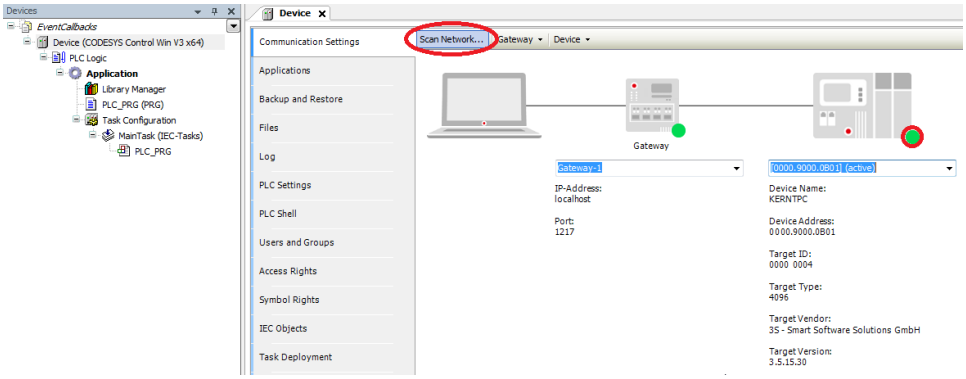
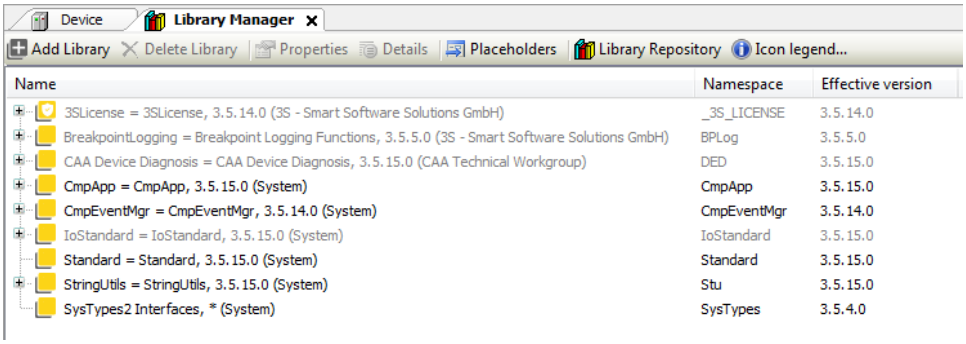
# Including an Interface (Example "ICmpEventCallback")

## Requirement

- Create a "Standard project" and select *CODESYS Control Win V3* as the device.
- Define the target system by means of the *Network scan*.

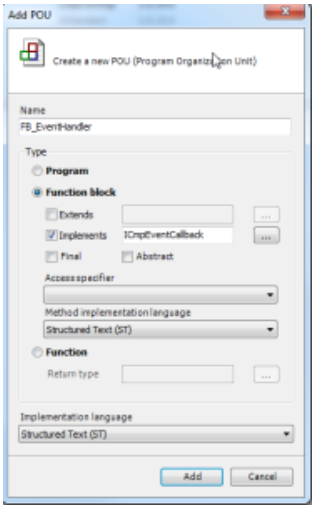



- Open the *Library Manager* and add the following libraries:
  *CmpApp*
  *CmpEventMgr*
  *StringUtils*
  *SysTypes2 Interfaces*

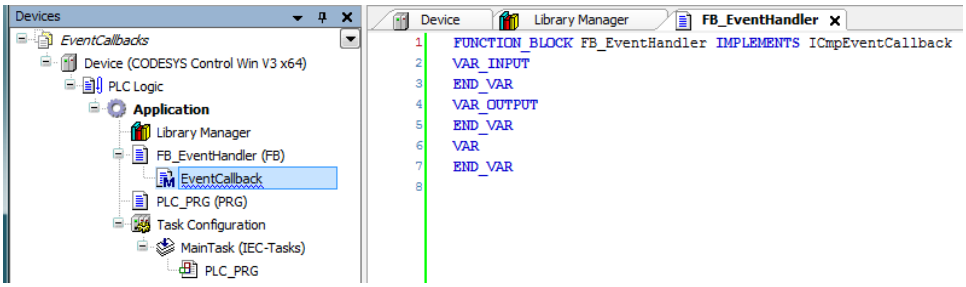



## Creating the EventHandler

- Create a new FB named *FB_EventHandler* and implement the interface *CmpEventMgr.ICmpEventCallback*.




