0000053D	ERROR_SERVER_DISABLED	The server is currently disabled.
1342	0x0000053E	ERROR_SERVER_NOT_DISABLED	The server is currently enabled.
1343	0x0000053F	ERROR_INVALID_ID_AUTHORITY	The value provided was an invalid value for an identifier authority.
1344	0x00000540	ERROR_ALLOTTED_SPACE_EXCEEDED	No more memory is available for security information updates.
1345	0x00000541	ERROR_INVALID_GROUP_ATTRIBUTES	The specified attributes are invalid, or incompatible with the attributes for the group as a whole.
1346	0x00000542	ERROR_BAD_IMPERSONATION_LEVEL	Either a required impersonation level was not provided, or the provided impersonation level is invalid.
1347	0x00000543	ERROR_CANT_OPEN_ANONYMOUS	Cannot open an anonymous level security token.
1348	0x00000544	ERROR_BAD_VALIDATION_CLASS	The validation information class requested was invalid.
1349	0x00000545	ERROR_BAD_TOKEN_TYPE	The type of the token is inappropriate for its attempted use.
1350	0x00000546	ERROR_NO_SECURITY_ON_OBJECT	Unable to perform a security operation on an object that has no associated security.
1351	0x00000547	ERROR_CANT_ACCESS_DOMAIN_INFO	Configuration information could not be read from the domain controller, either because the machine is unavailable, or access has been denied.
1352	0x00000548	ERROR_INVALID_SERVER_STATE	The security account manager (SAM) or local security authority (LSA) server was in the wrong state to perform the security operation.
1353	0x00000549	ERROR_INVALID_DOMAIN_STATE	The domain was in the wrong state to perform the security operation.
1354	0x0000054A	ERROR_INVALID_DOMAIN_ROLE	This operation is only allowed for the Primary Domain Controller of the domain.
1355	0x0000054B	ERROR_NO_SUCH_DOMAIN	The specified domain either does not exist or could not be contacted.
1356	0x0000054C	ERROR_DOMAIN_EXISTS	The specified domain already exists.
1357	0x0000054D	ERROR_DOMAIN_LIMIT_EXCEEDED	An attempt was made to exceed the limit on the number of domains per server.
1358	0x0000054E	ERROR_INTERNAL_DB_CORRUPTION	Unable to complete the requested operation because of either a catastrophic media failure or a data structure corruption on the disk.
1359	0x0000054F	ERROR_INTERNAL_ERROR	An internal error occurred.
1360	0x00000550	ERROR_GENERIC_NOT_MAPPED	Generic access types were contained in an access mask which should already be mapped to nongeneric types.
1361	0x00000551	ERROR_BAD_DESCRIPTOR_FORMAT	A security descriptor is not in the right format (absolute or self-relative).
1362	0x00000552	ERROR_NOT_LOGON_PROCESS	The requested action is restricted for use by logon processes only. The calling process has not registered as a logon process.
1363	0x00000553	ERROR_LOGON_SESSION_EXISTS	Cannot start a new logon session with an ID that is already in use.
1364	0x00000554	ERROR_NO_SUCH_PACKAGE	A specified authentication package is unknown.
1365	0x00000555	ERROR_BAD_LOGON_SESSION_STATE	The logon session is not in a state that is consistent with the requested operation.
1366	0x00000556	ERROR_LOGON_SESSION_COLLISION	The logon session ID is already in use.

Error			Description
decimal	Hexadecimal	Name	
1367	0x00000557	ERROR_INVALID_LOGON_TYPE	A logon request contained an invalid logon type value.
1368	0x00000558	ERROR_CANNOT_IMPERSONATE	Unable to impersonate using a named pipe until data has been read from that pipe.
1369	0x00000559	ERROR_RXACT_INVALID_STATE	The transaction state of a registry subtree is incompatible with the requested operation.
1370	0x0000055A	ERROR_RXACT_COMMIT_FAILURE	An internal security database corruption has been encountered.
1371	0x0000055B	ERROR_SPECIAL_ACCOUNT	Cannot perform this operation on built-in accounts.
1372	0x0000055C	ERROR_SPECIAL_GROUP	Cannot perform this operation on this built-in special group.
1373	0x0000055D	ERROR_SPECIAL_USER	Cannot perform this operation on this built-in special user.
1374	0x0000055E	ERROR_MEMBERS_PRIMARY_GROUP	The user cannot be removed from a group because the group is currently the user's primary group.
1375	0x0000055F	ERROR_TOKEN_ALREADY_IN_USE	The token is already in use as a primary token.
1376	0x00000560	ERROR_NO_SUCH_ALIAS	The specified local group does not exist.
1377	0x00000561	ERROR_MEMBER_NOT_IN_ALIAS	The specified account name is not a member of the local group.
1378	0x00000562	ERROR_MEMBER_IN_ALIAS	The specified account name is already a member of the local group.
1379	0x00000563	ERROR_ALIAS_EXISTS	The specified local group already exists.
1380	0x00000564	ERROR_LOGON_NOT_GRANTED	Logon failure: the user has not been granted the requested logon type at this computer.
1381	0x00000565	ERROR_TOO_MANY_SECRETS	The maximum number of secrets that may be stored in a single system has been exceeded.
1382	0x00000566	ERROR_SECRET_TOO_LONG	The length of a secret exceeds the maximum length allowed.
1383	0x00000567	ERROR_INTERNAL_DB_ERROR	The local security authority database contains an internal inconsistency.
1384	0x00000568	ERROR_TOO_MANY_CONTEXT_IDS	During a logon attempt, the user's security context accumulated too many security IDs.
1385	0x00000569	ERROR_LOGON_TYPE_NOT_GRANTED	Logon failure: the user has not been granted the requested logon type at this computer.
1386	0x0000056A	ERROR_NT_CROSS_ENCRYPTION_REQUIRED	A cross-encrypted password is necessary to change a user password.
1387	0x0000056B	ERROR_NO_SUCH_MEMBER	A new member could not be added to or removed from the local group because the member does not exist.
1388	0x0000056C	ERROR_INVALID_MEMBER	A new member could not be added to a local group because the member has the wrong account type.
1389	0x0000056D	ERROR_TOO_MANY_SIDS	Too many security IDs have been specified.
1390	0x0000056E	ERROR_LM_CROSS_ENCRYPTION_REQUIRED	A cross-encrypted password is necessary to change this user password.
1391	0x0000056F	ERROR_NO_INHERITANCE	Indicates an ACL contains no inheritable components.
1392	0x00000570	ERROR_FILE_CORRUPT	The file or directory is corrupted and unreadable.
1393	0x00000571	ERROR_DISK_CORRUPT	The disk structure is corrupted and unreadable.
1394	0x00000572	ERROR_NO_USER_SESSION_KEY	There is no user session key for the specified logon session.
1395	0x00000573	ERROR_LICENSE_QUOTA_EXCEEDED	The service being accessed is licensed for a particular number of connections. No more connections can be made to the service at this time because there are already as many connections as the service can accept.
1396	0x00000574	ERROR_WRONG_TARGET_NAME	Logon Failure: The target account name is incorrect.
1397	0x00000575	ERROR_MUTUAL_AUTH_FAILED	Mutual Authentication failed. The server's password is out of date at the domain controller.
1398	0x00000576	ERROR_TIME_SKEW	There is a time difference between the client and server.

Error			Description
decimal	Hexadecimal	Name	
1399	0x00000577	ERROR_CURRENT_DOMAIN_NOT_ALLOWED	This operation can not be performed on the current domain.

Error			Description
decimal	Hexadecimal	Name	
1400	0x00000578	ERROR_INVALID_WINDOW_HANDLE	Invalid window handle.
1401	0x00000579	ERROR_INVALID_MENU_HANDLE	Invalid menu handle.
1402	0x0000057A	ERROR_INVALID_CURSOR_HANDLE	Invalid cursor handle.
1403	0x0000057B	ERROR_INVALID_ACCEL_HANDLE	Invalid accelerator table handle.
1404	0x0000057C	ERROR_INVALID_HOOK_HANDLE	Invalid hook handle.
1405	0x0000057D	ERROR_INVALID_DWP_HANDLE	Invalid handle to a multiple-window position structure.
1406	0x0000057E	ERROR_TLW_WITH_WSCHILD	Cannot create a top-level child window.
1407	0x0000057F	ERROR_CANNOT_FIND_WND_CLASS	Cannot find window class.
1408	0x00000580	ERROR_WINDOW_OF_OTHER_THREAD	Invalid window; it belongs to other thread.
1409	0x00000581	ERROR_HOTKEY_ALREADY_REGISTERED	Hot key is already registered.
1410	0x00000582	ERROR_CLASS_ALREADY_EXISTS	Class already exists.
1411	0x00000583	ERROR_CLASS_DOES_NOT_EXIST	Class does not exist.
1412	0x00000584	ERROR_CLASS_HAS_WINDOWS	Class still has open windows.
1413	0x00000585	ERROR_INVALID_INDEX	Invalid index.
1414	0x00000586	ERROR_INVALID_ICON_HANDLE	Invalid icon handle.
1415	0x00000587	ERROR_PRIVATE_DIALOG_INDEX	Using private DIALOG window words.
1416	0x00000588	ERROR_LISTBOX_ID_NOT_FOUND	The list box identifier was not found.
1417	0x00000589	ERROR_NO_WILDCARD_CHARACTERS	No wildcards were found.
1418	0x0000058A	ERROR_CLIPBOARD_NOT_OPEN	Thread does not have a clipboard open.
1419	0x0000058B	ERROR_HOTKEY_NOT_REGISTERED	Hot key is not registered.
1420	0x0000058C	ERROR_WINDOW_NOT_DIALOG	The window is not a valid dialog window.
1421	0x0000058D	ERROR_CONTROL_ID_NOT_FOUND	Control ID not found.
1422	0x0000058E	ERROR_INVALID_COMBOBOX_MESSAGE	Invalid message for a combo box because it does not have an edit control.
1423	0x0000058F	ERROR_WINDOW_NOT_COMBOBOX	The window is not a combo box.
1424	0x00000590	ERROR_INVALID_EDIT_HEIGHT	Height must be less than 256.
1425	0x00000591	ERROR_DC_NOT_FOUND	Invalid device context (DC) handle.
1426	0x00000592	ERROR_INVALID_HOOK_FILTER	Invalid hook procedure type.
1427	0x00000593	ERROR_INVALID_FILTER_PROC	Invalid hook procedure.
1428	0x00000594	ERROR_HOOK_NEEDS_HMOD	Cannot set nonlocal hook without a module handle.
1429	0x00000595	ERROR_GLOBAL_ONLY_HOOK	This hook procedure can only be set globally.
1430	0x00000596	ERROR_JOURNAL_HOOK_SET	The journal hook procedure is already installed.
1431	0x00000597	ERROR_HOOK_NOT_INSTALLED	The hook procedure is not installed.
1432	0x00000598	ERROR_INVALID_LB_MESSAGE	Invalid message for single-selection list box.
1433	0x00000599	ERROR_SETCOUNT_ON_BAD_LB	LB_SETCOUNT sent to non-lazy list box.
1434	0x0000059A	ERROR_LB_WITHOUT_TABSTOPS	This list box does not support tab stops.
1435	0x0000059B	ERROR_DESTROY_OBJECT_OF_OTHER_THREAD	Cannot destroy object created by another thread.
1436	0x0000059C	ERROR_CHILD_WINDOW_MENU	Child windows cannot have menus.
1437	0x0000059D	ERROR_NO_SYSTEM_MENU	The window does not have a system menu.
1438	0x0000059E	ERROR_INVALID_MSGBOX_STYLE	Invalid message box style.
1439	0x0000059F	ERROR_INVALID_SPI_VALUE	Invalid system-wide (SPI_*) parameter.
1440	0x000005A0	ERROR_SCREEN_ALREADY_LOCKED	Screen already locked.
1441	0x000005A1	ERROR_HWNDS_HAVE_DIFF_PARENT	All handles to windows in a multiple-window position structure must have the same parent.
1442	0x000005A2	ERROR_NOT_CHILD_WINDOW	The window is not a child window.
1443	0x000005A3	ERROR_INVALID_GW_COMMAND	Invalid GW_* command.
1444	0x000005A4	ERROR_INVALID_THREAD_ID	Invalid thread identifier.
1445	0x000005A5	ERROR_NON_MDICHILD_WINDOW	Cannot process a message from a window that is not a multiple document interface (MDI) window.
1446	0x000005A6	ERROR_POPUP_ALREADY_ACTIVE	Popup menu already active.
1447	0x000005A7	ERROR_NO_SCROLLBARS	The window does not have scroll bars.

Error			Description
decimal	Hexadecimal	Name	
1448	0x000005A8	ERROR_INVALID_SCROLLBAR_RANGE	Scroll bar range cannot be greater than MAXLONG.
1449	0x000005A9	ERROR_INVALID_SHOWWIN_COMMAND	Cannot show or remove the window in the way specified.
1450	0x000005AA	ERROR_NO_SYSTEM_RESOURCES	Insufficient system resources exist to complete the requested service.
1451	0x000005AB	ERROR_NONPAGED_SYSTEM_RESOURCES	Insufficient system resources exist to complete the requested service.
1452	0x000005AC	ERROR_PAGED_SYSTEM_RESOURCES	Insufficient system resources exist to complete the requested service.
1453	0x000005AD	ERROR_WORKING_SET_QUOTA	Insufficient quota to complete the requested service.
1454	0x000005AE	ERROR_PAGEFILE_QUOTA	Insufficient quota to complete the requested service.
1455	0x000005AF	ERROR_COMMITMENT_LIMIT	The paging file is too small for this operation to complete.
1456	0x000005B0	ERROR_MENU_ITEM_NOT_FOUND	A menu item was not found.
1457	0x000005B1	ERROR_INVALID_KEYBOARD_HANDLE	Invalid keyboard layout handle.
1458	0x000005B2	ERROR_HOOK_TYPE_NOT_ALLOWED	Hook type not allowed.
1459	0x000005B3	ERROR_REQUIRES_INTERACTIVE_WINDOWSTATION	This operation requires an interactive window station.
1460	0x000005B4	ERROR_TIMEOUT	This operation returned because the timeout period expired.
1461	0x000005B5	ERROR_INVALID_MONITOR_HANDLE	Invalid monitor handle.
1500	0x000005DC	ERROR_EVENTLOG_FILE_CORRUPT	The event log file is corrupted.
1501	0x000005DD	ERROR_EVENTLOG_CANT_START	No event log file could be opened, so the event logging service did not start.
1502	0x000005DE	ERROR_LOG_FILE_FULL	The event log file is full.
1503	0x000005DF	ERROR_EVENTLOG_FILE_CHANGED	The event log file has changed between read operations.

Error			Description
decimal	Hexadecimal	Name	
1601	0x00000641	ERROR_INSTALL_SERVICE_FAILURE	The Windows Installer service could not be accessed. Contact your support personnel to verify that the Windows Installer service is properly registered.
1602	0x00000642	ERROR_INSTALL_USEREXIT	User cancelled installation.
1603	0x00000643	ERROR_INSTALL_FAILURE	Fatal error during installation.
1604	0x00000644	ERROR_INSTALL_SUSPEND	Installation suspended, incomplete.
1605	0x00000645	ERROR_UNKNOWN_PRODUCT	This action is only valid for products that are currently installed.
1606	0x00000646	ERROR_UNKNOWN_FEATURE	Feature ID not registered.
1607	0x00000647	ERROR_UNKNOWN_COMPONENT	Component ID not registered.
1608	0x00000648	ERROR_UNKNOWN_PROPERTY	Unknown property.
1609	0x00000649	ERROR_INVALID_HANDLE_STATE	Handle is in an invalid state.
1610	0x0000064A	ERROR_BAD_CONFIGURATION	The configuration data for this product is corrupt. Contact your support personnel.
1611	0x0000064B	ERROR_INDEX_ABSENT	Component qualifier not present.
1612	0x0000064C	ERROR_INSTALL_SOURCE_ABSENT	The installation source for this product is not available. Verify that the source exists and that you can access it.
1613	0x0000064D	ERROR_INSTALL_PACKAGE_VERSION	This installation package cannot be installed by the Windows Installer service. You must install a Windows service pack that contains a newer version of the Windows Installer service.
1614	0x0000064E	ERROR_PRODUCT_UNINSTALLED	Product is uninstalled.
1615	0x0000064F	ERROR_BAD_QUERY_SYNTAX	SQL query syntax invalid or unsupported.
1616	0x00000650	ERROR_INVALID_FIELD	Record field does not exist.
1617	0x00000651	ERROR_DEVICE_REMOVED	The device has been removed.
1618	0x00000652	ERROR_INSTALL_ALREADY_RUNNING	Another installation is already in progress. Complete that installation before proceeding with this install.

Error			Description
decimal	Hexadecimal	Name	
1619	0x00000653	ERROR_INSTALL_PACKAGE_OPEN_FAILED	This installation package could not be opened. Verify that the package exists and that you can access it, or contact the application vendor to verify that this is a valid Windows Installer package.
1620	0x00000654	ERROR_INSTALL_PACKAGE_INVALID	This installation package could not be opened. Contact the application vendor to verify that this is a valid Windows Installer package.
1621	0x00000655	ERROR_INSTALL_UI_FAILURE	There was an error starting the Windows Installer service user interface. Contact your support personnel.
1622	0x00000656	ERROR_INSTALL_LOG_FAILURE	Error opening installation log file. Verify that the specified log file location exists and that you can write to it.
1623	0x00000657	ERROR_INSTALL_LANGUAGE_UNSUPPORTED	The language of this installation package is not supported by your system.
1624	0x00000658	ERROR_INSTALL_TRANSFORM_FAILURE	Error applying transforms. Verify that the specified transform paths are valid.
1625	0x00000659	ERROR_INSTALL_PACKAGE_REJECTED	This installation is forbidden by system policy. Contact your system administrator.
1626	0x0000065A	ERROR_FUNCTION_NOT_CALLED	Function could not be executed.
1627	0x0000065B	ERROR_FUNCTION_FAILED	Function failed during execution.
1628	0x0000065C	ERROR_INVALID_TABLE	Invalid or unknown table specified.
1629	0x0000065D	ERROR_DATATYPE_MISMATCH	Data supplied is of wrong type.
1630	0x0000065E	ERROR_UNSUPPORTED_TYPE	Data of this type is not supported.
1631	0x0000065F	ERROR_CREATE_FAILED	The Windows Installer service failed to start. Contact your support personnel.
1632	0x00000660	ERROR_INSTALL_TEMP_UNWRITABLE	The temp folder is either full or inaccessible. Verify that the temp folder exists and that you can write to it.
1633	0x00000661	ERROR_INSTALL_PLATFORM_UNSUPPORTED	This installation package is not supported by this processor type. Contact your product vendor.
1634	0x00000662	ERROR_INSTALL_NOTUSED	Component not used on this computer.
1635	0x00000663	ERROR_PATCH_PACKAGE_OPEN_FAILED	This patch package could not be opened. Verify that the patch package exists and that you can access it, or contact the application vendor to verify that this is a valid Windows Installer patch package.
1636	0x00000664	ERROR_PATCH_PACKAGE_INVALID	This patch package could not be opened. Contact the application vendor to verify that this is a valid Windows Installer patch package.
1637	0x00000665	ERROR_PATCH_PACKAGE_UNSUPPORTED.	This patch package cannot be processed by the Windows Installer service. You must install a Windows service pack that contains a newer version of the Windows Installer service.
1638	0x00000666	ERROR_PRODUCT_VERSION	Another version of this product is already installed. Installation of this version cannot continue. To configure or remove the existing version of this product, use Add/Remove Programs on the Control Panel.
1639	0x00000667	ERROR_INVALID_COMMAND_LINE	Invalid command line argument. Consult the Windows Installer SDK for detailed command line help.
1640	0x00000668	ERROR_INSTALL_REMOTE_DISALLOWED	Only administrators have permission to add, remove, or configure server software during a Terminal Services remote session. If you want to install or configure software on the server, contact your network administrator.
1641	0x00000669	ERROR_SUCCESS_REBOOT_INITIATED	The requested operation completed successfully. The system will be restarted so the changes can take effect.
1642	0x0000066A	ERROR_PATCH_TARGET_NOT_FOUND	The upgrade patch cannot be installed by the Windows Installer service because the program to be upgraded may be missing, or the upgrade patch may update a different version of the program. Verify that the program to be upgraded exists on your computer and that you have the correct upgrade patch.

