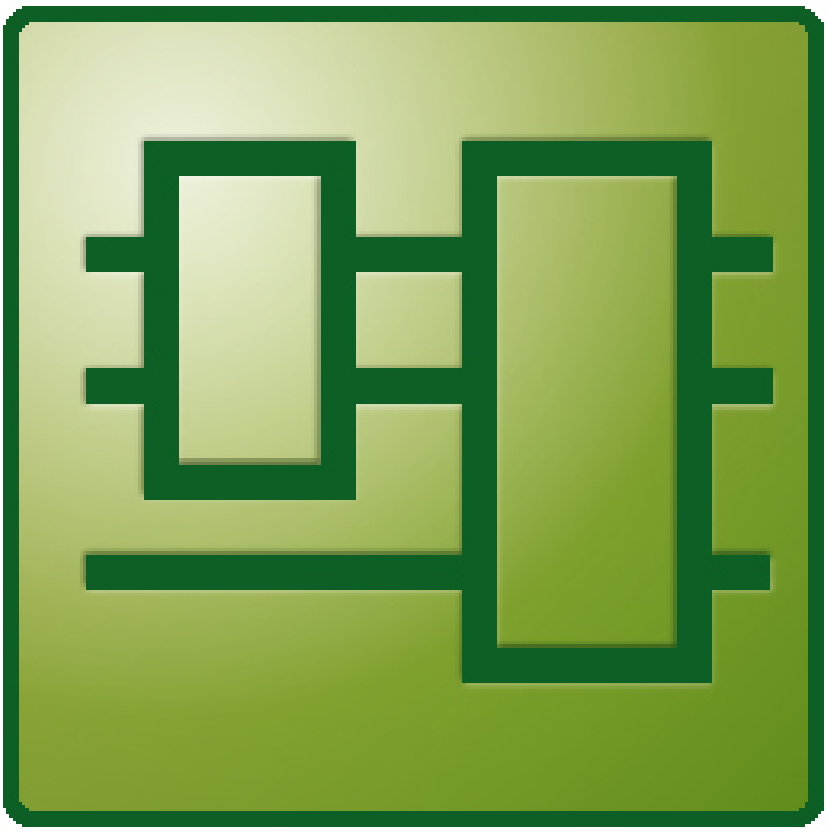


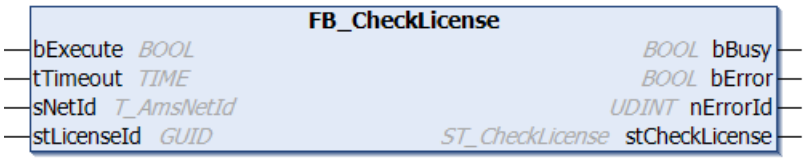
Overview new FBs in Tc2_Utilities v3.3.24

TwinCAT 3

Version: Extract
Date: 2018-06-25

BECKHOFF

1 FB_CheckLicense



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

Use case: Check license state for an OEM license

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

1.1 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

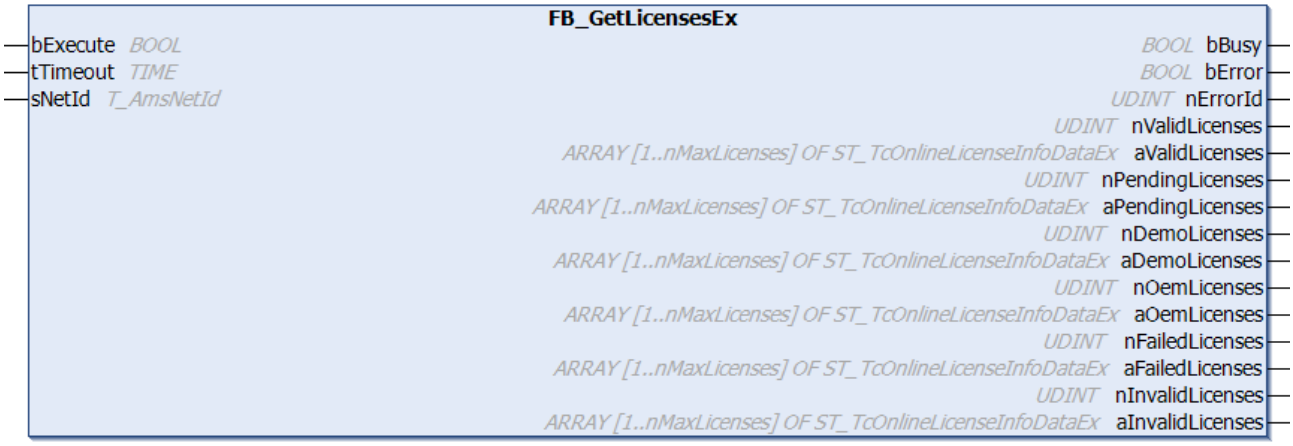
Name	Description
tExpirationTime	Expiry date
sExpirationTime	Expiry date
eResult	License status (see E_LicenseHResult [▶ 4])
nCount	Number of instances for this license (0=unlimited)

1.2 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
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

2 FB_GetLicensesEx



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

Use cases:

- Check which dongles are missing (→ array “pending licenses”)
- Get an overview of all licenses in the system

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

2.1 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)
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

Name	Description
bGenDevTypeLic	License is for the device type

3 FB_GetLicenseDongles

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

Use case: Get an overview of connected dongles and their states and addresses

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

3.1 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 [▶ 9]
nFlags	0: statically configured 1: dynamic dongle
eDongleStatus	Validation status of the license dongle (see E_LDongleStatus [▶ 9])
nSerialNo	ID number of the dongle / License Key Terminal
nReserved1	Reserved for future use
nReserved2	Reserved for future use

3.2 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

3.3 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

4 FB_GetVolumeId

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

Use case: Bind OEM application to customer-specific Beckhoff IPCs with a specific integrated Tc3 volume license

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 when the bError output is set.

stVolumeId: Supplies the Volume-System-ID as a GUID.

stSystemId: Supplies the System-ID as GUID.

Requirements

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

5 FB_GetDongleSystemId

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

Use case: Bind OEM application to customer-specific Tc3 dongle with a specific integrated Tc3 volume license

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