Error			Description
decimal	Hexadecimal	Name	
1643	0x0000066B	ERROR_PATCH_PACKAGE_REJECTED	The patch package is not permitted by system policy. It is not signed with an appropriate certificate.
1644	0x0000066C	ERROR_INSTALL_TRANSFORM_REJECTED	One or more customizations are not permitted by system policy. They are not signed with an appropriate certificate.
1700	0x000006A4	RPC_S_INVALID_STRING_BINDING	The string binding is invalid.
1701	0x000006A5	RPC_S_WRONG_KIND_OF_BINDING	The binding handle is not the correct type.
1702	0x000006A6	RPC_S_INVALID_BINDING	The binding handle is invalid.
1703	0x000006A7	RPC_S_PROTSEQ_NOT_SUPPORTED	The RPC protocol sequence is not supported.
1704	0x000006A8	RPC_S_INVALID_RPC_PROTSEQ	The RPC protocol sequence is invalid.
1705	0x000006A9	RPC_S_INVALID_STRING_UUID	The string universal unique identifier (UUID) is invalid.
1706	0x000006AA	RPC_S_INVALID_ENDPOINT_FORMAT	The endpoint format is invalid.
1707	0x000006AB	RPC_S_INVALID_NET_ADDR	The network address is invalid.
1708	0x000006AC	RPC_S_NO_ENDPOINT_FOUND	No endpoint was found.
1709	0x000006AD	RPC_S_INVALID_TIMEOUT	The timeout value is invalid.
1710	0x000006AE	RPC_S_OBJECT_NOT_FOUND	The object universal unique identifier (UUID) was not found.
1711	0x000006AF	RPC_S_ALREADY_REGISTERED	The object universal unique identifier (UUID) has already been registered.
1712	0x000006B0	RPC_S_TYPE_ALREADY_REGISTERED	The type universal unique identifier (UUID) has already been registered.
1713	0x000006B1	RPC_S_ALREADY_LISTENING	The RPC server is already listening.
1714	0x000006B2	RPC_S_NO_PROTSEQS_REGISTERED	No protocol sequences have been registered.
1715	0x000006B3	RPC_S_NOT_LISTENING	The RPC server is not listening.
1716	0x000006B4	RPC_S_UNKNOWN_MGR_TYPE	The manager type is unknown.
1717	0x000006B5	RPC_S_UNKNOWN_IF	The interface is unknown.
1718	0x000006B6	RPC_S_NO_BINDINGS	There are no bindings.
1719	0x000006B7	RPC_S_NO_PROTSEQS	There are no protocol sequences.
1720	0x000006B8	RPC_S_CANT_CREATE_ENDPOINT	The endpoint cannot be created.
1721	0x000006B9	RPC_S_OUT_OF_RESOURCES	Not enough resources are available to complete this operation.
1722	0x000006BA	RPC_S_SERVER_UNAVAILABLE	The RPC server is unavailable.
1723	0x000006BB	RPC_S_SERVER_TOO_BUSY	The RPC server is too busy to complete this operation.
1724	0x000006BC	RPC_S_INVALID_NETWORK_OPTIONS	The network options are invalid.
1725	0x000006BD	RPC_S_NO_CALL_ACTIVE	There are no remote procedure calls active on this thread.
1726	0x000006BE	RPC_S_CALL_FAILED	The remote procedure call failed.
1727	0x000006BF	RPC_S_CALL_FAILED_DNE	The remote procedure call failed and did not execute.
1728	0x000006C0	RPC_S_PROTOCOL_ERROR	A remote procedure call (RPC) protocol error occurred.
1730	0x000006C2	RPC_S_UNSUPPORTED_TRANS_SYN	The transfer syntax is not supported by the RPC server.
1732	0x000006C4	RPC_S_UNSUPPORTED_TYPE	The universal unique identifier (UUID) type is not supported.
1733	0x000006C5	RPC_S_INVALID_TAG	The tag is invalid.
1734	0x000006C6	RPC_S_INVALID_BOUND	The array bounds are invalid.
1735	0x000006C7	RPC_S_NO_ENTRY_NAME	The binding does not contain an entry name.
1736	0x000006C8	RPC_S_INVALID_NAME_SYNTAX	The name syntax is invalid.
1737	0x000006C9	RPC_S_UNSUPPORTED_NAME_SYNTAX	The name syntax is not supported.
1739	0x000006CB	RPC_S_UUID_NO_ADDRESS	No network address is available to use to construct a universal unique identifier (UUID).
1740	0x000006CC	RPC_S_DUPLICATE_ENDPOINT	The endpoint is a duplicate.
1741	0x000006CD	RPC_S_UNKNOWN_AUTHN_TYPE	The authentication type is unknown.
1742	0x000006CE	RPC_S_MAX_CALLS_TOO_SMALL	The maximum number of calls is too small.
1743	0x000006CF	RPC_S_STRING_TOO_LONG	The string is too long.
1744	0x000006D0	RPC_S_PROTSEQ_NOT_FOUND	The RPC protocol sequence was not found.
1745	0x000006D1	RPC_S_PROCNUM_OUT_OF_RANGE	The procedure number is out of range.

Error			Description
decimal	Hexadecimal	Name	
1746	0x000006D2	RPC_S_BINDING_HAS_NO_AUTH	The binding does not contain any authentication information.
1747	0x000006D3	RPC_S_UNKNOWN_AUTHN_SERVICE	The authentication service is unknown.
1748	0x000006D4	RPC_S_UNKNOWN_AUTHN_LEVEL	The authentication level is unknown.
1749	0x000006D5	RPC_S_INVALID_AUTH_IDENTITY	The security context is invalid.
1750	0x000006D6	RPC_S_UNKNOWN_AUTHZ_SERVICE	The authorization service is unknown.
1751	0x000006D7	EPT_S_INVALID_ENTRY	The entry is invalid.
1752	0x000006D8	EPT_S_CANT_PERFORM_OP	The server endpoint cannot perform the operation.
1753	0x000006D9	EPT_S_NOT_REGISTERED	There are no more endpoints available from the endpoint mapper.
1754	0x000006DA	RPC_S_NOTHING_TO_EXPORT	No interfaces have been exported.
1755	0x000006DB	RPC_S_INCOMPLETE_NAME	The entry name is incomplete.
1756	0x000006DC	RPC_S_INVALID_VERS_OPTION	The version option is invalid.
1757	0x000006DD	RPC_S_NO_MORE_MEMBERS	There are no more members.
1758	0x000006DE	RPC_S_NOT_ALL_OBJS_UNEXPORTED	There is nothing to unexport.
1759	0x000006DF	RPC_S_INTERFACE_NOT_FOUND	The interface was not found.
1760	0x000006E0	RPC_S_ENTRY_ALREADY_EXISTS	The entry already exists.
1761	0x000006E1	RPC_S_ENTRY_NOT_FOUND	The entry is not found.
1762	0x000006E2	RPC_S_NAME_SERVICE_UNAVAILABLE	The name service is unavailable.
1763	0x000006E3	RPC_S_INVALID_NAF_ID	The network address family is invalid.
1764	0x000006E4	RPC_S_CANNOT_SUPPORT	The requested operation is not supported.
1765	0x000006E5	RPC_S_NO_CONTEXT_AVAILABLE	No security context is available to allow impersonation.
1766	0x000006E6	RPC_S_INTERNAL_ERROR	An internal error occurred in a remote procedure call (RPC).
1767	0x000006E7	RPC_S_ZERO_DIVIDE	The RPC server attempted an integer division by zero.
1768	0x000006E8	RPC_S_ADDRESS_ERROR	An addressing error occurred in the RPC server.
1769	0x000006E9	RPC_S_FP_DIV_ZERO	A floating-point operation at the RPC server caused a division by zero.
1770	0x000006EA	RPC_S_FP_UNDERFLOW	A floating-point underflow occurred at the RPC server.
1771	0x000006EB	RPC_S_FP_OVERFLOW	A floating-point overflow occurred at the RPC server.
1772	0x000006EC	RPC_X_NO_MORE_ENTRIES	The list of RPC servers available for the binding of auto handles has been exhausted.
1773	0x000006ED	RPC_X_SS_CHAR_TRANS_OPEN_FAIL	Unable to open the character translation table file.
1774	0x000006EE	RPC_X_SS_CHAR_TRANS_SHORT_FILE	The file containing the character translation table has fewer than 512 bytes.
1775	0x000006EF	RPC_X_SS_IN_NULL_CONTEXT	A null context handle was passed from the client to the host during a remote procedure call.
1777	0x000006F1	RPC_X_SS_CONTEXT_DAMAGED	The context handle changed during a remote procedure call.
1778	0x000006F2	RPC_X_SS_HANDLES_MISMATCH	The binding handles passed to a remote procedure call do not match.
1779	0x000006F3	RPC_X_SS_CANNOT_GET_CALL_HANDLE	The stub is unable to get the remote procedure call handle.
1780	0x000006F4	RPC_X_NULL_REF_POINTER	A null reference pointer was passed to the stub.
1781	0x000006F5	RPC_X_ENUM_VALUE_OUT_OF_RANGE	The enumeration value is out of range.
1782	0x000006F6	RPC_X_BYTE_COUNT_TOO_SMALL	The byte count is too small.
1783	0x000006F7	RPC_X_BAD_STUB_DATA	The stub received bad data.
1784	0x000006F8	ERROR_INVALID_USER_BUFFER	The supplied user buffer is not valid for the requested operation.
1785	0x000006F9	ERROR_UNRECOGNIZED_MEDIA	The disk media is not recognized. It may not be formatted.
1786	0x000006FA	ERROR_NO_TRUST_LSA_SECRET	The workstation does not have a trust secret.
1787	0x000006FB	ERROR_NO_TRUST_SAM_ACCOUNT	The security database on the server does not have a computer account for this workstation trust relationship.
1788	0x000006FC	ERROR_TRUSTED_DOMAIN_FAILURE	The trust relationship between the primary domain and the trusted domain failed.

Error			Description
decimal	Hexadecimal	Name	
1789	0x000006FD	ERROR_TRUSTED_RELATIONSHIP_FAILURE	The trust relationship between this workstation and the primary domain failed.
1790	0x000006FE	ERROR_TRUST_FAILURE	The network logon failed.
1791	0x000006FF	RPC_S_CALL_IN_PROGRESS	A remote procedure call is already in progress for this thread.
1792	0x00000700	ERROR_NETLOGON_NOT_STARTED	An attempt was made to logon, but the network logon service was not started.
1793	0x00000701	ERROR_ACCOUNT_EXPIRED	The user's account has expired.
1794	0x00000702	ERROR_REDIRECTOR_HAS_OPEN_HANDLES	The redirector is in use and cannot be unloaded.
1795	0x00000703	ERROR_PRINTER_DRIVER_ALREADY_INSTALLED	The specified printer driver is already installed.
1796	0x00000704	ERROR_UNKNOWN_PORT	The specified port is unknown.
1797	0x00000705	ERROR_UNKNOWN_PRINTER_DRIVER	The printer driver is unknown.
1798	0x00000706	ERROR_UNKNOWN_PRINTPROCESSOR	The print processor is unknown.
1799	0x00000707	ERROR_INVALID_SEPARATOR_FILE	The specified separator file is invalid.

Error			Description
decimal	Hexadecimal	Name	
1800	0x00000708	ERROR_INVALID_PRIORITY	The specified priority is invalid.
1801	0x00000709	ERROR_INVALID_PRINTER_NAME	The printer name is invalid.
1802	0x0000070A	ERROR_PRINTER_ALREADY_EXISTS	The printer already exists.
1803	0x0000070B	ERROR_INVALID_PRINTER_COMMAND	The printer command is invalid.
1804	0x0000070C	ERROR_INVALID_DATATYPE	The specified datatype is invalid.
1805	0x0000070D	ERROR_INVALID_ENVIRONMENT	The environment specified is invalid.
1806	0x0000070E	RPC_S_NO_MORE_BINDINGS	There are no more bindings.
1807	0x0000070F	ERROR_NOLOGON_INTERDOMAIN_TRUST_ACCOUNT	The account used is an interdomain trust account. Use your global user account or local user account to access this server.
1808	0x00000710	ERROR_NOLOGON_WORKSTATION_TRUST_ACCOUNT	The account used is a computer account. Use your global user account or local user account to access this server.
1809	0x00000711	ERROR_NOLOGON_SERVER_TRUST_ACCOUNT	The account used is a server trust account. Use your global user account or local user account to access this server.
1810	0x00000712	ERROR_DOMAIN_TRUST_INCONSISTENT	The name or security ID (SID) of the domain specified is inconsistent with the trust information for that domain.
1811	0x00000713	ERROR_SERVER_HAS_OPEN_HANDLES	The server is in use and cannot be unloaded.
1812	0x00000714	ERROR_RESOURCE_DATA_NOT_FOUND	The specified image file did not contain a resource section.
1813	0x00000715	ERROR_RESOURCE_TYPE_NOT_FOUND	The specified resource type cannot be found in the image file.
1814	0x00000716	ERROR_RESOURCE_NAME_NOT_FOUND	The specified resource name cannot be found in the image file.
1815	0x00000717	ERROR_RESOURCE_LANG_NOT_FOUND	The specified resource language ID cannot be found in the image file.
1816	0x00000718	ERROR_NOT_ENOUGH_QUOTA	Not enough quota is available to process this command.
1817	0x00000719	RPC_S_NO_INTERFACES	No interfaces have been registered.
1818	0x0000071A	RPC_S_CALL_CANCELLED	The remote procedure call was cancelled.
1819	0x0000071B	RPC_S_BINDING_INCOMPLETE	The binding handle does not contain all required information.
1820	0x0000071C	RPC_S_COMM_FAILURE	A communications failure occurred during a remote procedure call.
1821	0x0000071D	RPC_S_UNSUPPORTED_AUTHN_LEVEL	The requested authentication level is not supported.
1822	0x0000071E	RPC_S_NO_PRINC_NAME	No principal name registered.
1823	0x0000071F	RPC_S_NOT_RPC_ERROR	The error specified is not a valid Windows RPC error code.
1824	0x00000720	RPC_S_UUID_LOCAL_ONLY	A UUID that is valid only on this computer has been allocated.
1825	0x00000721	RPC_S_SEC_PKG_ERROR	A security package specific error occurred.
1826	0x00000722	RPC_S_NOT_CANCELLED	Thread is not canceled.

Error			Description
decimal	Hexadecimal	Name	
1827	0x00000723	RPC_X_INVALID_ES_ACTION	Invalid operation on the encoding/decoding handle.
1828	0x00000724	RPC_X_WRONG_ES_VERSION	Incompatible version of the serializing package.
1829	0x00000725	RPC_X_WRONG_STUB_VERSION	Incompatible version of the RPC stub.
1830	0x00000726	RPC_X_INVALID_PIPE_OBJECT	The RPC pipe object is invalid or corrupted.
1831	0x00000727	RPC_X_WRONG_PIPE_ORDER	An invalid operation was attempted on an RPC pipe object.
1832	0x00000728	RPC_X_WRONG_PIPE_VERSION	Unsupported RPC pipe version.
1898	0x0000076A	RPC_S_GROUP_MEMBER_NOT_FOUND	The group member was not found.
1899	0x0000076B	EPT_S_CANT_CREATE	The endpoint mapper database entry could not be created.
1900	0x0000076C	RPC_S_INVALID_OBJECT	The object universal unique identifier (UUID) is the nil UUID.
1901	0x0000076D	ERROR_INVALID_TIME	The specified time is invalid.
1902	0x0000076E	ERROR_INVALID_FORM_NAME	The specified form name is invalid.
1903	0x0000076F	ERROR_INVALID_FORM_SIZE	The specified form size is invalid.
1904	0x00000770	ERROR_ALREADY_WAITING	The specified printer handle is already being waited on
1905	0x00000771	ERROR_PRINTER_DELETED	The specified printer has been deleted.
1906	0x00000772	ERROR_INVALID_PRINTER_STATE	The state of the printer is invalid.
1907	0x00000773	ERROR_PASSWORD_MUST_CHANGE	The user's password must be changed before logging on the first time.
1908	0x00000774	ERROR_DOMAIN_CONTROLLER_NOT_FOUND	Could not find the domain controller for this domain.
1909	0x00000775	ERROR_ACCOUNT_LOCKED_OUT	The referenced account is currently locked out and may not be logged on to.
1910	0x00000776	OR_INVALID_OXID	The object exporter specified was not found.
1911	0x00000777	OR_INVALID_OID	The object specified was not found.
1912	0x00000778	OR_INVALID_SET	The object resolver set specified was not found.
1913	0x00000779	RPC_S_SEND_INCOMPLETE	Some data remains to be sent in the request buffer.
1914	0x0000077A	RPC_S_INVALID_ASYNC_HANDLE	Invalid asynchronous remote procedure call handle.
1915	0x0000077B	RPC_S_INVALID_ASYNC_CALL	Invalid asynchronous RPC call handle for this operation.
1916	0x0000077C	RPC_X_PIPE_CLOSED	The RPC pipe object has already been closed.
1917	0x0000077D	RPC_X_PIPE_DISCIPLINE_ERROR	The RPC call completed before all pipes were processed.
1918	0x0000077E	RPC_X_PIPE_EMPTY	No more data is available from the RPC pipe.
1919	0x0000077F	ERROR_NO_SITENAME	No site name is available for this machine.
1920	0x00000780	ERROR_CANT_ACCESS_FILE	The file can not be accessed by the system.
1921	0x00000781	ERROR_CANT_RESOLVE_FILENAME	The name of the file cannot be resolved by the system.
1922	0x00000782	RPC_S_ENTRY_TYPE_MISMATCH	The entry is not of the expected type.
1923	0x00000783	RPC_S_NOT_ALL_OBJS_EXPORTED	Not all object UUIDs could be exported to the specified entry.
1924	0x00000784	RPC_S_INTERFACE_NOT_EXPORTED	Interface could not be exported to the specified entry.
1925	0x00000785	RPC_S_PROFILE_NOT_ADDED	The specified profile entry could not be added.
1926	0x00000786	RPC_S_PRF_ELT_NOT_ADDED	The specified profile element could not be added.
1927	0x00000787	RPC_S_PRF_ELT_NOT_REMOVED	The specified profile element could not be removed.
1928	0x00000788	RPC_S_GRP_ELT_NOT_ADDED	The group element could not be added.
1929	0x00000789	RPC_S_GRP_ELT_NOT_REMOVED	The group element could not be removed.
1930	0x0000078A	ERROR_KM_DRIVER_BLOCKED	The printer driver is not compatible with a policy enabled on your computer that blocks NT 4.0 drivers.

Error			Description
decimal	Hexadecimal	Name	
2000	0x000007D0	ERROR_INVALID_PIXEL_FORMAT	The pixel format is invalid.
2001	0x000007D1	ERROR_BAD_DRIVER	The specified driver is invalid.

Error			Description
decimal	Hexadecimal	Name	
2002	0x000007D2	ERROR_INVALID_WINDOW_STYLE	The window style or class attribute is invalid for this operation.
2003	0x000007D3	ERROR_METAFILE_NOT_SUPPORTED	The requested metafile operation is not supported.
2004	0x000007D4	ERROR_TRANSFORM_NOT_SUPPORTED	The requested transformation operation is not supported.
2005	0x000007D5	ERROR_CLIPPING_NOT_SUPPORTED	The requested clipping operation is not supported.
2010	0x000007DA	ERROR_INVALID_CMM	The specified color management module is invalid.
2011	0x000007DB	ERROR_INVALID_PROFILE	The specified color profile is invalid.
2012	0x000007DC	ERROR_TAG_NOT_FOUND	The specified tag was not found.
2013	0x000007DD	ERROR_TAG_NOT_PRESENT	A required tag is not present.
2014	0x000007DE	ERROR_DUPLICATE_TAG	The specified tag is already present.
2015	0x000007DF	ERROR_PROFILE_NOT_ASSOCIATED_WITH_DEVICE	The specified color profile is not associated with any device.
2016	0x000007E0	ERROR_PROFILE_NOT_FOUND	The specified color profile was not found.
2017	0x000007E1	ERROR_INVALID_COLORSPACE	The specified color space is invalid.
2018	0x000007E2	ERROR_ICM_NOT_ENABLED	Image Color Management is not enabled.
2019	0x000007E3	ERROR_DELETING_ICM_XFORM	There was an error while deleting the color transform.
2020	0x000007E4	ERROR_INVALID_TRANSFORM	The specified color transform is invalid.
2021	0x000007E5	ERROR_COLORSPACE_MISMATCH	The specified transform does not match the bitmap's color space.
2022	0x000007E6	ERROR_INVALID_COLORINDEX	The specified named color index is not present in the profile.
2108	0x0000083C	ERROR_CONNECTED_OTHER_PASSWORD	The network connection was made successfully, but the user had to be prompted for a password other than the one originally specified.
2202	0x0000089A	ERROR_BAD_USERNAME	The specified username is invalid.
2250	0x000008CA	ERROR_NOT_CONNECTED	This network connection does not exist.
2401	0x00000961	ERROR_OPEN_FILES	This network connection has files open or requests pending.
2402	0x00000962	ERROR_ACTIVE_CONNECTIONS	Active connections still exist.
2404	0x00000964	ERROR_DEVICE_IN_USE	The device is in use by an active process and cannot be disconnected.
2500	0x000009C4	ERROR_PKINIT_FAILURE	The kerberos protocol encountered an error while validating the KDC certificate during smartcard logon.
2501	0x000009C5	ERROR_SMARTCARD_SUBSYSTEM_FAILURE	The kerberos protocol encountered an error while attempting to utilize the smartcard subsystem.

Error			Description
decimal	Hexadecimal	Name	
3000	0x00000BB8	ERROR_UNKNOWN_PRINT_MONITOR	The specified print monitor is unknown.
3001	0x00000BB9	ERROR_PRINTER_DRIVER_IN_USE	The specified printer driver is currently in use.
3002	0x00000BBA	ERROR_SPOOL_FILE_NOT_FOUND	The spool file was not found.
3003	0x00000BBB	ERROR_SPL_NO_STARTDOC	A StartDocPrinter call was not issued.
3004	0x00000BBC	ERROR_SPL_NO_ADDJOB	An AddJob call was not issued.
3005	0x00000BBD	ERROR_PRINT_PROCESSOR_ALREADY_INSTALLED	The specified print processor has already been installed.
3006	0x00000BBE	ERROR_PRINT_MONITOR_ALREADY_INSTALLED	The specified print monitor has already been installed.
3007	0x00000BBF	ERROR_INVALID_PRINT_MONITOR	The specified print monitor does not have the required functions.
3008	0x00000BC0	ERROR_PRINT_MONITOR_IN_USE	The specified print monitor is currently in use.
3009	0x00000BC1	ERROR_PRINTER_HAS_JOBS_QUEUED	The requested operation is not allowed when there are jobs queued to the printer.
3010	0x00000BC2	ERROR_SUCCESS_REBOOT_REQUIRED	The requested operation is successful. Changes will not be effective until the system is rebooted.
3011	0x00000BC3	ERROR_SUCCESS_RESTART_REQUIRED	The requested operation is successful. Changes will not be effective until the service is restarted.
3012	0x00000BC4	ERROR_PRINTER_NOT_FOUND	No printers were found.

Error			Description
decimal	Hexadecimal	Name	
4000	0x00000FA0	ERROR_WINS_INTERNAL	WINS encountered an error while processing the command.
4001	0x00000FA1	ERROR_CAN_NOT_DEL_LOCAL_WINS	The local WINS can not be deleted.
4002	0x00000FA2	ERROR_STATIC_INIT	The importation from the file failed.
4003	0x00000FA3	ERROR_INC_BACKUP	The backup failed. Was a full backup done before?
4004	0x00000FA4	ERROR_FULL_BACKUP	The backup failed. Check the directory to which you are backing the database.
4005	0x00000FA5	ERROR_REC_NON_EXISTENT	The name does not exist in the WINS database.
4006	0x00000FA6	ERROR_RPL_NOT_ALLOWED	Replication with a nonconfigured partner is not allowed.
4100	0x00001004	ERROR_DHCP_ADDRESS_CONFLICT	The DHCP client has obtained an IP address that is already in use on the network. The local interface will be disabled until the DHCP client can obtain a new address.
4200	0x00001068	ERROR_WMI_GUID_NOT_FOUND	The GUID passed was not recognized as valid by a WMI data provider.
4201	0x00001069	ERROR_WMI_INSTANCE_NOT_FOUND	The instance name passed was not recognized as valid by a WMI data provider.
4202	0x0000106A	ERROR_WMI_ITEMID_NOT_FOUND	The data item ID passed was not recognized as valid by a WMI data provider.
4203	0x0000106B	ERROR_WMI_TRY_AGAIN	The WMI request could not be completed and should be retried.
4204	0x0000106C	ERROR_WMI_DP_NOT_FOUND	The WMI data provider could not be located.
4205	0x0000106D	ERROR_WMI_UNRESOLVED_INSTANCE_REF	The WMI data provider references an instance set that has not been registered.
4206	0x0000106E	ERROR_WMI_ALREADY_ENABLED	The WMI data block or event notification has already been enabled.
4207	0x0000106F	ERROR_WMI_GUID_DISCONNECTED	The WMI data block is no longer available.
4208	0x00001070	ERROR_WMI_SERVER_UNAVAILABLE	The WMI data service is not available.
4209	0x00001071	ERROR_WMI_DP_FAILED	The WMI data provider failed to carry out the request.
4210	0x00001072	ERROR_WMI_INVALID_MOF	The WMI MOF information is not valid.
4211	0x00001073	ERROR_WMI_INVALID_REGINFO	The WMI registration information is not valid.
4212	0x00001074	ERROR_WMI_ALREADY_DISABLED	The WMI data block or event notification has already been disabled.
4213	0x00001075	ERROR_WMI_READ_ONLY	The WMI data item or data block is read only.
4214	0x00001076	ERROR_WMI_SET_FAILURE	The WMI data item or data block could not be changed.
4300	0x000010CC	ERROR_INVALID_MEDIA	The media identifier does not represent a valid medium.
4301	0x000010CD	ERROR_INVALID_LIBRARY	The library identifier does not represent a valid library.
4302	0x000010CE	ERROR_INVALID_MEDIA_POOL	The media pool identifier does not represent a valid media pool.
4303	0x000010CF	ERROR_DRIVE_MEDIA_MISMATCH	The drive and medium are not compatible or exist in different libraries.
4304	0x000010D0	ERROR_MEDIA_OFFLINE	The medium currently exists in an offline library and must be online to perform this operation.
4305	0x000010D1	ERROR_LIBRARY_OFFLINE	The operation cannot be performed on an offline library.
4306	0x000010D2	ERROR_EMPTY	The library, drive, or media pool is empty.
4307	0x000010D3	ERROR_NOT_EMPTY	The library, drive, or media pool must be empty to perform this operation.
4308	0x000010D4	ERROR_MEDIA_UNAVAILABLE	No media is currently available in this media pool or library.
4309	0x000010D5	ERROR_RESOURCE_DISABLED	A resource required for this operation is disabled.
4310	0x000010D6	ERROR_INVALID_CLEANER	The media identifier does not represent a valid cleaner.
4311	0x000010D7	ERROR_UNABLE_TO_CLEAN	The drive cannot be cleaned or does not support cleaning.
4312	0x000010D8	ERROR_OBJECT_NOT_FOUND	The object identifier does not represent a valid object.
4313	0x000010D9	ERROR_DATABASE_FAILURE	Unable to read from or write to the database.

Error			Description
decimal	Hexadecimal	Name	
4314	0x000010DA	ERROR_DATABASE_FULL	The database is full.
4315	0x000010DB	ERROR_MEDIA_INCOMPATIBLE	The medium is not compatible with the device or media pool.
4316	0x000010DC	ERROR_RESOURCE_NOT_PRESENT	The resource required for this operation does not exist.
4317	0x000010DD	ERROR_INVALID_OPERATION	The operation identifier is not valid.
4318	0x000010DE	ERROR_MEDIA_NOT_AVAILABLE	The media is not mounted or ready for use.
4319	0x000010DF	ERROR_DEVICE_NOT_AVAILABLE	The device is not ready for use.
4320	0x000010E0	ERROR_REQUEST_REFUSED	The operator or administrator has refused the request.
4321	0x000010E1	ERROR_INVALID_DRIVE_OBJECT	The drive identifier does not represent a valid drive.
4322	0x000010E2	ERROR_LIBRARY_FULL	Library is full. No slot is available for use.
4323	0x000010E3	ERROR_MEDIUM_NOT_ACCESSIBLE	The transport cannot access the medium.
4324	0x000010E4	ERROR_UNABLE_TO_LOAD_MEDIUM	Unable to load the medium into the drive.
4325	0x000010E5	ERROR_UNABLE_TO_INVENTORY_DRIVE	Unable to retrieve status about the drive.
4326	0x000010E6	ERROR_UNABLE_TO_INVENTORY_SLOT	Unable to retrieve status about the slot.
4327	0x000010E7	ERROR_UNABLE_TO_INVENTORY_TRANSPORT	Unable to retrieve status about the transport.
4328	0x000010E8	ERROR_TRANSPORT_FULL	Cannot use the transport because it is already in use.
4329	0x000010E9	ERROR_CONTROLLING_IEPORT	Unable to open or close the inject/eject port.
4330	0x000010EA	ERROR_UNABLE_TO_EJECT_MOUNTED_MEDIA	Unable to eject the media because it is in a drive.
4331	0x000010EB	ERROR_CLEANER_SLOT_SET	A cleaner slot is already reserved.
4332	0x000010EC	ERROR_CLEANER_SLOT_NOT_SET	A cleaner slot is not reserved.
4333	0x000010ED	ERROR_CLEANER_CARTRIDGE_SPENT	The cleaner cartridge has performed the maximum number of drive cleanings.
4334	0x000010EE	ERROR_UNEXPECTED_OMID	Unexpected on-medium identifier.
4335	0x000010EF	ERROR_CANT_DELETE_LAST_ITEM	The last remaining item in this group or resource cannot be deleted.
4336	0x000010F0	ERROR_MESSAGE_EXCEEDS_MAX_SIZE	The message provided exceeds the maximum size allowed for this parameter.
4337	0x000010F1	ERROR_VOLUME_CONTAINS_SYS_FILES	The volume contains system or paging files.
4338	0x000010F2	ERROR_INDIGENOUS_TYPE	The media type cannot be removed from this library since at least one drive in the library reports it can support this media type.
4339	0x000010F3	ERROR_NO_SUPPORTING_DRIVES	This offline media cannot be mounted on this system since no enabled drives are present which can be used.
4340	0x000010F4	ERROR_CLEANER_CARTRIDGE_INSTALLED	A cleaner cartridge is present in the tape library.
4350	0x000010FE	ERROR_FILE_OFFLINE	The remote storage service was not able to recall the file.
4351	0x000010FF	ERROR_REMOTE_STORAGE_NOT_ACTIVE	The remote storage service is not operational at this time.
4352	0x00001100	ERROR_REMOTE_STORAGE_MEDIA_ERROR	The remote storage service encountered a media error.
4390	0x00001126	ERROR_NOT_A_REPARSE_POINT	The file or directory is not a reparse point.
4391	0x00001127	ERROR_REPARSE_ATTRIBUTE_CONFLICT	The reparse point attribute cannot be set because it conflicts with an existing attribute.
4392	0x00001128	ERROR_INVALID_REPARSE_DATA	The data present in the reparse point buffer is invalid.
4393	0x00001129	ERROR_REPARSE_TAG_INVALID	The tag present in the reparse point buffer is invalid.
4394	0x0000112A	ERROR_REPARSE_TAG_MISMATCH	There is a mismatch between the tag specified in the request and the tag present in the reparse point.
4500	0x00001194	ERROR_VOLUME_NOT_SIS_ENABLED	Single Instance Storage is not available on this volume.

Error			Description
decimal	Hexadecimal	Name	
5001	0x00001389	ERROR_DEPENDENT_RESOURCE_EXISTS	The cluster resource cannot be moved to another group because other resources are dependent on it.

Error			Description
decimal	Hexadecimal	Name	
5002	0x0000138A	ERROR_DEPENDENCY_NOT_FOUND	The cluster resource dependency cannot be found.
5003	0x0000138B	ERROR_DEPENDENCY_ALREADY_EXISTS	The cluster resource cannot be made dependent on the specified resource because it is already dependent.
5004	0x0000138C	ERROR_RESOURCE_NOT_ONLINE	The cluster resource is not online.
5005	0x0000138D	ERROR_HOST_NODE_NOT_AVAILABLE	A cluster node is not available for this operation.
5006	0x0000138E	ERROR_RESOURCE_NOT_AVAILABLE	The cluster resource is not available.
5007	0x0000138F	ERROR_RESOURCE_NOT_FOUND	The cluster resource could not be found.
5008	0x00001390	ERROR_SHUTDOWN_CLUSTER	The cluster is being shut down.
5009	0x00001391	ERROR_CANT_EVICT_ACTIVE_NODE	A cluster node cannot be evicted from the cluster unless the node is down.
5010	0x00001392	ERROR_OBJECT_ALREADY_EXISTS	The object already exists.
5011	0x00001393	ERROR_OBJECT_IN_LIST	The object is already in the list.
5012	0x00001394	ERROR_GROUP_NOT_AVAILABLE	The cluster group is not available for any new requests.
5013	0x00001395	ERROR_GROUP_NOT_FOUND	The cluster group could not be found.
5014	0x00001396	ERROR_GROUP_NOT_ONLINE	The operation could not be completed because the cluster group is not online.
5015	0x00001397	ERROR_HOST_NODE_NOT_RESOURCE_OWNER	The cluster node is not the owner of the resource.
5016	0x00001398	ERROR_HOST_NODE_NOT_GROUP_OWNER	The cluster node is not the owner of the group.
5017	0x00001399	ERROR_RESMON_CREATE_FAILED	The cluster resource could not be created in the specified resource monitor.
5018	0x0000139A	ERROR_RESMON_ONLINE_FAILED	The cluster resource could not be brought online by the resource monitor.
5019	0x0000139B	ERROR_RESOURCE_ONLINE	The operation could not be completed because the cluster resource is online.
5020	0x0000139C	ERROR_QUORUM_RESOURCE	The cluster resource could not be deleted or brought offline because it is the quorum resource.
5021	0x0000139D	ERROR_NOT_QUORUM_CAPABLE	The cluster could not make the specified resource a quorum resource because it is not capable of being a quorum resource.
5022	0x0000139E	ERROR_CLUSTER_SHUTTING_DOWN	The cluster software is shutting down.
5023	0x0000139F	ERROR_INVALID_STATE	The group or resource is not in the correct state to perform the requested operation.
5024	0x000013A0	ERROR_RESOURCE_PROPERTIES_STORED	The properties were stored but not all changes will take effect until the next time the resource is brought online.
5025	0x000013A1	ERROR_NOT_QUORUM_CLASS	The cluster could not make the specified resource a quorum resource because it does not belong to a shared storage class.
5026	0x000013A2	ERROR_CORE_RESOURCE	The cluster resource could not be deleted since it is a core resource.
5027	0x000013A3	ERROR_QUORUM_RESOURCE_ONLINE_FAILED	The quorum resource failed to come online.
5028	0x000013A4	ERROR_QUORUMLOG_OPEN_FAILED	The quorum log could not be created or mounted successfully.
5029	0x000013A5	ERROR_CLUSTERLOG_CORRUPT	The cluster log is corrupt.
5030	0x000013A6	ERROR_CLUSTERLOG_RECORD_EXCEEDS_MAXSIZE	The record could not be written to the cluster log since it exceeds the maximum size.
5031	0x000013A7	ERROR_CLUSTERLOG_EXCEEDS_MAXSIZE	The cluster log exceeds its maximum size.
5032	0x000013A8	ERROR_CLUSTERLOG_CHKPOINT_NOT_FOUND	No checkpoint record was found in the cluster log.
5033	0x000013A9	ERROR_CLUSTERLOG_NOT_ENOUGH_SPACE	The minimum required disk space needed for logging is not available.
5034	0x000013AA	ERROR_QUORUM_OWNER_ALIVE	The cluster node failed to take control of the quorum resource because the resource is owned by another active node.
5035	0x000013AB	ERROR_NETWORK_NOT_AVAILABLE	A cluster network is not available for this operation.
5036	0x000013AC	ERROR_NODE_NOT_AVAILABLE	A cluster node is not available for this operation.
5037	0x000013AD	ERROR_ALL_NODES_NOT_AVAILABLE	All cluster nodes must be running to perform this operation.
5038	0x000013AE	ERROR_RESOURCE_FAILED	A cluster resource failed.

Error			Description
decimal	Hexadecimal	Name	
5039	0x000013AF	ERROR_CLUSTER_INVALID_NODE	The cluster node is not valid.
5040	0x000013B0	ERROR_CLUSTER_NODE_EXISTS	The cluster node already exists.
5041	0x000013B1	ERROR_CLUSTER_JOIN_IN_PROGRESS	A node is in the process of joining the cluster.
5042	0x000013B2	ERROR_CLUSTER_NODE_NOT_FOUND	The cluster node was not found.
5043	0x000013B3	ERROR_CLUSTER_LOCAL_NODE_NOT_FOUND	The cluster local node information was not found.
5044	0x000013B4	ERROR_CLUSTER_NETWORK_EXISTS	The cluster network already exists.
5045	0x000013B5	ERROR_CLUSTER_NETWORK_NOT_FOUND	The cluster network was not found.
5046	0x000013B6	ERROR_CLUSTER_NETINTERFACE_EXISTS	The cluster network interface already exists.
5047	0x000013B7	ERROR_CLUSTER_NETINTERFACE_NOT_FOUND	The cluster network interface was not found.
5048	0x000013B8	ERROR_CLUSTER_INVALID_REQUEST	The cluster request is not valid for this object.
5049	0x000013B9	ERROR_CLUSTER_INVALID_NETWORK_PROVIDER	The cluster network provider is not valid.
5050	0x000013BA	ERROR_CLUSTER_NODE_DOWN	The cluster node is down.
5051	0x000013BB	ERROR_CLUSTER_NODE_UNREACHABLE	The cluster node is not reachable.
5052	0x000013BC	ERROR_CLUSTER_NODE_NOT_MEMBER	The cluster node is not a member of the cluster.
5053	0x000013BD	ERROR_CLUSTER_JOIN_NOT_IN_PROGRESS	A cluster join operation is not in progress.
5054	0x000013BE	ERROR_CLUSTER_INVALID_NETWORK	The cluster network is not valid.
5056	0x000013C0	ERROR_CLUSTER_NODE_UP	The cluster node is up.
5057	0x000013C1	ERROR_CLUSTER_IPADDR_IN_USE	The cluster IP address is already in use.
5058	0x000013C2	ERROR_CLUSTER_NODE_NOT_PAUSED	The cluster node is not paused.
5059	0x000013C3	ERROR_CLUSTER_NO_SECURITY_CONTEXT	No cluster security context is available.
5060	0x000013C4	ERROR_CLUSTER_NETWORK_NOT_INTERNAL	The cluster network is not configured for internal cluster communication.
5061	0x000013C5	ERROR_CLUSTER_NODE_ALREADY_UP	The cluster node is already up.
5062	0x000013C6	ERROR_CLUSTER_NODE_ALREADY_DOWN	The cluster node is already down.
5063	0x000013C7	ERROR_CLUSTER_NETWORK_ALREADY_ONLINE	The cluster network is already online.
5064	0x000013C8	ERROR_CLUSTER_NETWORK_ALREADY_OFFLINE	The cluster network is already offline.
5065	0x000013C9	ERROR_CLUSTER_NODE_ALREADY_MEMBER	The cluster node is already a member of the cluster.
5066	0x000013CA	ERROR_CLUSTER_LAST_INTERNAL_NETWORK	The cluster network is the only one configured for internal cluster communication between two or more active cluster nodes. The internal communication capability cannot be removed from the network.
5067	0x000013CB	ERROR_CLUSTER_NETWORK_HAS_DEPENDENTS	One or more cluster resources depend on the network to provide service to clients. The client access capability cannot be removed from the network.
5068	0x000013CC	ERROR_INVALID_OPERATION_ON_QUORUM	This operation cannot be performed on the cluster resource as it the quorum resource. You may not bring the quorum resource offline or modify its possible owners list.
5069	0x000013CD	ERROR_DEPENDENCY_NOT_ALLOWED	The cluster quorum resource is not allowed to have any dependencies.
5070	0x000013CE	ERROR_CLUSTER_NODE_PAUSED	The cluster node is paused.
5071	0x000013CF	ERROR_NODE_CANT_HOST_RESOURCE	The cluster resource cannot be brought online. The owner node cannot run this resource.
5072	0x000013D0	ERROR_CLUSTER_NODE_NOT_READY	The cluster node is not ready to perform the requested operation.
5073	0x000013D1	ERROR_CLUSTER_NODE_SHUTTING_DOWN	The cluster node is shutting down.
5074	0x000013D2	ERROR_CLUSTER_JOIN_ABORTED	The cluster join operation was aborted.
5075	0x000013D3	ERROR_CLUSTER_INCOMPATIBLE_VERSIONS	The cluster join operation failed due to incompatible software versions between the joining node and its sponsor.
5076	0x000013D4	ERROR_CLUSTER_MAXNUM_OF_RESOURCES_EXCEEDED	This resource cannot be created because the cluster has reached the limit on the number of resources it can monitor.
5077	0x000013D5	ERROR_CLUSTER_SYSTEM_CONFIG_CHANGED	The system configuration changed during the cluster join or form operation. The join or form operation was aborted.

Error			Description
decimal	Hexadecimal	Name	
5078	0x000013D6	ERROR_CLUSTER_RESOURCE_TYPE_NOT_FOUND	The specified resource type was not found.
5079	0x000013D7	ERROR_CLUSTER_RESTYPE_NOT_SUPPORTED	The specified node does not support a resource of this type. This may be due to version inconsistencies or due to the absence of the resource DLL on this node.
5080	0x000013D8	ERROR_CLUSTER_RESNAME_NOT_FOUND	The specified resource name is supported by this resource DLL. This may be due to a bad (or changed) name supplied to the resource DLL.
5081	0x000013D9	ERROR_CLUSTER_NO_RPC_PACKAGES_REGISTERED	No authentication package could be registered with the RPC server.
5082	0x000013DA	ERROR_CLUSTER_OWNER_NOT_IN_PREFLIST	You cannot bring the group online because the owner of the group is not in the preferred list for the group. To change the owner node for the group, move the group.
5083	0x000013DB	ERROR_CLUSTER_DATABASE_SEQMISMATCH	The join operation failed because the cluster database sequence number has changed or is incompatible with the locker node. This may happen during a join operation if the cluster database was changing during the join.
5084	0x000013DC	ERROR_RESMON_INVALID_STATE	The resource monitor will not allow the fail operation to be performed while the resource is in its current state. This may happen if the resource is in a pending state.
5085	0x000013DD	ERROR_CLUSTER_GUM_NOT_LOCKER	A non locker code got a request to reserve the lock for making global updates.
5086	0x000013DE	ERROR_QUORUM_DISK_NOT_FOUND	The quorum disk could not be located by the cluster service.
5087	0x000013DF	ERROR_DATABASE_BACKUP_CORRUPT	The backup up cluster database is possibly corrupt.
5088	0x000013E0	ERROR_CLUSTER_NODE_ALREADY_HAS_DFS_ROOT	A DFS root already exists in this cluster node.
5089	0x000013E1	ERROR_RESOURCE_PROPERTY_UNCHANGEABLE	An attempt to modify a resource property failed because it conflicts with another existing property.
5890	0x00001702	ERROR_CLUSTER_MEMBERSHIP_INVALID_STATE	An operation was attempted that is incompatible with the current membership state of the node.
5891	0x00001703	ERROR_CLUSTER_QUORUMLOG_NOT_FOUND	The quorum resource does not contain the quorum log.
5892	0x00001704	ERROR_CLUSTER_MEMBERSHIP_HALT	The membership engine requested shutdown of the cluster service on this node.
5893	0x00001705	ERROR_CLUSTER_INSTANCE_ID_MISMATCH	The join operation failed because the cluster instance ID of the joining node does not match the cluster instance ID of the sponsor node.
5894	0x00001706	ERROR_CLUSTER_NETWORK_NOT_FOUND_FOR_IP	A matching network for the specified IP address could not be found. Please also specify a subnet mask and a cluster network.
5895	0x00001707	ERROR_CLUSTER_PROPERTY_DATA_TYPE_MISMATCH	The actual data type of the property did not match the expected data type of the property.
5896	0x00001708	ERROR_CLUSTER_EVICT_WITHOUT_CLEANUP	The cluster node was evicted from the cluster successfully. The node was not cleaned up because it does not support the evict cleanup functionality.

Error			Description
decimal	Hexadecimal	Name	
6000	0x00001770	ERROR_ENCRYPTION_FAILED	The specified file could not be encrypted.
6001	0x00001771	ERROR_DECRYPTION_FAILED	The specified file could not be decrypted.
6002	0x00001772	ERROR_FILE_ENCRYPTED	The specified file is encrypted and the user does not have the ability to decrypt it.
6003	0x00001773	ERROR_NO_RECOVERY_POLICY	There is no valid encryption recovery policy configured for this system.
6004	0x00001774	ERROR_NO_EFS	The required encryption driver is not loaded for this system.
6005	0x00001775	ERROR_WRONG_EFS	The file was encrypted with a different encryption driver than is currently loaded.
6006	0x00001776	ERROR_NO_USER_KEYS	There are no EFS keys defined for the user.

Error			Description
decimal	Hexadecimal	Name	
6007	0x00001777	ERROR_FILE_NOT_ENCRYPTED	The specified file is not encrypted.
6008	0x00001778	ERROR_NOT_EXPORT_FORMAT	The specified file is not in the defined EFS export format.
6009	0x00001779	ERROR_FILE_READ_ONLY	The specified file is read only.
6010	0x0000177A	ERROR_DIR_EFS_DISALLOWED	The directory has been disabled for encryption.
6011	0x0000177B	ERROR_EFS_SERVER_NOT_TRUSTED	The server is not trusted for remote encryption operation.
6012	0x0000177C	ERROR_BAD_RECOVERY_POLICY	Recovery policy configured for this system contains invalid recovery certificate.
6013	0x0000177D	ERROR_EFS_ALG_BLOB_TOO_BIG	The encryption algorithm used on the source file needs a bigger key buffer than the one on the destination file.
6014	0x0000177E	ERROR_VOLUME_NOT_SUPPORT_EFS	The disk partition does not support file encryption.
6118	0x000017E6	ERROR_NO_BROWSER_SERVERS_FOUND	The list of servers for this workgroup is not currently available.
6200	0x00001838	SCHED_E_SERVICE_NOT_LOCALSYSTEM	The Task Scheduler service must be configured to run in the System account to function properly. Individual tasks may be configured to run in other accounts.
7001	0x00001B59	ERROR_CTX_WINSTATION_NAME_INVALID	The specified session name is invalid.
7002	0x00001B5A	ERROR_CTX_INVALID_PD	The specified protocol driver is invalid.
7003	0x00001B5B	ERROR_CTX_PD_NOT_FOUND	The specified protocol driver was not found in the system path.
7004	0x00001B5C	ERROR_CTX_WD_NOT_FOUND	The specified terminal connection driver was not found in the system path.
7005	0x00001B5D	ERROR_CTX_CANNOT_MAKE_EVENTLOG_ENTRY	A registry key for event logging could not be created for this session.
7006	0x00001B5E	ERROR_CTX_SERVICE_NAME_COLLISION	A service with the same name already exists on the system.
7007	0x00001B5F	ERROR_CTX_CLOSE_PENDING	A close operation is pending on the session.
7008	0x00001B60	ERROR_CTX_NO_OUTBUF	There are no free output buffers available.
7009	0x00001B61	ERROR_CTX_MODEM_INF_NOT_FOUND	The MODEM.INF file was not found.
7010	0x00001B62	ERROR_CTX_INVALID_MODEMNAME	The modem name was not found in MODEM.INF.
7011	0x00001B63	ERROR_CTX_MODEM_RESPONSE_ERROR	The modem did not accept the command sent to it. Verify that the configured modem name matches the attached modem.
7012	0x00001B64	ERROR_CTX_MODEM_RESPONSE_TIMEOUT	The modem did not respond to the command sent to it. Verify that the modem is properly cabled and powered on.
7013	0x00001B65	ERROR_CTX_MODEM_RESPONSE_NO_CARRIER	Carrier detect has failed or carrier has been dropped due to disconnect.
7014	0x00001B66	ERROR_CTX_MODEM_RESPONSE_NO_DIALTONE	Dial tone not detected within the required time. Verify that the phone cable is properly attached and functional.
7015	0x00001B67	ERROR_CTX_MODEM_RESPONSE_BUSY	Busy signal detected at remote site on callback.
7016	0x00001B68	ERROR_CTX_MODEM_RESPONSE_VOICE	Voice detected at remote site on callback.
7017	0x00001B69	ERROR_CTX_TD_ERROR	Transport driver error
7022	0x00001B6E	ERROR_CTX_WINSTATION_NOT_FOUND	The specified session cannot be found.
7023	0x00001B6F	ERROR_CTX_WINSTATION_ALREADY_EXISTS	The specified session name is already in use.
7024	0x00001B70	ERROR_CTX_WINSTATION_BUSY	The requested operation cannot be completed because the terminal connection is currently busy processing a connect, disconnect, reset, or delete operation.
7025	0x00001B71	ERROR_CTX_BAD_VIDEO_MODE	An attempt has been made to connect to a session whose video mode is not supported by the current client.
7035	0x00001B7B	ERROR_CTX_GRAPHICS_INVALID	The application attempted to enable DOS graphics mode. DOS graphics mode is not supported.
7037	0x00001B7D	ERROR_CTX_LOGON_DISABLED	Your interactive logon privilege has been disabled. Please contact your administrator.
7038	0x00001B7E	ERROR_CTX_NOT_CONSOLE	The requested operation can be performed only on the system console. This is most often the result of a driver or system DLL requiring direct console access.

Error			Description
decimal	Hexadecimal	Name	
7040	0x00001B80	ERROR_CTX_CLIENT_QUERY_TIMEOUT	The client failed to respond to the server connect message.
7041	0x00001B81	ERROR_CTX_CONSOLE_DISCONNECT	Disconnecting the console session is not supported.
7042	0x00001B82	ERROR_CTX_CONSOLE_CONNECT	Reconnecting a disconnected session to the console is not supported.
7044	0x00001B84	ERROR_CTX_SHADOW_DENIED	The request to control another session remotely was denied.
7045	0x00001B85	ERROR_CTX_WINSTATION_ACCESS_DENIED	The requested session access is denied.
7049	0x00001B89	ERROR_CTX_INVALID_WD	The specified terminal connection driver is invalid.
7050	0x00001B8A	ERROR_CTX_SHADOW_INVALID	The requested session cannot be controlled remotely. This may be because the session is disconnected or does not currently have a user logged on.
7051	0x00001B8B	ERROR_CTX_SHADOW_DISABLED	The requested session is not configured to allow remote control.
7052	0x00001B8C	ERROR_CTX_CLIENT_LICENSE_IN_USE	Your request to connect to this Terminal Server has been rejected. Your Terminal Server client license number is currently being used by another user. Please call your system administrator to obtain a unique license number.
7053	0x00001B8D	ERROR_CTX_CLIENT_LICENSE_NOT_SET	Your request to connect to this Terminal Server has been rejected. Your Terminal Server client license number has not been entered for this copy of the Terminal Server client. Please contact your system administrator.
7054	0x00001B8E	ERROR_CTX_LICENSE_NOT_AVAILABLE	The system has reached its licensed logon limit. Please try again later.
7055	0x00001B8F	ERROR_CTX_LICENSE_CLIENT_INVALID	The client you are using is not licensed to use this system. Your logon request is denied.
7056	0x00001B90	ERROR_CTX_LICENSE_EXPIRED	The system license has expired. Your logon request is denied.
7057	0x00001B91	ERROR_CTX_SHADOW_NOT_RUNNING	Remote control could not be terminated because the specified session is not currently being remotely controlled.

Error			Description
decimal	Hexadecimal	Name	
8001	0x00001F41	FRS_ERR_INVALID_API_SEQUENCE	The file replication service API was called incorrectly.
8002	0x00001F42	FRS_ERR_STARTING_SERVICE	The file replication service cannot be started.
8003	0x00001F43	FRS_ERR_STOPPING_SERVICE	The file replication service cannot be stopped.
8004	0x00001F44	FRS_ERR_INTERNAL_API	The file replication service API terminated the request. The event log may have more information.
8005	0x00001F45	FRS_ERR_INTERNAL	The file replication service terminated the request. The event log may have more information.
8006	0x00001F46	FRS_ERR_SERVICE_COMM	The file replication service cannot be contacted. The event log may have more information.
8007	0x00001F47	FRS_ERR_INSUFFICIENT_PRIV	The file replication service cannot satisfy the request because the user has insufficient privileges. The event log may have more information.
8008	0x00001F48	FRS_ERR_AUTHENTICATION	The file replication service cannot satisfy the request because authenticated RPC is not available. The event log may have more information.
8009	0x00001F49	FRS_ERR_PARENT_INSUFFICIENT_PRIV	The file replication service cannot satisfy the request because the user has insufficient privileges on the domain controller. The event log may have more information.
8010	0x00001F4A	FRS_ERR_PARENT_AUTHENTICATION	The file replication service cannot satisfy the request because authenticated RPC is not available on the domain controller. The event log may have more information.

Error			Description
decimal	Hexadecimal	Name	
8011	0x00001F4B	FRS_ERR_CHILD_TO_PARENT_COMM	The file replication service cannot communicate with the file replication service on the domain controller. The event log may have more information.
8012	0x00001F4C	FRS_ERR_PARENT_TO_CHILD_COMM	The file replication service on the domain controller cannot communicate with the file replication service on this computer. The event log may have more information.
8013	0x00001F4D	FRS_ERR_SYSVOL_POPULATE	The file replication service cannot populate the system volume because of an internal error. The event log may have more information.
8014	0x00001F4E	FRS_ERR_SYSVOL_POPULATE_TIMEOUT	The file replication service cannot populate the system volume because of an internal timeout. The event log may have more information.
8015	0x00001F4F	FRS_ERR_SYSVOL_IS_BUSY	The file replication service cannot process the request. The system volume is busy with a previous request.
8016	0x00001F50	FRS_ERR_SYSVOL_DEMOTE	The file replication service cannot stop replicating the system volume because of an internal error. The event log may have more information.
8017	0x00001F51	FRS_ERR_INVALID_SERVICE_PARAMETER	The file replication service detected an invalid parameter.
8200	0x00002008	ERROR_DS_NOT_INSTALLED	An error occurred while installing the directory service. For more information, see the event log.
8201	0x00002009	ERROR_DS_MEMBERSHIP_EVALUATED_LOCALLY	The directory service evaluated group memberships locally.
8202	0x0000200A	ERROR_DS_NO_ATTRIBUTE_OR_VALUE	The specified directory service attribute or value does not exist.
8203	0x0000200B	ERROR_DS_INVALID_ATTRIBUTE_SYNTAX	The attribute syntax specified to the directory service is invalid.
8204	0x0000200C	ERROR_DS_ATTRIBUTE_TYPE_UNDEFINED	The attribute type specified to the directory service is not defined.
8205	0x0000200D	ERROR_DS_ATTRIBUTE_OR_VALUE_EXISTS	The specified directory service attribute or value already exists.
8206	0x0000200E	ERROR_DS_BUSY	The directory service is busy.
8207	0x0000200F	ERROR_DS_UNAVAILABLE	The directory service is unavailable.
8208	0x00002010	ERROR_DS_NO_RIDS_ALLOCATED	The directory service was unable to allocate a relative identifier.
8209	0x00002011	ERROR_DS_NO_MORE_RIDS	The directory service has exhausted the pool of relative identifiers.
8210	0x00002012	ERROR_DS_INCORRECT_ROLE_OWNER	The requested operation could not be performed because the directory service is not the master for that type of operation.
8211	0x00002013	ERROR_DS_RIDMGR_INIT_ERROR	The directory service was unable to initialize the subsystem that allocates relative identifiers.
8212	0x00002014	ERROR_DS_OBJ_CLASS_VIOLATION	The requested operation did not satisfy one or more constraints associated with the class of the object.
8213	0x00002015	ERROR_DS_CANT_ON_NON_LEAF	The directory service can perform the requested operation only on a leaf object.
8214	0x00002016	ERROR_DS_CANT_ON_RDN	The directory service cannot perform the requested operation on the RDN attribute of an object.
8215	0x00002017	ERROR_DS_CANT_MOD_OBJ_CLASS	The directory service detected an attempt to modify the object class of an object.
8216	0x00002018	ERROR_DS_CROSS_DOM_MOVE_ERROR	The requested cross-domain move operation could not be performed.
8217	0x00002019	ERROR_DS_GC_NOT_AVAILABLE	Unable to contact the global catalog server.
8218	0x0000201A	ERROR_SHARED_POLICY	The policy object is shared and can only be modified at the root.
8219	0x0000201B	ERROR_POLICY_OBJECT_NOT_FOUND	The policy object does not exist.
8220	0x0000201C	ERROR_POLICY_ONLY_IN_DS	The requested policy information is only in the directory service.
8221	0x0000201D	ERROR_PROMOTION_ACTIVE	A domain controller promotion is currently active.
8222	0x0000201E	ERROR_NO_PROMOTION_ACTIVE	A domain controller promotion is not currently active

Error			Description
decimal	Hexadecimal	Name	
8224	0x00002020	ERROR_DS_OPERATIONS_ERROR	An operations error occurred.
8225	0x00002021	ERROR_DS_PROTOCOL_ERROR	A protocol error occurred.
8226	0x00002022	ERROR_DS_TIMELIMIT_EXCEEDED	The time limit for this request was exceeded.
8227	0x00002023	ERROR_DS_SIZELIMIT_EXCEEDED	The size limit for this request was exceeded.
8228	0x00002024	ERROR_DS_ADMIN_LIMIT_EXCEEDED	The administrative limit for this request was exceeded.
8229	0x00002025	ERROR_DS_COMPARE_FALSE	The compare response was false.
8230	0x00002026	ERROR_DS_COMPARE_TRUE	The compare response was true.
8231	0x00002027	ERROR_DS_AUTH_METHOD_NOT_SUPPORTED	The requested authentication method is not supported by the server.
8232	0x00002028	ERROR_DS_STRONG_AUTH_REQUIRED	A more secure authentication method is required for this server.
8233	0x00002029	ERROR_DS_INAPPROPRIATE_AUTH	Inappropriate authentication.
8234	0x0000202A	ERROR_DS_AUTH_UNKNOWN	The authentication mechanism is unknown.
8235	0x0000202B	ERROR_DS_REFERRAL	A referral was returned from the server.
8236	0x0000202C	ERROR_DS_UNAVAILABLE_CRIT_EXTENSION	The server does not support the requested critical extension.
8237	0x0000202D	ERROR_DS_CONFIDENTIALITY_REQUIRED	This request requires a secure connection.
8238	0x0000202E	ERROR_DS_INAPPROPRIATE_MATCHING	Inappropriate matching.
8239	0x0000202F	ERROR_DS_CONSTRAINT_VIOLATION	A constraint violation occurred.
8240	0x00002030	ERROR_DS_NO_SUCH_OBJECT	There is no such object on the server.
8241	0x00002031	ERROR_DS_ALIAS_PROBLEM	There is an alias problem.
8242	0x00002032	ERROR_DS_INVALID_DN_SYNTAX	An invalid dn syntax has been specified.
8243	0x00002033	ERROR_DS_IS_LEAF	The object is a leaf object.
8244	0x00002034	ERROR_DS_ALIAS_DEREF_PROBLEM	There is an alias dereferencing problem.
8245	0x00002035	ERROR_DS_UNWILLING_TO_PERFORM	The server is unwilling to process the request.
8246	0x00002036	ERROR_DS_LOOP_DETECT	A loop has been detected.
8247	0x00002037	ERROR_DS_NAMING_VIOLATION	There is a naming violation.
8248	0x00002038	ERROR_DS_OBJECT_RESULTS_TOO_LARGE	The result set is too large.
8249	0x00002039	ERROR_DS_AFFECTS_MULTIPLE_DSAS	The operation affects multiple DSAs
8250	0x0000203A	ERROR_DS_SERVER_DOWN	The server is not operational.
8251	0x0000203B	ERROR_DS_LOCAL_ERROR	A local error has occurred.
8252	0x0000203C	ERROR_DS_ENCODING_ERROR	An encoding error has occurred.
8253	0x0000203D	ERROR_DS_DECODING_ERROR	A decoding error has occurred.
8254	0x0000203E	ERROR_DS_FILTER_UNKNOWN	The search filter cannot be recognized.
8255	0x0000203F	ERROR_DS_PARAM_ERROR	One or more parameters are illegal.
8256	0x00002040	ERROR_DS_NOT_SUPPORTED	The specified method is not supported.
8257	0x00002041	ERROR_DS_NO_RESULTS_RETURNED	No results were returned.
8258	0x00002042	ERROR_DS_CONTROL_NOT_FOUND	The specified control is not supported by the server.
8259	0x00002043	ERROR_DS_CLIENT_LOOP	A referral loop was detected by the client.
8260	0x00002044	ERROR_DS_REFERRAL_LIMIT_EXCEEDED	The preset referral limit was exceeded.
8261	0x00002045	ERROR_DS_SORT_CONTROL_MISSING	The search requires a SORT control.
8262	0x00002046	ERROR_DS_OFFSET_RANGE_ERROR	The search results exceed the offset range specified.
8301	0x0000206D	ERROR_DS_ROOT_MUST_BE_NC	The root object must be the head of a naming context. The root object cannot have an instantiated parent.
8302	0x0000206E	ERROR_DS_ADD_REPLICA_INHIBITED	The add replica operation cannot be performed. The naming context must be writable in order to create the replica.
8303	0x0000206F	ERROR_DS_ATT_NOT_DEF_IN_SCHEMA	A reference to an attribute that is not defined in the schema occurred.
8304	0x00002070	ERROR_DS_MAX_OBJ_SIZE_EXCEEDED	The maximum size of an object has been exceeded.
8305	0x00002071	ERROR_DS_OBJ_STRING_NAME_EXISTS	An attempt was made to add an object to the directory with a name that is already in use.
8306	0x00002072	ERROR_DS_NO_RDN_DEFINED_IN_SCHEMA	An attempt was made to add an object of a class that does not have an RDN defined in the schema.

Error			Description
decimal	Hexadecimal	Name	
8307	0x00002073	ERROR_DS_RDN_DOESNT_MATCH_SCHEMA	An attempt was made to add an object using an RDN that is not the RDN defined in the schema.
8308	0x00002074	ERROR_DS_NO_REQUESTED_ATTS_FOUND	None of the requested attributes were found on the objects.
8309	0x00002075	ERROR_DS_USER_BUFFER_TO_SMALL	The user buffer is too small.
8310	0x00002076	ERROR_DS_ATT_IS_NOT_ON_OBJ	The attribute specified in the operation is not present on the object.
8311	0x00002077	ERROR_DS_ILLEGAL_MOD_OPERATION	Illegal modify operation. Some aspect of the modification is not permitted.
8312	0x00002078	ERROR_DS_OBJ_TOO_LARGE	The specified object is too large.
8313	0x00002079	ERROR_DS_BAD_INSTANCE_TYPE	The specified instance type is not valid.
8314	0x0000207A	ERROR_DS_MASTERDSA_REQUIRED	The operation must be performed at a master DSA.
8315	0x0000207B	ERROR_DS_OBJECT_CLASS_REQUIRED	The object class attribute must be specified.
8316	0x0000207C	ERROR_DS_MISSING_REQUIRED_ATT	A required attribute is missing.
8317	0x0000207D	ERROR_DS_ATT_NOT_DEF_FOR_CLASS	An attempt was made to modify an object to include an attribute that is not legal for its class
8318	0x0000207E	ERROR_DS_ATT_ALREADY_EXISTS	The specified attribute is already present on the object.
8320	0x00002080	ERROR_DS_CANT_ADD_ATT_VALUES	The specified attribute is not present, or has no values.
8321	0x00002081	ERROR_DS_SINGLE_VALUE_CONSTRAINT	Multiple values were specified for an attribute that can have only one value.
8322	0x00002082	ERROR_DS_RANGE_CONSTRAINT	A value for the attribute was not in the acceptable range of values.
8323	0x00002083	ERROR_DS_ATT_VAL_ALREADY_EXISTS	The specified value already exists.
8324	0x00002084	ERROR_DS_CANT_REM_MISSING_ATT	The attribute cannot be removed because it is not present on the object.
8325	0x00002085	ERROR_DS_CANT_REM_MISSING_ATT_VAL	The attribute value cannot be removed because it is not present on the object.
8326	0x00002086	ERROR_DS_ROOT_CANT_BE_SUBREF	The specified root object cannot be a subref.
8327	0x00002087	ERROR_DS_NO_CHAINING	Chaining is not permitted.
8328	0x00002088	ERROR_DS_NO_CHAINED_EVAL	Chained evaluation is not permitted.
8329	0x00002089	ERROR_DS_NO_PARENT_OBJECT	The operation could not be performed because the object's parent is either uninstantiated or deleted.
8330	0x0000208A	ERROR_DS_PARENT_IS_AN_ALIAS	Having a parent that is an alias is not permitted. Aliases are leaf objects.
8331	0x0000208B	ERROR_DS_CANT_MIX_MASTER_AND_REPS	The object and parent must be of the same type, either both masters or both replicas.
8332	0x0000208C	ERROR_DS_CHILDREN_EXIST	The operation cannot be performed because child objects exist. This operation can only be performed on a leaf object.
8333	0x0000208D	ERROR_DS_OBJ_NOT_FOUND	Directory object not found.
8334	0x0000208E	ERROR_DS_ALIASED_OBJ_MISSING	The aliased object is missing.
8335	0x0000208F	ERROR_DS_BAD_NAME_SYNTAX	The object name has bad syntax.
8336	0x00002090	ERROR_DS_ALIAS_POINTS_TO_ALIAS	It is not permitted for an alias to refer to another alias.
8337	0x00002091	ERROR_DS_CANT_DEREF_ALIAS	The alias cannot be dereferenced.
8338	0x00002092	ERROR_DS_OUT_OF_SCOPE	The operation is out of scope.
8339	0x00002093	ERROR_DS_OBJECT_BEING_REMOVED	The operation cannot continue because the object is in the process of being removed.
8340	0x00002094	ERROR_DS_CANT_DELETE_DSA_OBJ	The DSA object cannot be deleted.
8341	0x00002095	ERROR_DS_GENERIC_ERROR	A directory service error has occurred.
8342	0x00002096	ERROR_DS_DSA_MUST_BE_INT_MASTER	The operation can only be performed on an internal master DSA object.
8343	0x00002097	ERROR_DS_CLASS_NOT_DSA	The object must be of class DSA.
8344	0x00002098	ERROR_DS_INSUFF_ACCESS_RIGHTS	Insufficient access rights to perform the operation.
8345	0x00002099	ERROR_DS_ILLEGAL_SUPERIOR	The object cannot be added because the parent is not on the list of possible superiors.
8346	0x0000209A	ERROR_DS_ATTRIBUTE_OWNED_BY_SAM	Access to the attribute is not permitted because the attribute is owned by the Security Accounts Manager (SAM).

Error			Description
decimal	Hexadecimal	Name	
8347	0x0000209B	ERROR_DS_NAME_TOO_MANY_PARTS	The name has too many parts.
8348	0x0000209C	ERROR_DS_NAME_TOO_LONG	The name is too long.
8349	0x0000209D	ERROR_DS_NAME_VALUE_TOO_LONG	The name value is too long.
8350	0x0000209E	ERROR_DS_NAME_UNPARSEABLE	The directory service encountered an error parsing a name.
8351	0x0000209F	ERROR_DS_NAME_TYPE_UNKNOWN	The directory service cannot get the attribute type for a name.
8352	0x000020A0	ERROR_DS_NOT_AN_OBJECT	The name does not identify an object; the name identifies a phantom.
8353	0x000020A1	ERROR_DS_SEC_DESC_TOO_SHORT	The security descriptor is too short.
8354	0x000020A2	ERROR_DS_SEC_DESC_INVALID	The security descriptor is invalid.
8355	0x000020A3	ERROR_DS_NO_DELETED_NAME	Failed to create name for deleted object.
8356	0x000020A4	ERROR_DS_SUBREF_MUST_HAVE_PARENT	The parent of a new subref must exist.
8357	0x000020A5	ERROR_DS_NCNAME_MUST_BE_NC	The object must be a naming context.
8358	0x000020A6	ERROR_DS_CANT_ADD_SYSTEM_ONLY	It is not permitted to add an attribute which is owned by the system.
8359	0x000020A7	ERROR_DS_CLASS_MUST_BE_CONCRETE	The class of the object must be structural; you cannot instantiate an abstract class.
8360	0x000020A8	ERROR_DS_INVALID_DMD	The schema object could not be found.
8361	0x000020A9	ERROR_DS_OBJ_GUID_EXISTS	A local object with this GUID (dead or alive) already exists.
8362	0x000020AA	ERROR_DS_NOT_ON_BACKLINK	The operation cannot be performed on a back link.
8363	0x000020AB	ERROR_DS_NO_CROSSREF_FOR_NC	The cross reference for the specified naming context could not be found.
8364	0x000020AC	ERROR_DS_SHUTTING_DOWN	The operation could not be performed because the directory service is shutting down.
8365	0x000020AD	ERROR_DS_UNKNOWN_OPERATION	The directory service request is invalid.
8366	0x000020AE	ERROR_DS_INVALID_ROLE_OWNER	The role owner attribute could not be read.
8367	0x000020AF	ERROR_DS_COULDNT_CONTACT_FSMO	The requested FSMO operation failed. The current FSMO holder could not be reached.
8368	0x000020B0	ERROR_DS_CROSS_NC_DN_RENAME	Modification of a DN across a naming context is not permitted.
8369	0x000020B1	ERROR_DS_CANT_MOD_SYSTEM_ONLY	The attribute cannot be modified because it is owned by the system.
8370	0x000020B2	ERROR_DS_REPLICATOR_ONLY	Only the replicator can perform this function.
8371	0x000020B3	ERROR_DS_OBJ_CLASS_NOT_DEFINED	The specified class is not defined.
8372	0x000020B4	ERROR_DS_OBJ_CLASS_NOT_SUBCLASS	The specified class is not a subclass.
8373	0x000020B5	ERROR_DS_NAME_REFERENCE_INVALID	The name reference is invalid.
8374	0x000020B6	ERROR_DS_CROSS_REF_EXISTS	A cross reference already exists.
8375	0x000020B7	ERROR_DS_CANT_DEL_MASTER_CROSSREF	It is not permitted to delete a master cross reference.
8376	0x000020B8	ERROR_DS_SUBTREE_NOTIFY_NOT_NC_HEAD	Subtree notifications are only supported on NC heads.
8377	0x000020B9	ERROR_DS_NOTIFY_FILTER_TOO_COMPLEX	Notification filter is too complex.
8378	0x000020BA	ERROR_DS_DUP_RDN	Schema update failed: duplicate RDN.
8379	0x000020BB	ERROR_DS_DUP_OID	Schema update failed: duplicate OID
8380	0x000020BC	ERROR_DS_DUP_MAPI_ID	Schema update failed: duplicate MAPI identifier.
8381	0x000020BD	ERROR_DS_DUP_SCHEMA_ID_GUID	Schema update failed: duplicate schema-id GUID.
8382	0x000020BE	ERROR_DS_DUP_LDAP_DISPLAY_NAME	Schema update failed: duplicate LDAP display name.
8383	0x000020BF	ERROR_DS_SEMANTIC_ATT_TEST	Schema update failed: range-lower less than range upper
8384	0x000020C0	ERROR_DS_SYNTAX_MISMATCH	Schema update failed: syntax mismatch
8385	0x000020C1	ERROR_DS_EXISTS_IN_MUST_HAVE	Schema deletion failed: attribute is used in must-contain
8386	0x000020C2	ERROR_DS_EXISTS_IN_MAY_HAVE	Schema deletion failed: attribute is used in may-contain
8387	0x000020C3	ERROR_DS_NONEXISTENT_MAY_HAVE	Schema update failed: attribute in may-contain does not exist
8388	0x000020C4	ERROR_DS_NONEXISTENT_MUST_HAVE	Schema update failed: attribute in must-contain does not exist

Error			Description
decimal	Hexadecimal	Name	
8389	0x000020C5	ERROR_DS_AUX_CLS_TEST_FAIL	Schema update failed: class in aux-class list does not exist or is not an auxiliary class
8390	0x000020C6	ERROR_DS_NONEXISTENT_POSS_SUP	Schema update failed: class in poss-superiors does not exist
8391	0x000020C7	ERROR_DS_SUB_CLS_TEST_FAIL	Schema update failed: class in subclassof list does not exist or does not satisfy hierarchy rules
8392	0x000020C8	ERROR_DS_BAD_RDN_ATT_ID_SYNTAX	Schema update failed: Rdn-Att-Id has wrong syntax
8393	0x000020C9	ERROR_DS_EXISTS_IN_AUX_CLS	Schema deletion failed: class is used as auxiliary class
8394	0x000020CA	ERROR_DS_EXISTS_IN_SUB_CLS	Schema deletion failed: class is used as sub class
8395	0x000020CB	ERROR_DS_EXISTS_IN_POSS_SUP	Schema deletion failed: class is used as poss superior
8396	0x000020CC	ERROR_DS_RECALCSCHEMA_FAILED	Schema update failed in recalculating validation cache.
8397	0x000020CD	ERROR_DS_TREE_DELETE_NOT_FINISHED	The tree deletion is not finished.
8398	0x000020CE	ERROR_DS_CANT_DELETE	The requested delete operation could not be performed.
8399	0x000020CF	ERROR_DS_ATT_SCHEMA_REQ_ID	Cannot read the governs class identifier for the schema record.
8400	0x000020D0	ERROR_DS_BAD_ATT_SCHEMA_SYNTAX	The attribute schema has bad syntax.
8401	0x000020D1	ERROR_DS_CANT_CACHE_ATT	The attribute could not be cached.
8402	0x000020D2	ERROR_DS_CANT_CACHE_CLASS	The class could not be cached.
8403	0x000020D3	ERROR_DS_CANT_REMOVE_ATT_CACHE	The attribute could not be removed from the cache.
8404	0x000020D4	ERROR_DS_CANT_REMOVE_CLASS_CACHE	The class could not be removed from the cache.
8405	0x000020D5	ERROR_DS_CANT_RETRIEVE_DN	The distinguished name attribute could not be read.
8406	0x000020D6	ERROR_DS_MISSING_SUPREF	A required subref is missing.
8407	0x000020D7	ERROR_DS_CANT_RETRIEVE_INSTANCE	The instance type attribute could not be retrieved.
8408	0x000020D8	ERROR_DS_CODE_INCONSISTENCY	An internal error has occurred.
8409	0x000020D9	ERROR_DS_DATABASE_ERROR	A database error has occurred.
8410	0x000020DA	ERROR_DS_GOVERNSID_MISSING	The attribute GOVERNSID is missing.
8411	0x000020DB	ERROR_DS_MISSING_EXPECTED_ATT	An expected attribute is missing.
8412	0x000020DC	ERROR_DS_NCNAME_MISSING_CR_REF	The specified naming context is missing a cross reference.
8413	0x000020DD	ERROR_DS_SECURITY_CHECKING_ERROR	A security checking error has occurred.
8414	0x000020DE	ERROR_DS_SCHEMA_NOT_LOADED	The schema is not loaded.
8415	0x000020DF	ERROR_DS_SCHEMA_ALLOC_FAILED	Schema allocation failed. Please check if the machine is running low on memory.
8416	0x000020E0	ERROR_DS_ATT_SCHEMA_REQ_SYNTAX	Failed to obtain the required syntax for the attribute schema.
8417	0x000020E1	ERROR_DS_GCVERIFY_ERROR	The global catalog verification failed. The global catalog is not available or does not support the operation. Some part of the directory is currently not available.
8418	0x000020E2	ERROR_DS_DRA_SCHEMA_MISMATCH	The replication operation failed because of a schema mismatch between the servers involved.
8419	0x000020E3	ERROR_DS_CANT_FIND_DSA_OBJ	The DSA object could not be found.
8420	0x000020E4	ERROR_DS_CANT_FIND_EXPECTED_NC	The naming context could not be found.
8421	0x000020E5	ERROR_DS_CANT_FIND_NC_IN_CACHE	The naming context could not be found in the cache.
8422	0x000020E6	ERROR_DS_CANT_RETRIEVE_CHILD	The child object could not be retrieved.
8423	0x000020E7	ERROR_DS_SECURITY_ILLEGAL_MODIFY	The modification was not permitted for security reasons.
8424	0x000020E8	ERROR_DS_CANT_REPLACE_HIDDEN_REC	The operation cannot replace the hidden record.
8425	0x000020E9	ERROR_DS_BAD_HIERARCHY_FILE	The hierarchy file is invalid.
8426	0x000020EA	ERROR_DS_BUILD_HIERARCHY_TABLE_FAILED	The attempt to build the hierarchy table failed.
8427	0x000020EB	ERROR_DS_CONFIG_PARAM_MISSING	The directory configuration parameter is missing from the registry.
8428	0x000020EC	ERROR_DS_COUNTING_AB_INDICES_FAILED	The attempt to count the address book indices failed.

Error			Description
decimal	Hexadecimal	Name	
8429	0x000020ED	ERROR_DS_HIERARCHY_TABLE_MALLOC_FAILED	The allocation of the hierarchy table failed.
8430	0x000020EE	ERROR_DS_INTERNAL_FAILURE	The directory service encountered an internal failure.
8431	0x000020EF	ERROR_DS_UNKNOWN_ERROR	The directory service encountered an unknown failure.
8432	0x000020F0	ERROR_DS_ROOT_REQUIRES_CLASS_TOP	A root object requires a class of 'top'.
8433	0x000020F1	ERROR_DS_REFUSING_FSMO_ROLES	This directory server is shutting down, and cannot take ownership of new floating single-master operation roles.
8434	0x000020F2	ERROR_DS_MISSING_FSMO_SETTINGS	The directory service is missing mandatory configuration information, and is unable to determine the ownership of floating single-master operation roles.
8435	0x000020F3	ERROR_DS_UNABLE_TO_SURRENDER_ROLES	The directory service was unable to transfer ownership of one or more floating single-master operation roles to other servers.
8436	0x000020F4	ERROR_DS_DRA_GENERIC	The replication operation failed.
8437	0x000020F5	ERROR_DS_DRA_INVALID_PARAMETER	An invalid parameter was specified for this replication operation.
8438	0x000020F6	ERROR_DS_DRA_BUSY	The directory service is too busy to complete the replication operation at this time.
8439	0x000020F7	ERROR_DS_DRA_BAD_DN	The distinguished name specified for this replication operation is invalid.
8440	0x000020F8	ERROR_DS_DRA_BAD_NC	The naming context specified for this replication operation is invalid.
8441	0x000020F9	ERROR_DS_DRA_DN_EXISTS	The distinguished name specified for this replication operation already exists.
8442	0x000020FA	ERROR_DS_DRA_INTERNAL_ERROR	The replication system encountered an internal error.
8443	0x000020FB	ERROR_DS_DRA_INCONSISTENT_DIT	The replication operation encountered a database inconsistency.
8444	0x000020FC	ERROR_DS_DRA_CONNECTION_FAILED	The server specified for this replication operation could not be contacted.
8445	0x000020FD	ERROR_DS_DRA_BAD_INSTANCE_TYPE	The replication operation encountered an object with an invalid instance type.
8446	0x000020FE	ERROR_DS_DRA_OUT_OF_MEM	The replication operation failed to allocate memory.
8447	0x000020FF	ERROR_DS_DRA_MAIL_PROBLEM	The replication operation encountered an error with the mail system.
8448	0x00002100	ERROR_DS_DRA_REF_ALREADY_EXISTS	The replication reference information for the target server already exists.
8449	0x00002101	ERROR_DS_DRA_REF_NOT_FOUND	The replication reference information for the target server does not exist.
8450	0x00002102	ERROR_DS_DRA_OBJ_IS_REP_SOURCE	The naming context cannot be removed because it is replicated to another server.
8451	0x00002103	ERROR_DS_DRA_DB_ERROR	The replication operation encountered a database error.
8452	0x00002104	ERROR_DS_DRA_NO_REPLICA	The naming context is in the process of being removed or is not replicated from the specified server.
8453	0x00002105	ERROR_DS_DRA_ACCESS_DENIED	Replication access was denied.
8454	0x00002106	ERROR_DS_DRA_NOT_SUPPORTED	The requested operation is not supported by this version of the directory service.
8455	0x00002107	ERROR_DS_DRA_RPC_CANCELLED	The replication remote procedure call was cancelled.
8456	0x00002108	ERROR_DS_DRA_SOURCE_DISABLED	The source server is currently rejecting replication requests.
8457	0x00002109	ERROR_DS_DRA_SINK_DISABLED	The destination server is currently rejecting replication requests.
8458	0x0000210A	ERROR_DS_DRA_NAME_COLLISION	The replication operation failed due to a collision of object names.
8459	0x0000210B	ERROR_DS_DRA_SOURCE_REINSTALLED	The replication source has been reinstalled.
8460	0x0000210C	ERROR_DS_DRA_MISSING_PARENT	The replication operation failed because a required parent object is missing.

Error			Description
decimal	Hexadecimal	Name	
8461	0x0000210D	ERROR_DS_DRA_PREEMPTED	The replication operation was preempted.
8462	0x0000210E	ERROR_DS_DRA_ABANDON_SYNC	The replication synchronization attempt was abandoned because of a lack of updates.
8463	0x0000210F	ERROR_DS_DRA_SHUTDOWN	The replication operation was terminated because the system is shutting down.
8464	0x00002110	ERROR_DS_DRA_INCOMPATIBLE_PARTIAL_SET	The replication synchronization attempt failed as the destination partial attribute set is not a subset of source partial attribute set.
8465	0x00002111	ERROR_DS_DRA_SOURCE_IS_PARTIAL_REPLICA	The replication synchronization attempt failed because a master replica attempted to sync from a partial replica.
8466	0x00002112	ERROR_DS_DRA_EXTN_CONNECTION_FAILED	The server specified for this replication operation was contacted, but that server was unable to contact an additional server needed to complete the operation.
8467	0x00002113	ERROR_DS_INSTALL_SCHEMA_MISMATCH	The version of the Active Directory schema of the source forest is not compatible with the version of Active Directory on this computer. You must upgrade the operating system on a domain controller in the source forest before this computer can be added as a domain controller to that forest.
8468	0x00002114	ERROR_DS_DUP_LINK_ID	Schema update failed: An attribute with the same link identifier already exists.
8469	0x00002115	ERROR_DS_NAME_ERROR_RESOLVING	Name translation: Generic processing error.
8470	0x00002116	ERROR_DS_NAME_ERROR_NOT_FOUND	Name translation: Could not find the name or insufficient right to see name.
8471	0x00002117	ERROR_DS_NAME_ERROR_NOT_UNIQUE	Name translation: Input name mapped to more than one output name.
8472	0x00002118	ERROR_DS_NAME_ERROR_NO_MAPPING	Name translation: Input name found, but not the associated output format.
8473	0x00002119	ERROR_DS_NAME_ERROR_DOMAIN_ONLY	Name translation: Unable to resolve completely, only the domain was found.
8474	0x0000211A	ERROR_DS_NAME_ERROR_NO_SYNTACTICAL_MAPPING	Name translation: Unable to perform purely syntactical mapping at the client without going out to the wire.
8475	0x0000211B	ERROR_DS_CONSTRUCTED_ATT_MOD	Modification of a constructed att is not allowed.
8476	0x0000211C	ERROR_DS_WRONG_OM_OBJ_CLASS	The OM-Object-Class specified is incorrect for an attribute with the specified syntax.
8477	0x0000211D	ERROR_DS_DRA_REPL_PENDING	The replication request has been posted; waiting for reply.
8478	0x0000211E	ERROR_DS_DS_REQUIRED	The requested operation requires a directory service, and none was available.
8479	0x0000211F	ERROR_DS_INVALID_LDAP_DISPLAY_NAME	The LDAP display name of the class or attribute contains non-ASCII characters.
8480	0x00002120	ERROR_DS_NON_BASE_SEARCH	The requested search operation is only supported for base searches.
8481	0x00002121	ERROR_DS_CANT_RETRIEVE_ATTS	The search failed to retrieve attributes from the database.
8482	0x00002122	ERROR_DS_BACKLINK_WITHOUT_LINK	The schema update operation tried to add a backward link attribute that has no corresponding forward link.
8483	0x00002123	ERROR_DS_EPOCH_MISMATCH	Source and destination of a cross domain move do not agree on the object's epoch number. Either source or destination does not have the latest version of the object.
8484	0x00002124	ERROR_DS_SRC_NAME_MISMATCH	Source and destination of a cross domain move do not agree on the object's current name. Either source or destination does not have the latest version of the object.
8485	0x00002125	ERROR_DS_SRC_AND_DST_NC_IDENTICAL	Source and destination of a cross domain move operation are identical. Caller should use local move operation instead of cross domain move operation.

Error			Description
decimal	Hexadecimal	Name	
8486	0x00002126	ERROR_DS_DST_NC_MISMATCH	Source and destination for a cross domain move are not in agreement on the naming contexts in the forest. Either source or destination does not have the latest version of the Partitions container.
8487	0x00002127	ERROR_DS_NOT_AUTHORITIVE_FOR_DST_NC	Destination of a cross domain move is not authoritative for the destination naming context.
8488	0x00002128	ERROR_DS_SRC_GUID_MISMATCH	Source and destination of a cross domain move do not agree on the identity of the source object. Either source or destination does not have the latest version of the source object.
8489	0x00002129	ERROR_DS_CANT_MOVE_DELETED_OBJECT	Object being moved across domains is already known to be deleted by the destination server. The source server does not have the latest version of the source object.
8490	0x0000212A	ERROR_DS_PDC_OPERATION_IN_PROGRESS	Another operation which requires exclusive access to the PDC PSMO is already in progress.
8491	0x0000212B	ERROR_DS_CROSS_DOMAIN_CLEANUP_REQD	A cross domain move operation failed such that the two versions of the moved object exist - one each in the source and destination domains. The destination object needs to be removed to restore the system to a consistent state.
8492	0x0000212C	ERROR_DS_ILLEGAL_XDOM_MOVE_OPERATIO N	This object may not be moved across domain boundaries either because cross domain moves for this class are disallowed, or the object has some special characteristics, eg: trust account or restricted RID, which prevent its move.
8493	0x0000212D	ERROR_DS_CANT_WITH_ACCT_GROUP_MEMB ERSHPS	Can't move objects with memberships across domain boundaries as once moved, this would violate the membership conditions of the account group. Remove the object from any account group memberships and retry.
8494	0x0000212E	ERROR_DS_NC_MUST_HAVE_NC_PARENT	A naming context head must be the immediate child of another naming context head, not of an interior node.
8495	0x0000212F	ERROR_DS_CR_IMPOSSIBLE_TO_VALIDATE	The directory cannot validate the proposed naming context name because it does not hold a replica of the naming context above the proposed naming context. Please ensure that the domain naming master role is held by a server that is configured as a global catalog server, and that the server is up to date with its replication partners. (Applies only to Windows 2000 Domain Naming masters)
8496	0x00002130	ERROR_DS_DST_DOMAIN_NOT_NATIVE	Destination domain must be in native mode.
8497	0x00002131	ERROR_DS_MISSING_INFRASTRUCTURE_CON TAINER	The operation can not be performed because the server does not have an infrastructure container in the domain of interest.
8498	0x00002132	ERROR_DS_CANT_MOVE_ACCOUNT_GROUP	Cross-domain move of non-empty account groups is not allowed.
8499	0x00002133	ERROR_DS_CANT_MOVE_RESOURCE_GROUP	Cross-domain move of non-empty resource groups is not allowed.

Error			Description
decimal	Hexadecimal	Name	
8500	0x00002134	ERROR_DS_INVALID_SEARCH_FLAG	The search flags for the attribute are invalid. The ANR bit is valid only on attributes of Unicode or Teletex strings.
8501	0x00002135	ERROR_DS_NO_TREE_DELETE_ABOVE_NC	Tree deletions starting at an object which has an NC head as a descendant are not allowed.
8502	0x00002136	ERROR_DS_COULDNT_LOCK_TREE_FOR_DELE TE	The directory service failed to lock a tree in preparation for a tree deletion because the tree was in use.
8503	0x00002137	ERROR_DS_COULDNT_IDENTIFY_OBJECTS_FO R_TREE_DELETE	The directory service failed to identify the list of objects to delete while attempting a tree deletion.
8504	0x00002138	ERROR_DS_SAM_INIT_FAILURE	Security Accounts Manager initialization failed because of the following error: %1. Error Status: 0x%2. Click OK to shut down the system and reboot into Directory Services Restore Mode. Check the event log for detailed information.

Error			Description
decimal	Hexadecimal	Name	
8505	0x00002139	ERROR_DS_SENSITIVE_GROUP_VIOLATION	Only an administrator can modify the membership list of an administrative group.
8506	0x0000213A	ERROR_DS_CANT_MOD_PRIMARYGROUPID	Cannot change the primary group ID of a domain controller account.
8507	0x0000213B	ERROR_DS_ILLEGAL_BASE_SCHEMA_MOD	An attempt is made to modify the base schema.
8508	0x0000213C	ERROR_DS_NONSAFE_SCHEMA_CHANGE	Adding a new mandatory attribute to an existing class, deleting a mandatory attribute from an existing class, or adding an optional attribute to the special class Top that is not a backlink attribute (directly or through inheritance, for example, by adding or deleting an auxiliary class) is not allowed.
8509	0x0000213D	ERROR_DS_SCHEMA_UPDATE_DISALLOWED	Schema update is not allowed on this DC because the DC is not the schema FSMO Role Owner.
8510	0x0000213E	ERROR_DS_CANT_CREATE_UNDER_SCHEMA	An object of this class cannot be created under the schema container. You can only create attribute-schema and class-schema objects under the schema container.
8511	0x0000213F	ERROR_DS_INSTALL_NO_SRC_SCH_VERSION	The replica/child install failed to get the objectVersion attribute on the schema container on the source DC. Either the attribute is missing on the schema container or the credentials supplied do not have permission to read it.
8512	0x00002140	ERROR_DS_INSTALL_NO_SCH_VERSION_IN_INIFILE	The replica/child install failed to read the objectVersion attribute in the SCHEMA section of the file schema.ini in the system32 directory.
8513	0x00002141	ERROR_DS_INVALID_GROUP_TYPE	The specified group type is invalid.
8514	0x00002142	ERROR_DS_NO_NEST_GLOBALGROUP_IN_MIXEDDOMAIN	Cannot nest global groups in a mixed domain if the group is security-enabled.
8515	0x00002143	ERROR_DS_NO_NEST_LOCALGROUP_IN_MIXEDDOMAIN	Cannot nest local groups in a mixed domain if the group is security-enabled.
8516	0x00002144	ERROR_DS_GLOBAL_CANT_HAVE_LOCAL_MEMBER	A global group cannot have a local group as a member.
8517	0x00002145	ERROR_DS_GLOBAL_CANT_HAVE_UNIVERSAL_MEMBER	A global group cannot have a universal group as a member.
8518	0x00002146	ERROR_DS_UNIVERSAL_CANT_HAVE_LOCAL_MEMBER	A universal group cannot have a local group as a member.
8519	0x00002147	ERROR_DS_GLOBAL_CANT_HAVE_CROSSDOMAIN_MEMBER	A global group cannot have a cross-domain member.
8520	0x00002148	ERROR_DS_LOCAL_CANT_HAVE_CROSSDOMAIN_LOCAL_MEMBER	A local group cannot have another cross-domain local group as a member.
8521	0x00002149	ERROR_DS_HAVE_PRIMARY_MEMBERS	A group with primary members cannot change to a security-disabled group.
8522	0x0000214A	ERROR_DS_STRING_SD_CONVERSION_FAILED	The schema cache load failed to convert the string default SD on a class-schema object.
8523	0x0000214B	ERROR_DS_NAMING_MASTER_GC	Only DSAs configured to be Global Catalog servers should be allowed to hold the Domain Naming Master FSMO role. (Applies only to Windows 2000 servers)
8524	0x0000214C	ERROR_DS_LOOKUP_FAILURE	The DSA operation is unable to proceed because of a DNS lookup failure.
8525	0x0000214D	ERROR_DS_COULDNT_UPDATE_SPNS	While processing a change to the DNS Host Name for an object, the Service Principal Name values could not be kept in sync.
8526	0x0000214E	ERROR_DS_CANT_RETRIEVE_SD	The Security Descriptor attribute could not be read.
8527	0x0000214F	ERROR_DS_KEY_NOT_UNIQUE.	The object requested was not found, but an object with that key was found.
8528	0x00002150	ERROR_DS_WRONG_LINKED_ATT_SYNTAX	The syntax of the linked attributed being added is incorrect. Forward links can only have syntax 2.5.5.1, 2.5.5.7, and 2.5.5.14, and backlinks can only have syntax 2.5.5.1.
8529	0x00002151	ERROR_DS_SAM_NEED_BOOTKEY_PASSWORD	Security Account Manager needs to get the boot password.
8530	0x00002152	ERROR_DS_SAM_NEED_BOOTKEY_FLOPPY	Security Account Manager needs to get the boot key from floppy disk.

Error			Description
decimal	Hexadecimal	Name	
8531	0x00002153	ERROR_DS_CANT_START	Directory Service cannot start.
8532	0x00002154	ERROR_DS_INIT_FAILURE	Directory Services could not start.
8533	0x00002155	ERROR_DS_NO_PKT_PRIVACY_ON_CONNECTION	The connection between client and server requires packet privacy or better.
8534	0x00002156	ERROR_DS_SOURCE_DOMAIN_IN_FOREST	The source domain may not be in the same forest as destination.
8535	0x00002157	ERROR_DS_DESTINATION_DOMAIN_NOT_IN_FOREST	The destination domain must be in the forest.
8536	0x00002158	ERROR_DS_DESTINATION_AUDITING_NOT_ENABLED	The operation requires that destination domain auditing be enabled.
8537	0x00002159	ERROR_DS_CANT_FIND_DC_FOR_SRC_DOMAIN	The operation couldn't locate a DC for the source domain.
8538	0x0000215A	ERROR_DS_SRC_OBJ_NOT_GROUP_OR_USER	The source object must be a group or user.
8539	0x0000215B	ERROR_DS_SRC_SID_EXISTS_IN_FOREST	The source object's SID already exists in destination forest.
8540	0x0000215C	ERROR_DS_SRC_AND_DST_OBJECT_CLASS_MISMATCH	The source and destination object must be of the same type.
8541	0x0000215D	ERROR_SAM_INIT_FAILURE	Security Accounts Manager initialization failed because of the following error: %1. Error Status: 0x%2. Click OK to shut down the system and reboot into Safe Mode. Check the event log for detailed information.
8542	0x0000215E	ERROR_DS_DRA_SCHEMA_INFO_SHIP	Schema information could not be included in the replication request.
8543	0x0000215F	ERROR_DS_DRA_SCHEMA_CONFLICT	The replication operation could not be completed due to a schema incompatibility.
8544	0x00002160	ERROR_DS_DRA_EARLIER_SCHEMA_CONLICT	The replication operation could not be completed due to a previous schema incompatibility.
8545	0x00002161	ERROR_DS_DRA_OBJ_NC_MISMATCH	The replication update could not be applied because either the source or the destination has not yet received information regarding a recent cross-domain move operation.
8546	0x00002162	ERROR_DS_NC_STILL_HAS_DSAS	The requested domain could not be deleted because there exist domain controllers that still host this domain.
8547	0x00002163	ERROR_DS_GC_REQUIRED	The requested operation can be performed only on a global catalog server.
8548	0x00002164	ERROR_DS_LOCAL_MEMBER_OF_LOCAL_ONLY	A local group can only be a member of other local groups in the same domain.
8549	0x00002165	ERROR_DS_NO_FPO_IN_UNIVERSAL_GROUPS	Foreign security principals cannot be members of universal groups.
8550	0x00002166	ERROR_DS_CANT_ADD_TO_GC	The attribute is not allowed to be replicated to the GC because of security reasons.
8551	0x00002167	ERROR_DS_NO_CHECKPOINT_WITH_PDC	The checkpoint with the PDC could not be taken because there are too many modifications being processed currently.
8552	0x00002168	ERROR_DS_SOURCE_AUDITING_NOT_ENABLED	The operation requires that source domain auditing be enabled.
8553	0x00002169	ERROR_DS_CANT_CREATE_IN_NONDOMAIN_NC	Security principal objects can only be created inside domain naming contexts.
8554	0x0000216A	ERROR_DS_INVALID_NAME_FOR_SPN	A Service Principal Name (SPN) could not be constructed because the provided hostname is not in the necessary format.
8555	0x0000216B	ERROR_DS_FILTER_USES_CONTRUCTED_ATTRS	A Filter was passed that uses constructed attributes.
8556	0x0000216C	ERROR_DS_UNICODEPWD_NOT_IN_QUOTES	The unicodePwd attribute value must be enclosed in double quotes.
8557	0x0000216D	ERROR_DS_MACHINE_ACCOUNT_QUOTA_EXCEEDED	Your computer could not be joined to the domain. You have exceeded the maximum number of computer accounts you are allowed to create in this domain. Contact your system administrator to have this limit reset or increased.
8558	0x0000216E	ERROR_DS_MUST_BE_RUN_ON_DST_DC	For security reasons, the operation must be run on the destination DC.
8559	0x0000216F	ERROR_DS_SRC_DC_MUST_BE_SP4_OR_GREATER	For security reasons, the source DC must be NT4SP4 or greater.

Error			Description
decimal	Hexadecimal	Name	
8560	0x00002170	ERROR_DS_CANT_TREE_DELETE_CRITICAL_OBJ	Critical Directory Service System objects cannot be deleted during tree delete operations. The tree delete may have been partially performed.
8561	0x00002171	ERROR_DS_INIT_FAILURE_CONSOLE	Directory Services could not start because of the following error: %1. Error Status: 0x%2. Please click OK to shutdown the system. You can use the recovery console to diagnose the system further.
8562	0x00002172	ERROR_DS_SAM_INIT_FAILURE_CONSOLE	Security Accounts Manager initialization failed because of the following error: %1. Error Status: 0x%2. Please click OK to shutdown the system. You can use the recovery console to diagnose the system further.
8563	0x00002173	ERROR_DS_FOREST_VERSION_TOO_HIGH	This version of Windows is too old to support the current directory forest behavior. You must upgrade the operating system on this server before it can become a domain controller in this forest.
8564	0x00002174	ERROR_DS_DOMAIN_VERSION_TOO_HIGH	This version of Windows is too old to support the current domain behavior. You must upgrade the operating system on this server before it can become a domain controller in this domain.
8565	0x00002175	ERROR_DS_FOREST_VERSION_TOO_LOW	This version of Windows no longer supports the behavior version in use in this directory forest. You must advance the forest behavior version before this server can become a domain controller in the forest.
8566	0x00002176	ERROR_DS_DOMAIN_VERSION_TOO_LOW	This version of Windows no longer supports the behavior version in use in this domain. You must advance the domain behavior version before this server can become a domain controller in the domain.
8567	0x00002177	ERROR_DS_INCOMPATIBLE_VERSION	The version of Windows is incompatible with the behavior version of the domain or forest.
8568	0x00002178	ERROR_DS_LOW_DSA_VERSION	The behavior version cannot be increased to the requested value because Domain Controllers still exist with versions lower than the requested value.
8569	0x00002179	ERROR_DS_NO_BEHAVIOR_VERSION_IN_MIXEDDOMAIN	The behavior version value cannot be increased while the domain is still in mixed domain mode. You must first change the domain to native mode before increasing the behavior version.
8570	0x0000217A	ERROR_DS_NOT_SUPPORTED_SORT_ORDER	The sort order requested is not supported.
8571	0x0000217B	ERROR_DS_NAME_NOT_UNIQUE	Found an object with a non unique name.
8572	0x0000217C	ERROR_DS_MACHINE_ACCOUNT_CREATED_PRENT4	The machine account was created pre-NT4. The account needs to be recreated.
8573	0x0000217D	ERROR_DS_OUT_OF_VERSION_STORE	The database is out of version store.
8574	0x0000217E	ERROR_DS_INCOMPATIBLE_CONTROLS_USED	Unable to continue operation because multiple conflicting controls were used.
8575	0x0000217F	ERROR_DS_NO_REF_DOMAIN	Unable to find a valid security descriptor reference domain for this partition.
8576	0x00002180	ERROR_DS_RESERVED_LINK_ID	Schema update failed: The link identifier is reserved.
8577	0x00002181	ERROR_DS_LINK_ID_NOT_AVAILABLE	Schema update failed: There are no link identifiers available.
8578	0x00002182	ERROR_DS_AG_CANT_HAVE_UNIVERSAL_MEMBER	An account group can not have a universal group as a member.
8579	0x00002183	ERROR_DS_MODIFYDN_DISALLOWED_BY_INSTANCE_TYPE	Rename or move operations on naming context heads or read-only objects are not allowed.
8580	0x00002184	ERROR_DS_NO_OBJECT_MOVE_IN_SCHEMA_NC	Move operations on objects in the schema naming context are not allowed.
8581	0x00002185	ERROR_DS_MODIFYDN_DISALLOWED_BY_FLAG	A system flag has been set on the object and does not allow the object to be moved or renamed.
8582	0x00002186	ERROR_DS_MODIFYDN_WRONG_GRANDPARENT	This object is not allowed to change its grandparent container. Moves are not forbidden on this object, but are restricted to sibling containers.

Error			Description
decimal	Hexadecimal	Name	
8583	0x00002187	ERROR_DS_NAME_ERROR_TRUST_REFERRAL	Unable to resolve completely, a referral to another forest is generated.
8584	0x00002188	ERROR_NOT_SUPPORTED_ON_STANDARD_SERVER	The requested action is not supported on standard server.
8585	0x00002189	ERROR_DS_CANT_ACCESS_REMOTE_PART_OF_AD	Could not access a partition of the Active Directory located on a remote server. Make sure at least one server is running for the partition in question.
8586	0x0000218A	ERROR_DS_CR_IMPOSSIBLE_TO_VALIDATE_V2	The directory cannot validate the proposed naming context (or partition) name because it does not hold a replica nor can it contact a replica of the naming context above the proposed naming context. Please ensure that the parent naming context is properly registered in DNS, and at least one replica of this naming context is reachable by the Domain Naming master.
8587	0x0000218B	ERROR_DS_THREAD_LIMIT_EXCEEDED	The thread limit for this request was exceeded.
8588	0x0000218C	ERROR_DS_NOT_CLOSEST	The Global catalog server is not in the closet site.

Error			Description
decimal	Hexadecimal	Name	
9001	0x00002329	DNS_ERROR_RCODE_FORMAT_ERROR	DNS server unable to interpret format.
9002	0x0000232A	DNS_ERROR_RCODE_SERVER_FAILURE	DNS server failure.
9003	0x0000232B	DNS_ERROR_RCODE_NAME_ERROR	DNS name does not exist.
9004	0x0000232C	DNS_ERROR_RCODE_NOT_IMPLEMENTED	DNS request not supported by name server.
9005	0x0000232D	DNS_ERROR_RCODE_REFUSED	DNS operation refused.
9006	0x0000232E	DNS_ERROR_RCODE_YXDOMAIN	DNS name that ought not exist, does exist.
9007	0x0000232F	DNS_ERROR_RCODE_YXRRSET	DNS RR set that ought not exist, does exist.
9008	0x00002330	DNS_ERROR_RCODE_NXRRSET	DNS RR set that ought to exist, does not exist.
9009	0x00002331	DNS_ERROR_RCODE_NOTAUTH	DNS server not authoritative for zone.
9010	0x00002332	DNS_ERROR_RCODE_NOTZONE	DNS name in update or prereq is not in zone.
9016	0x00002338	DNS_ERROR_RCODE_BADSIG	DNS signature failed to verify.
9017	0x00002339	DNS_ERROR_RCODE_BADKEY	DNS bad key.
9018	0x0000233A	DNS_ERROR_RCODE_BADTIME	DNS signature validity expired.
9501	0x0000251D	DNS_INFO_NO_RECORDS	No records found for given DNS query.
9502	0x0000251E	DNS_ERROR_BAD_PACKET	Bad DNS packet.
9503	0x0000251F	DNS_ERROR_NO_PACKET	No DNS packet.
9504	0x00002520	DNS_ERROR_RCODE	DNS error, check rcode.
9505	0x00002521	DNS_ERROR_UNSECURE_PACKET	Unsecured DNS packet.
9551	0x0000254F	DNS_ERROR_INVALID_TYPE	Invalid DNS type.
9552	0x00002550	DNS_ERROR_INVALID_IP_ADDRESS	Invalid IP address.
9553	0x00002551	DNS_ERROR_INVALID_PROPERTY	Invalid property.
9554	0x00002552	DNS_ERROR_TRY_AGAIN_LATER	Try DNS operation again later.
9555	0x00002553	DNS_ERROR_NOT_UNIQUE	Record for given name and type is not unique.
9556	0x00002554	DNS_ERROR_NON_RFC_NAME	DNS name does not comply with RFC specifications.
9557	0x00002555	DNS_STATUS_FQDN	DNS name is a fully-qualified DNS name.
9558	0x00002556	DNS_STATUS_DOTTED_NAME	DNS name is dotted (multi-label).
9559	0x00002557	DNS_STATUS_SINGLE_PART_NAME	DNS name is a single-part name.
9560	0x00002558	DNS_ERROR_INVALID_NAME_CHAR	DSN name contains an invalid character.
9561	0x00002559	DNS_ERROR_NUMERIC_NAME	DNS name is entirely numeric.
9601	0x00002581	DNS_ERROR_ZONE_DOES_NOT_EXIST	DNS zone does not exist.
9602	0x00002582	DNS_ERROR_NO_ZONE_INFO	DNS zone information not available.
9603	0x00002583	DNS_ERROR_INVALID_ZONE_OPERATION	Invalid operation for DNS zone.
9604	0x00002584	DNS_ERROR_ZONE_CONFIGURATION_ERROR	Invalid DNS zone configuration.
9605	0x00002585	DNS_ERROR_ZONE_HAS_NO_SOA_RECORD	DNS zone has no start of authority (SOA) record.
9606	0x00002586	DNS_ERROR_ZONE_HAS_NO_NS_RECORDS	DNS zone has no name server (NS) record.
9607	0x00002587	DNS_ERROR_ZONE_LOCKED	DNS zone is locked.
9608	0x00002588	DNS_ERROR_ZONE_CREATION_FAILED	DNS zone creation failed.
9609	0x00002589	DNS_ERROR_ZONE_ALREADY_EXISTS	DNS zone already exists.

Error			Description
decimal	Hexadecimal	Name	
9610	0x0000258A	DNS_ERROR_AUTOZONE_ALREADY_EXISTS	DNS automatic zone already exists.
9611	0x0000258B	DNS_ERROR_INVALID_ZONE_TYPE	Invalid DNS zone type.
9612	0x0000258C	DNS_ERROR_SECONDARY_REQUIRES_MASTER_IP	Secondary DNS zone requires master IP address.
9613	0x0000258D	DNS_ERROR_ZONE_NOT_SECONDARY	DNS zone not secondary.
9614	0x0000258E	DNS_ERROR_NEED_SECONDARY_ADDRESSES	Need secondary IP address.
9615	0x0000258F	DNS_ERROR_WINS_INIT_FAILED	WINS initialization failed.
9616	0x00002590	DNS_ERROR_NEED_WINS_SERVERS	Need WINS servers.
9617	0x00002591	DNS_ERROR_NBSTAT_INIT_FAILED	NBTSTAT initialization call failed.
9618	0x00002592	DNS_ERROR_SOA_DELETE_INVALID	Invalid delete of start of authority (SOA)
9619	0x00002593	DNS_ERROR_FORWARDER_ALREADY_EXISTS	A conditional forwarding zone already exists for that name.
9651	0x000025B3	DNS_ERROR_PRIMARY_REQUIRES_DATAFILE	Primary DNS zone requires datafile.
9652	0x000025B4	DNS_ERROR_INVALID_DATAFILE_NAME	Invalid datafile name for DNS zone.
9653	0x000025B5	DNS_ERROR_DATAFILE_OPEN_FAILURE	Failed to open datafile for DNS zone.
9654	0x000025B6	DNS_ERROR_FILE_WRITEBACK_FAILED	Failed to write datafile for DNS zone.
9655	0x000025B7	DNS_ERROR_DATAFILE_PARSING	Failure while reading datafile for DNS zone.
9701	0x000025E5	DNS_ERROR_RECORD_DOES_NOT_EXIST	DNS record does not exist.
9702	0x000025E6	DNS_ERROR_RECORD_FORMAT	DNS record format error.
9703	0x000025E7	DNS_ERROR_NODE_CREATION_FAILED	Node creation failure in DNS.
9704	0x000025E8	DNS_ERROR_UNKNOWN_RECORD_TYPE	Unknown DNS record type.
9705	0x000025E9	DNS_ERROR_RECORD_TIMED_OUT	DNS record timed out.
9706	0x000025EA	DNS_ERROR_NAME_NOT_IN_ZONE	Name not in DNS zone.
9707	0x000025EB	DNS_ERROR_CNAME_LOOP	CNAME loop detected.
9708	0x000025EC	DNS_ERROR_NODE_IS_CNAME	Node is a CNAME DNS record.
9709	0x000025ED	DNS_ERROR_CNAME_COLLISION	A CNAME record already exists for given name.
9710	0x000025EE	DNS_ERROR_RECORD_ONLY_AT_ZONE_ROOT	Record only at DNS zone root.
9711	0x000025EF	DNS_ERROR_RECORD_ALREADY_EXISTS	DNS record already exists.
9712	0x000025F0	DNS_ERROR_SECONDARY_DATA	Secondary DNS zone data error.
9713	0x000025F1	DNS_ERROR_NO_CREATE_CACHE_DATA	Could not create DNS cache data.
9714	0x000025F2	DNS_ERROR_NAME_DOES_NOT_EXIST	DNS name does not exist.
9715	0x000025F3	DNS_WARNING_PTR_CREATE_FAILED	Could not create pointer (PTR) record.
9716	0x000025F4	DNS_WARNING_DOMAIN_UNDELETED	DNS domain was undeleted.
9717	0x000025F5	DNS_ERROR_DS_UNAVAILABLE	The directory service is unavailable.
9718	0x000025F6	DNS_ERROR_DS_ZONE_ALREADY_EXISTS	DNS zone already exists in the directory service.
9719	0x000025F7	DNS_ERROR_NO_BOOTFILE_IF_DS_ZONE	DNS server not creating or reading the boot file for the directory service integrated DNS zone.
9751	0x00002617	DNS_INFO_AXFR_COMPLETE	DNS AXFR (zone transfer) complete.
9752	0x00002618	DNS_ERROR_AXFR	DNS zone transfer failed.
9753	0x00002619	DNS_INFO_ADDED_LOCAL_WINS	Added local WINS server.
9801	0x00002649	DNS_STATUS_CONTINUE_NEEDED	Secure update call needs to continue update request.
9851	0x0000267B	DNS_ERROR_NO_TCPIP	TCP/IP network protocol not installed.
9852	0x0000267C	DNS_ERROR_NO_DNS_SERVERS	No DNS servers configured for local system.
9901	0x000026AD	DNS_ERROR_DP_DOES_NOT_EXIST	The specified directory partition does not exist.
9902	0x000026AE	DNS_ERROR_DP_ALREADY_EXISTS	The specified directory partition already exists.
9903	0x000026AF	DNS_ERROR_DP_NOT_ENLISTED	The DS is not enlisted in the specified directory partition.
9904	0x000026B0	DNS_ERROR_DP_ALREADY_ENLISTED	The DS is already enlisted in the specified directory partition.

Error			Description
decimal	Hexadecimal	Name	
10004	0x00002714	WSAEINTR	A blocking operation was interrupted by a call to WSACancelBlockingCall.
10009	0x00002719	WSAEBADF	The file handle supplied is not valid.
10013	0x0000271D	WSAEACCES	An attempt was made to access a socket in a way forbidden by its access permissions.

Error			Description
decimal	Hexadecimal	Name	
10014	0x0000271E	WSAEFAULT	The system detected an invalid pointer address in attempting to use a pointer argument in a call.
10022	0x00002726	WSAEINVAL	An invalid argument was supplied.
10024	0x00002728	WSAEMFILE	Too many open sockets.
10035	0x00002733	WSAEWOULDBLOCK	A non-blocking socket operation could not be completed immediately.
10036	0x00002734	WSAEINPROGRESS	A blocking operation is currently executing.
10037	0x00002735	WSAEALREADY	An operation was attempted on a non-blocking socket that already had an operation in progress.
10038	0x00002736	WSAENOTSOCK	An operation was attempted on something that is not a socket.
10039	0x00002737	WSAEDESTADDRREQ	A required address was omitted from an operation on a socket.
10040	0x00002738	WSAEMSGSIZE	A message sent on a datagram socket was larger than the internal message buffer or some other network limit, or the buffer used to receive a datagram into was smaller than the datagram itself.
10041	0x00002739	WSAEPROTOTYPE	A protocol was specified in the socket function call that does not support the semantics of the socket type requested.
10042	0x0000273A	WSAENOPROTOOPT	An unknown, invalid, or unsupported option or level was specified in a getsockopt or setsockopt call.
10043	0x0000273B	WSAEPROTONOSUPPORT	The requested protocol has not been configured into the system, or no implementation for it exists.
10044	0x0000273C	WSAESOCKTNOSUPPORT	The support for the specified socket type does not exist in this address family.
10045	0x0000273D	WSAEOPNOTSUPP	The attempted operation is not supported for the type of object referenced.
10046	0x0000273E	WSAEPFNOSUPPORT	The protocol family has not been configured into the system or no implementation for it exists.
10047	0x0000273F	WSAEAFNOSUPPORT	An address incompatible with the requested protocol was used.
10048	0x00002740	WSAEADDRINUSE	Only one usage of each socket address (protocol/ network address/port) is normally permitted.
10049	0x00002741	WSAEADDRNOTAVAIL	The requested address is not valid in its context.
10050	0x00002742	WSAENETDOWN	A socket operation encountered a dead network.
10051	0x00002743	WSAENETUNREACH	A socket operation was attempted to an unreachable network.
10052	0x00002744	WSAENETRESET	The connection has been broken due to keep-alive activity detecting a failure while the operation was in progress.
10053	0x00002745	WSAECONNABORTED	An established connection was aborted by the software in your host machine.
10054	0x00002746	WSAECONNRESET	An existing connection was forcibly closed by the remote host.
10055	0x00002747	WSAENOBUFS	An operation on a socket could not be performed because the system lacked sufficient buffer space or because a queue was full.
10056	0x00002748	WSAEISCONN	A connect request was made on an already connected socket.
10057	0x00002749	WSAENOTCONN	A request to send or receive data was disallowed because the socket is not connected and (when sending on a datagram socket using a sendto call) no address was supplied.
10058	0x0000274A	WSAESHUTDOWN	A request to send or receive data was disallowed because the socket had already been shut down in that direction with a previous shutdown call.
10059	0x0000274B	WSAETOOMANYREFS	Too many references to some kernel object.
10060	0x0000274C	WSAETIMEDOUT	A connection attempt failed because the connected party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.
10061	0x0000274D	WSAECONNREFUSED	No connection could be made because the target machine actively refused it.

Error			Description
decimal	Hexadecimal	Name	
10062	0x0000274E	WSAELOOP	Cannot translate name.
10063	0x0000274F	WSAENAMETOOLONG	Name component or name was too long.
10064	0x00002750	WSAEHOSTDOWN	A socket operation failed because the destination host was down.
10065	0x00002751	WSAEHOSTUNREACH	A socket operation was attempted to an unreachable host.
10066	0x00002752	WSAENOTEMPTY	Cannot remove a directory that is not empty.
10067	0x00002753	WSAEPROCLIM	A Windows Sockets implementation may have a limit on the number of applications that may use it simultaneously.
10068	0x00002754	WSAEUSERS	Ran out of quota.
10069	0x00002755	WSAEDQUOT	Ran out of disk quota.
10070	0x00002756	WSAESTALE	File handle reference is no longer available.
10071	0x00002757	WSAEREMOTE	Item is not available locally.
10091	0x0000276B	WSASYSNOTREADY	WSAStartup cannot function at this time because the underlying system it uses to provide network services is currently unavailable.
10092	0x0000276C	WSAVERNOTSUPPORTED	The Windows Sockets version requested is not supported.
10093	0x0000276D	WSANOTINITIALISED	Either the application has not called WSAStartup, or WSAStartup failed.
10101	0x00002775	WSAEDISCON	Returned by WSARecv or WSARecvFrom to indicate the remote party has initiated a graceful shutdown sequence.
10102	0x00002776	WSAENOMORE	No more results can be returned by WSALookupServiceNext.
10103	0x00002777	WSAECANCELLED	A call to WSALookupServiceEnd was made while this call was still processing. The call has been canceled.
10104	0x00002778	WSAEINVALIDPROCTABLE	The procedure call table is invalid.
10105	0x00002779	WSAEINVALIDPROVIDER	The requested service provider is invalid.
10106	0x0000277A	WSAEPROVIDERFAILEDINIT	The requested service provider could not be loaded or initialized.
10107	0x0000277B	WSASYSCALLFAILURE	A system call that should never fail has failed.
10108	0x0000277C	WSASERVICE_NOT_FOUND	No such service is known. The service cannot be found in the specified name space.
10109	0x0000277D	WSATYPE_NOT_FOUND	The specified class was not found.
10110	0x0000277E	WSA_E_NO_MORE	No more results can be returned by WSALookupServiceNext.
10111	0x0000277F	WSA_E_CANCELLED	A call to WSALookupServiceEnd was made while this call was still processing. The call has been canceled.
10112	0x00002780	WSAEREFUSED	A database query failed because it was actively refused.
11001	0x00002AF9	WSAHOST_NOT_FOUND	No such host is known.
11002	0x00002AFA	WSATRY_AGAIN	This is usually a temporary error during hostname resolution and means that the local server did not receive a response from an authoritative server.
11003	0x00002AFB	WSANO_RECOVERY	A non-recoverable error occurred during a database lookup.
11004	0x00002AFC	WSANO_DATA	The requested name is valid and was found in the database, but it does not have the correct associated data being resolved for.
11005	0x00002AFD	WSA_QOS_RECEIVERS	At least one reserve has arrived.
11006	0x00002AFE	WSA_QOS_SENDERS	At least one path has arrived.
11007	0x00002AFF	WSA_QOS_NO_SENDERS	There are no senders.
11008	0x00002B00	WSA_QOS_NO_RECEIVERS	There are no receivers.
11009	0x00002B01	WSA_QOS_REQUEST_CONFIRMED	Reserve has been confirmed.
11010	0x00002B02	WSA_QOS_ADMISSION_FAILURE	Error due to lack of resources.
11011	0x00002B03	WSA_QOS_POLICY_FAILURE	Rejected for administrative reasons - bad credentials.
11012	0x00002B04	WSA_QOS_BAD_STYLE	Unknown or conflicting style.

Error			Description
decimal	Hexadecimal	Name	
11013	0x00002B05	WSA_QOS_BAD_OBJECT	Problem with some part of the filterspec or providerspecific buffer in general.
11014	0x00002B06	WSA_QOS_TRAFFIC_CTRL_ERROR	Problem with some part of the flowspec.
11015	0x00002B07	WSA_QOS_GENERIC_ERROR	General QOS error.
11016	0x00002B08	WSA_QOS_ESERVICETYPE	An invalid or unrecognized service type was found in the flowspec.
11017	0x00002B09	WSA_QOS_EFLOWSPEC	An invalid or inconsistent flowspec was found in the QOS structure.
11018	0x00002B0A	WSA_QOS_EPROVSPECBUF	Invalid QOS provider-specific buffer.
11019	0x00002B0B	WSA_QOS_EFILTERSTYLE	An invalid QOS filter style was used.
11020	0x00002B0C	WSA_QOS_EFILTERTYPE	An invalid QOS filter type was used.
11021	0x00002B0D	WSA_QOS_EFILTERCOUNT	An incorrect number of QOS FILTERSPECs were specified in the FLOWDESCRIPTOR.
11022	0x00002B0E	WSA_QOS_EOBJLENGTH	An object with an invalid ObjectLength field was specified in the QOS provider-specific buffer.
11023	0x00002B0F	WSA_QOS_EFLOWCOUNT	An incorrect number of flow descriptors was specified in the QOS structure.
11024	0x00002B10	WSA_QOS_EUNKNOWNPSOBJ	An unrecognized object was found in the QOS provider-specific buffer.
11025	0x00002B11	WSA_QOS_EPOLICYOBJ	An invalid policy object was found in the QOS provider-specific buffer.
11026	0x00002B12	WSA_QOS_EFLOWDESC	An invalid QOS flow descriptor was found in the flow descriptor list.
11027	0x00002B13	WSA_QOS_EPSFLOWSPEC	An invalid or inconsistent flowspec was found in the QOS provider-specific buffer.
11028	0x00002B14	WSA_QOS_EPSFILTERSPEC	An invalid FILTERSPEC was found in the QOS provider-specific buffer.
11029	0x00002B15	WSA_QOS_ESDMODEOBJ	An invalid shape discard mode object was found in the QOS provider-specific buffer.
11030	0x00002B16	WSA_QOS_ESHAPERATEOBJ	An invalid shaping rate object was found in the QOS provider-specific buffer.
11031	0x00002B17	WSA_QOS_RESERVED_PETYPE	A reserved policy element was found in the QOS provider-specific buffer.

Error			Description
decimal	Hexadecimal	Name	
12000	0x00002EE0	ERROR_SXS_SECTION_NOT_FOUND	The requested section was not present in the activation context.
12001	0x00002EE1	ERROR_SXS_CANT_GEN_ACTCTX	This application has failed to start because the application configuration is incorrect. Reinstalling the application may fix this problem.
12002	0x00002EE2	ERROR_SXS_INVALID_ACTCTXDATA_FORMAT	The application binding data format is invalid.
12003	0x00002EE3	ERROR_SXS_ASSEMBLY_NOT_FOUND	The referenced assembly is not installed on your system.
12004	0x00002EE4	ERROR_SXS_MANIFEST_FORMAT_ERROR	The manifest file does not begin with the required tag and format information.
12005	0x00002EE5	ERROR_SXS_MANIFEST_PARSE_ERROR	The manifest file contains one or more syntax errors.
12006	0x00002EE6	ERROR_SXS_ACTIVATION_CONTEXT_DISABLED	The application attempted to activate a disabled activation context.
12007	0x00002EE7	ERROR_SXS_KEY_NOT_FOUND	The requested lookup key was not found in any active activation context.
12008	0x00002EE8	ERROR_SXS_VERSION_CONFLICT	A component version required by the application conflicts with another component version already active.
12009	0x00002EE9	ERROR_SXS_WRONG_SECTION_TYPE	The type requested activation context section does not match the query API used.
12010	0x00002EEA	ERROR_SXS_THREAD_QUERIES_DISABLED	Lack of system resources has required isolated activation to be disabled for the current thread of execution.
12011	0x00002EEB	ERROR_SXS_PROCESS_DEFAULT_ALREADY_SET	An attempt to set the process default activation context failed because the process default activation context was already set.
12012	0x00002EEC	ERROR_SXS_UNKNOWN_ENCODING_GROUP	The encoding group identifier specified is not recognized.

Error			Description
decimal	Hexadecimal	Name	
12013	0x00002EED	ERROR_SXS_UNKNOWN_ENCODING	The encoding requested is not recognized.
12014	0x00002EEE	ERROR_SXS_INVALID_XML_NAMESPACE_URI	The manifest contains a reference to an invalid URI.
12015	0x00002EEF	ERROR_SXS_ROOT_MANIFEST_DEPENDENCY_NOT_INSTALLED	The application manifest contains a reference to a dependent assembly which is not installed.
12016	0x00002EF0	ERROR_SXS_LEAF_MANIFEST_DEPENDENCY_NOT_INSTALLED	The manifest for an assembly used by the application has a reference to a dependent assembly which is not installed.
12017	0x00002EF1	ERROR_SXS_INVALID_ASSEMBLY_IDENTITY_ATTRIBUTE	The manifest contains an attribute for the assembly identity which is not valid.
12018	0x00002EF2	ERROR_SXS_MANIFEST_MISSING_REQUIRED_DEFAULT_NAMESPACE	The manifest is missing the required default namespace specification on the assembly element.
12019	0x00002EF3	ERROR_SXS_MANIFEST_INVALID_REQUIRED_DEFAULT_NAMESPACE	The manifest has a default namespace specified on the assembly element but its value is not "urn:schemas-microsoft-com:asm.v1".
12020	0x00002EF4	ERROR_SXS_PRIVATE_MANIFEST_CROSS_PATH_ WITH_REPARSE_POINT	The private manifest probe has crossed the reparse-point-associated path.
12021	0x00002EF5	ERROR_SXS_DUPLICATE_DLL_NAME	Two or more components referenced directly or indirectly by the application manifest have files by the same name.
12022	0x00002EF6	ERROR_SXS_DUPLICATE_WINDOWCLASS_NAME	Two or more components referenced directly or indirectly by the application manifest have window classes with the same name.
12023	0x00002EF7	ERROR_SXS_DUPLICATE_CLSID	Two or more components referenced directly or indirectly by the application manifest have the same COM server CLSIDs.
12024	0x00002EF8	ERROR_SXS_DUPLICATE_IID	Two or more components referenced directly or indirectly by the application manifest have proxies for the same COM interface IIDs.
12025	0x00002EF9	ERROR_SXS_DUPLICATE_TLBID	Two or more components referenced directly or indirectly by the application manifest have the same COM type library TLBIDs.
12026	0x00002EFA	ERROR_SXS_DUPLICATE_PROGID	Two or more components referenced directly or indirectly by the application manifest have the same COM ProgIDs.
12027	0x00002EFB	ERROR_SXS_DUPLICATE_ASSEMBLY_NAME	Two or more components referenced directly or indirectly by the application manifest are different versions of the same component which is not permitted.
12028	0x00002EFC	ERROR_SXS_FILE_HASH_MISMATCH	A component's file does not match the verification information present in the component manifest.
12029	0x00002EFD	ERROR_SXS_POLICY_PARSE_ERROR	The policy manifest contains one or more syntax errors.
12030	0x00002EFE	ERROR_SXS_XML_E_MISSINGQUOTE	Manifest Parse Error : A string literal was expected, but no opening quote character was found.
12031	0x00002EFF	ERROR_SXS_XML_E_COMMENTSYNTAX	Manifest Parse Error : Incorrect syntax was used in a comment.
12032	0x00002F00	ERROR_SXS_XML_E_BADSTARTNAMECHAR	Manifest Parse Error : A name was started with an invalid character.
12033	0x00002F01	ERROR_SXS_XML_E_BADNAMECHAR	Manifest Parse Error : A name contained an invalid character.
12034	0x00002F02	ERROR_SXS_XML_E_BADCHARINSTRING	Manifest Parse Error : A string literal contained an invalid character.
12035	0x00002F03	ERROR_SXS_XML_E_XMLDECLSYNTAX	Manifest Parse Error : Invalid syntax for an XML declaration.
12036	0x00002F04	ERROR_SXS_XML_E_BADCHARDATA	Manifest Parse Error : An invalid character was found in text content.
12037	0x00002F05	ERROR_SXS_XML_E_MISSINGWHITESPACE	Manifest Parse Error : Required white space was missing.
12038	0x00002F06	ERROR_SXS_XML_E_EXPECTINGTAGEND	Manifest Parse Error : The character '>' was expected.
12039	0x00002F07	ERROR_SXS_XML_E_MISSINGSEMICOLON	Manifest Parse Error : A semi colon character was expected.
12040	0x00002F08	ERROR_SXS_XML_E_UNBALANCEDPAREN	Manifest Parse Error : Unbalanced parentheses.

Error			Description
decimal	Hexadecimal	Name	
12041	0x00002F09	ERROR_SXS_XML_E_INTERNALERROR	Manifest Parse Error : Internal error.
12042	0x00002F0A	ERROR_SXS_XML_E_UNEXPECTED_WHITESPACE	Manifest Parse Error : White space is not allowed at this location.
12043	0x00002F0B	ERROR_SXS_XML_E_INCOMPLETE_ENCODING	Manifest Parse Error : End of file reached in invalid state for current encoding.
12044	0x00002F0C	ERROR_SXS_XML_E_MISSING_PAREN	Manifest Parse Error : Missing parenthesis.
12045	0x00002F0D	ERROR_SXS_XML_E_EXPECTINGCLOSEQUOTE	Manifest Parse Error : A single or double closing quote character (\' or \") is missing.
12046	0x00002F0E	ERROR_SXS_XML_E_MULTIPLE_COLONS	Manifest Parse Error : Multiple colons are not allowed in a name.
12047	0x00002F0F	ERROR_SXS_XML_E_INVALID_DECIMAL	Manifest Parse Error : Invalid character for decimal digit.
12048	0x00002F10	ERROR_SXS_XML_E_INVALID_HEXIDECIMAL	Manifest Parse Error : Invalid character for hexadecimal digit.
12049	0x00002F11	ERROR_SXS_XML_E_INVALID_UNICODE	Manifest Parse Error : Invalid Unicode character value for this platform.
12050	0x00002F12	ERROR_SXS_XML_E_WHITESPACEORQUESTIONMARK	Manifest Parse Error : Expecting white space or '?'.
12051	0x00002F13	ERROR_SXS_XML_E_UNEXPECTEDENDTAG	Manifest Parse Error : End tag was not expected at this location.
12052	0x00002F14	ERROR_SXS_XML_E_UNCLOSEDTAG	Manifest Parse Error : The following tags were not closed: %1.
12053	0x00002F15	ERROR_SXS_XML_E_DUPLICATEATTRIBUTE	Manifest Parse Error : Duplicate attribute.
12054	0x00002F16	ERROR_SXS_XML_E_MULTIPLEROOTS	Manifest Parse Error : Only one top level element is allowed in an XML document.
12055	0x00002F17	ERROR_SXS_XML_E_INVALIDATROOTLEVEL	Manifest Parse Error : Invalid at the top level of the document.
12056	0x00002F18	ERROR_SXS_XML_E_BADXMLDECL	Manifest Parse Error : Invalid XML declaration.
12057	0x00002F19	ERROR_SXS_XML_E_MISSINGROOT	Manifest Parse Error : XML document must have a top level element.
12058	0x00002F1A	ERROR_SXS_XML_E_UNEXPECTEDEOF	Manifest Parse Error : Unexpected end of file.
12059	0x00002F1B	ERROR_SXS_XML_E_BADPEREFINSUBSET	Manifest Parse Error : Parameter entities cannot be used inside markup declarations in an internal subset.
12060	0x00002F1C	ERROR_SXS_XML_E_UNCLOSEDSTARTTAG	Manifest Parse Error : Element was not closed.
12061	0x00002F1D	ERROR_SXS_XML_E_UNCLOSEDENDTAG	Manifest Parse Error : End element was missing the character '>'.
12062	0x00002F1E	ERROR_SXS_XML_E_UNCLOSEDSTRING	Manifest Parse Error : A string literal was not closed.
12063	0x00002F1F	ERROR_SXS_XML_E_UNCLOSEDCOMMENT	Manifest Parse Error : A comment was not closed.
12064	0x00002F20	ERROR_SXS_XML_E_UNCLOSEDDECL	Manifest Parse Error : A declaration was not closed.
12065	0x00002F21	ERROR_SXS_XML_E_UNCLOSEDCDATA	Manifest Parse Error : A CDATA section was not closed.
12066	0x00002F22	ERROR_SXS_XML_E_RESERVEDNAMESPACE	Manifest Parse Error : The namespace prefix is not allowed to start with the reserved string "xml".
12067	0x00002F23	ERROR_SXS_XML_E_INVALIDENCODING	Manifest Parse Error : System does not support the specified encoding.
12068	0x00002F24	ERROR_SXS_XML_E_INVALIDSWITCH	Manifest Parse Error : Switch from current encoding to specified encoding not supported.
12069	0x00002F25	ERROR_SXS_XML_E_BADXMLCASE	Manifest Parse Error : The name 'xml' is reserved and must be lower case.
12070	0x00002F26	ERROR_SXS_XML_E_INVALID_STANDALONE	Manifest Parse Error : The standalone attribute must have the value 'yes' or 'no'.
12071	0x00002F27	ERROR_SXS_XML_E_UNEXPECTED_STANDALONE	Manifest Parse Error : The standalone attribute cannot be used in external entities.
12072	0x00002F28	ERROR_SXS_XML_E_INVALID_VERSION	Manifest Parse Error : Invalid version number.
12073	0x00002F29	ERROR_SXS_XML_E_MISSINGEQUALS	Manifest Parse Error : Missing equals sign between attribute and attribute value.
13000	0x000032C8	ERROR_IPSEC_QM_POLICY_EXISTS	The specified quick mode policy already exists.
13001	0x000032C9	ERROR_IPSEC_QM_POLICY_NOT_FOUND	The specified quick mode policy was not found.
13002	0x000032CA	ERROR_IPSEC_QM_POLICY_IN_USE	The specified quick mode policy is being used.
13003	0x000032CB	ERROR_IPSEC_MM_POLICY_EXISTS	The specified main mode policy already exists.

Error			Description
decimal	Hexadecimal	Name	
13004	0x000032CC	ERROR_IPSEC_MM_POLICY_NOT_FOUND	The specified main mode policy was not found.
13005	0x000032CD	ERROR_IPSEC_MM_POLICY_IN_USE	The specified main mode policy is being used.
13006	0x000032CE	ERROR_IPSEC_MM_FILTER_EXISTS	The specified main mode filter already exists.
13007	0x000032CF	ERROR_IPSEC_MM_FILTER_NOT_FOUND	The specified main mode filter was not found.
13008	0x000032D0	ERROR_IPSEC_TRANSPORT_FILTER_EXISTS	The specified transport mode filter already exists.
13009	0x000032D1	ERROR_IPSEC_TRANSPORT_FILTER_NOT_FOUND	The specified transport mode filter does not exist.
13010	0x000032D2	ERROR_IPSEC_MM_AUTH_EXISTS	The specified main mode authentication list exists.
13011	0x000032D3	ERROR_IPSEC_MM_AUTH_NOT_FOUND	The specified main mode authentication list was not found.
13012	0x000032D4	ERROR_IPSEC_MM_AUTH_IN_USE	The specified quick mode policy is being used.
13013	0x000032D5	ERROR_IPSEC_DEFAULT_MM_POLICY_NOT_FOUND	The specified main mode policy was not found.
13014	0x000032D6	ERROR_IPSEC_DEFAULT_MM_AUTH_NOT_FOUND	The specified quick mode policy was not found.
13015	0x000032D7	ERROR_IPSEC_DEFAULT_QM_POLICY_NOT_FOUND	The manifest file contains one or more syntax errors.
13016	0x000032D8	ERROR_IPSEC_TUNNEL_FILTER_EXISTS	The application attempted to activate a disabled activation context.
13017	0x000032D9	ERROR_IPSEC_TUNNEL_FILTER_NOT_FOUND	The requested lookup key was not found in any active activation context.
13018	0x000032DA	ERROR_IPSEC_MM_FILTER_PENDING_DELETION	The Main Mode filter is pending deletion.
13019	0x000032DB	ERROR_IPSEC_TRANSPORT_FILTER_PENDING_DELETION	The transport filter is pending deletion.
13020	0x000032DC	ERROR_IPSEC_TUNNEL_FILTER_PENDING_DELETION	The tunnel filter is pending deletion.
13021	0x000032DD	ERROR_IPSEC_MM_POLICY_PENDING_DELETION	The Main Mode policy is pending deletion.
13022	0x000032DE	ERROR_IPSEC_MM_AUTH_PENDING_DELETION	The Main Mode authentication bundle is pending deletion.
13023	0x000032DF	ERROR_IPSEC_QM_POLICY_PENDING_DELETION	The Quick Mode policy is pending deletion.
13801	0x000035E9	ERROR_IPSEC_IKE_AUTH_FAIL	IKE authentication credentials are unacceptable.
13802	0x000035EA	ERROR_IPSEC_IKE_ATTRIB_FAIL	IKE security attributes are unacceptable.
13803	0x000035EB	ERROR_IPSEC_IKE_NEGOTIATION_PENDING	IKE Negotiation in progress.
13804	0x000035EC	ERROR_IPSEC_IKE_GENERAL_PROCESSING_ERROR	General processing error.
13805	0x000035ED	ERROR_IPSEC_IKE_TIMED_OUT	Negotiation timed out.
13806	0x000035EE	ERROR_IPSEC_IKE_NO_CERT	IKE failed to find valid machine certificate.
13807	0x000035EF	ERROR_IPSEC_IKE_SA_DELETED	IKE SA deleted by peer before establishment completed.
13808	0x000035F0	ERROR_IPSEC_IKE_SA_REAPED	IKE SA deleted before establishment completed.
13809	0x000035F1	ERROR_IPSEC_IKE_MM_ACQUIRE_DROP	Negotiation request sat in Queue too long.
13810	0x000035F2	ERROR_IPSEC_IKE_QM_ACQUIRE_DROP	Negotiation request sat in Queue too long.
13811	0x000035F3	ERROR_IPSEC_IKE_QUEUE_DROP_MM	Negotiation request sat in Queue too long.
13812	0x000035F4	ERROR_IPSEC_IKE_QUEUE_DROP_NO_MM	Negotiation request sat in Queue too long.
13813	0x000035F5	ERROR_IPSEC_IKE_DROP_NO_RESPONSE	No response from peer.
13814	0x000035F6	ERROR_IPSEC_IKE_MM_DELAY_DROP	Negotiation took too long.
13815	0x000035F7	ERROR_IPSEC_IKE_QM_DELAY_DROP	Negotiation took too long.
13816	0x000035F8	ERROR_IPSEC_IKE_ERROR	Unknown error occurred.
13817	0x000035F9	ERROR_IPSEC_IKE_CRL_FAILED	Certificate Revocation Check failed.
13818	0x000035FA	ERROR_IPSEC_IKE_INVALID_KEY_USAGE	Invalid certificate key usage.
13819	0x000035FB	ERROR_IPSEC_IKE_INVALID_CERT_TYPE	Invalid certificate type.
13820	0x000035FC	ERROR_IPSEC_IKE_NO_PRIVATE_KEY	No private key associated with machine certificate.
13822	0x000035FE	ERROR_IPSEC_IKE_DH_FAIL	Failure in Diffie-Helman computation.
13824	0x00003600	ERROR_IPSEC_IKE_INVALID_HEADER	Invalid header.
13825	0x00003601	ERROR_IPSEC_IKE_NO_POLICY	No policy configured.
13826	0x00003602	ERROR_IPSEC_IKE_INVALID_SIGNATURE	Failed to verify signature.

Error			Description
decimal	Hexadecimal	Name	
13827	0x00003603	ERROR_IPSEC_IKE_KERBEROS_ERROR	Failed to authenticate using Kerberos.
13828	0x00003604	ERROR_IPSEC_IKE_NO_PUBLIC_KEY	Peer's certificate did not have a public key.
13829	0x00003605	ERROR_IPSEC_IKE_PROCESS_ERR	Error processing error payload.
13830	0x00003606	ERROR_IPSEC_IKE_PROCESS_ERR_SA	Error processing SA payload.
13831	0x00003607	ERROR_IPSEC_IKE_PROCESS_ERR_PROP	Error processing Proposal payload.
13832	0x00003608	ERROR_IPSEC_IKE_PROCESS_ERR_TRANS	Error processing Transform payload.
13833	0x00003609	ERROR_IPSEC_IKE_PROCESS_ERR_KE	Error processing KE payload.
13834	0x0000360A	ERROR_IPSEC_IKE_PROCESS_ERR_ID	Error processing ID payload.
13835	0x0000360B	ERROR_IPSEC_IKE_PROCESS_ERR_CERT	Error processing Cert payload.
13836	0x0000360C	ERROR_IPSEC_IKE_PROCESS_ERR_CERT_REQ	Error processing Certificate Request payload.
13837	0x0000360D	ERROR_IPSEC_IKE_PROCESS_ERR_HASH	Error processing Hash payload.
13838	0x0000360E	ERROR_IPSEC_IKE_PROCESS_ERR_SIG	Error processing Signature payload.
13839	0x0000360F	ERROR_IPSEC_IKE_PROCESS_ERR_NONCE	Error processing Nonce payload.
13840	0x00003610	ERROR_IPSEC_IKE_PROCESS_ERR_NOTIFY	Error processing Notify payload.
13841	0x00003611	ERROR_IPSEC_IKE_PROCESS_ERR_DELETE	Error processing Delete Payload.
13842	0x00003612	ERROR_IPSEC_IKE_PROCESS_ERR_VENDOR	Error processing VendorId payload.
13843	0x00003613	ERROR_IPSEC_IKE_INVALID_PAYLOAD	Invalid payload received.
13844	0x00003614	ERROR_IPSEC_IKE_LOAD_SOFT_SA	Soft SA loaded.
13845	0x00003615	ERROR_IPSEC_IKE_SOFT_SA_TORN_DOWN	Soft SA torn down.
13846	0x00003616	ERROR_IPSEC_IKE_INVALID_COOKIE	Invalid cookie received..
13847	0x00003617	ERROR_IPSEC_IKE_NO_PEER_CERT	Peer failed to send valid machine certificate.
13848	0x00003618	ERROR_IPSEC_IKE_PEER_CRL_FAILED	Certification Revocation check of peer's certificate failed.
13849	0x00003619	ERROR_IPSEC_IKE_POLICY_CHANGE	New policy invalidated SAs formed with old policy.
13850	0x0000361A	ERROR_IPSEC_IKE_NO_MM_POLICY	There is no available Main Mode IKE policy.
13851	0x0000361B	ERROR_IPSEC_IKE_NOTCBPRIV	Failed to enabled TCB privilege.
13852	0x0000361C	ERROR_IPSEC_IKE_SECLOADFAIL	Failed to load SECURITY.DLL.
13853	0x0000361D	ERROR_IPSEC_IKE_FAILSSPINIT	Failed to obtain security function table dispatch address from SSPI.
13854	0x0000361E	ERROR_IPSEC_IKE_FAILQUERYSSP	Failed to query Kerberos package to obtain max token size.
13855	0x0000361F	ERROR_IPSEC_IKE_SRVACQFAIL	Failed to obtain Kerberos server credentials for ISAKMP/ERROR_IPSEC_IKE service. Kerberos authentication will not function. The most likely reason for this is lack of domain membership. This is normal if your computer is a member of a workgroup.
13856	0x00003620	ERROR_IPSEC_IKE_SRVQUERYCRED	Failed to determine SSPI principal name for ISAKMP/ERROR_IPSEC_IKE service (QueryCredentialsAttributes).
13857	0x00003621	ERROR_IPSEC_IKE_GETSPIFAIL	Failed to obtain new SPI for the inbound SA from Ipsec driver. The most common cause for this is that the driver does not have the correct filter. Check your policy to verify the filters.
13858	0x00003622	ERROR_IPSEC_IKE_INVALID_FILTER	Given filter is invalid.
13859	0x00003623	ERROR_IPSEC_IKE_OUT_OF_MEMORY	Memory allocation failed.
13860	0x00003624	ERROR_IPSEC_IKE_ADD_UPDATE_KEY_FAILED	Failed to add Security Association to IPSec Driver. The most common cause for this is if the IKE negotiation took too long to complete. If the problem persists, reduce the load on the faulting machine.
13861	0x00003625	ERROR_IPSEC_IKE_INVALID_POLICY	Invalid policy.
13862	0x00003626	ERROR_IPSEC_IKE_UNKNOWN_DOI	Invalid DOI.
13863	0x00003627	ERROR_IPSEC_IKE_INVALID_SITUATION	Invalid situation.
13864	0x00003628	ERROR_IPSEC_IKE_DH_FAILURE	Diffie-Hellman failure.
13865	0x00003629	ERROR_IPSEC_IKE_INVALID_GROUP	Invalid Diffie-Hellman group.
13866	0x0000362A	ERROR_IPSEC_IKE_ENCRYPT	Error encrypting payload.
13867	0x0000362B	ERROR_IPSEC_IKE_DECRYPT	Error decrypting payload.
13868	0x0000362C	ERROR_IPSEC_IKE_POLICY_MATCH	Policy match error.
13869	0x0000362D	ERROR_IPSEC_IKE_UNSUPPORTED_ID	Unsupported ID.

Error			Description
decimal	Hexadecimal	Name	
13870	0x0000362E	ERROR_IPSEC_IKE_INVALID_HASH	Hash verification failed.
13871	0x0000362F	ERROR_IPSEC_IKE_INVALID_HASH_ALG	Invalid hash algorithm.
13872	0x00003630	ERROR_IPSEC_IKE_INVALID_HASH_SIZE	Invalid hash size.
13873	0x00003631	ERROR_IPSEC_IKE_INVALID_ENCRYPT_ALG	Invalid encryption algorithm.
13874	0x00003632	ERROR_IPSEC_IKE_INVALID_AUTH_ALG	Invalid authentication algorithm.
13875	0x00003633	ERROR_IPSEC_IKE_INVALID_SIG	Invalid certificate signature.
13876	0x00003634	ERROR_IPSEC_IKE_LOAD_FAILED	Load failed.
13877	0x00003635	ERROR_IPSEC_IKE_RPC_DELETE	Deleted via RPC call.
13878	0x00003636	ERROR_IPSEC_IKE_BENIGN_REINIT	Temporary state created to perform reinitialization. This is not a real failure.
13879	0x00003637	ERROR_IPSEC_IKE_INVALID_RESPONDER_LIFETIME_NOTIFY	The lifetime value received in the Responder Lifetime Notify is below the Windows 2000 configured minimum value. Please fix the policy on the peer machine.
13880	0x00003638	ERROR_IPSEC_IKE_QM_LIMIT_REAP	SA reaped because QM limit was reached.
13881	0x00003639	ERROR_IPSEC_IKE_INVALID_CERT_KEYLEN	Key length in certificate is too small for configured security requirements.
13882	0x0000363A	ERROR_IPSEC_IKE_MM_LIMIT	Max number of established MM SAs to peer exceeded.
13883	0x0000363B	ERROR_IPSEC_IKE_NEGOTIATION_DISABLED	IKE received a policy that disables negotiation.
13884	0x0000363C	ERROR_IPSEC_IKE_QM_LIMIT	Reached maximum quick mode limit for the main mode. New main mode will be started.

More Information:
www.beckhoff.com/te1000

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