The method *EventCallback* is created automatically with the FB:




- Add the following variables to the *FB_EventHandler*:

Declaration

```
FUNCTION_BLOCK FB_EventHandler IMPLEMENTS ICmpEventCallback
VAR_OUTPUT
   udiStartDone     : UDINT;
   udiStopDone      : UDINT;
   udiLogin         : UDINT;
END_VAR
VAR
   hEventStart      : SysTypes.RTS_IEC_HANDLE;
   hEventStop       : SysTypes.RTS_IEC_HANDLE;
   hInterfaceStart  : SysTypes.RTS_IEC_HANDLE;
   hInterfaceStop   : SysTypes.RTS_IEC_HANDLE;
   iecResult        : SysTypes.RTS_IEC_RESULT;
   itfEvtCallback   : CmpEventMgr.ICmpEventCallback;
   hEventLogin      : SysTypes.RTS_IEC_HANDLE;
   hInterfaceLogin  : SysTypes.RTS_IEC_HANDLE;
   _dwStopReason    : DWORD;
  _sAppName         : STRING;
END_VAR
```

- Adapt the method *EventCallback* as follows:

Declaration

```
METHOD EventCallback : CmpEventMgr.RTS_IEC_RESULT
VAR_INPUT
    (*Pointer to the event parameters, see Struct EventParam*)
    pEventParam : POINTER TO CmpEventMgr.EventParam;
END_VAR
VAR
    pStartParam : POINTER TO EVTPARAM_CmpApp;
    pStopParam  : POINTER TO EVTPARAM_CmpAppStop;
    pLoginParam : POINTER TO EVTPARAM_CmpAppComm;
END_VAR
```

Implementation

```
CASE pEventParam^.EventId OF
    CmpApp.EventIDs.EVT_StartDone:
        pStartParam := pEventParam^.pParameter;
        IF pStartParam <> 0 THEN
            StrCpyA(ADR(_sAppName), SIZEOF(_sAppName), ADR(pStartParam^.pApp^.szName));
        END_IF
        udiStartDone := udiStartDone + 1;
    CmpApp.EventIDs.EVT_PrepareStop:
        pStopParam := pEventParam^.pParameter;
        IF pStopParam <> 0 THEN
            _dwStopReason := pStopParam^.ulStopReason;
        END_IF
        udiStopDone := udiStopDone + 1;
    CmpApp.EventIDs.EVT_Login:
        pLoginParam := pEventParam^.pParameter;
        udiLogin := udiLogin + 1;
END_CASE
```

---

- Add the method *FB_Init* to the *FB_EventHandler* and adapt the code as follows:

---

Declaration

```
METHOD FB_Init: BOOL
VAR_INPUT
   bInitRetains: BOOL; // TRUE: the retain variables are initialized (reset warm / reset cold)
   bInCopyCode : BOOL; // TRUE: the instance will be copied to the copy code afterward (online
change)
END_VAR
```

---

Implementation

```
itfEvtCallback := This^;
hEventStart := EventOpen(CmpApp.EventIds.EVT_StartDone, CmpApp.EventIds.CMPID_CmpApp, iecResult);
hEventStop := EventOpen(CmpApp.EventIds.EVT_PrepareStop, CmpApp.EventIds.CMPID_CmpApp, iecResult);
hEventLogin := EventOpen(CmpApp.EventIds.EVT_Login , CmpApp.EventIds.CMPID_CmpApp, iecResult);

hInterfaceStart := EventRegisterCallback(hEventStart, itfEvtCallback, iecResult);
hInterfaceStop := EventRegisterCallback(hEventStop, itfEvtCallback, iecResult);
hInterfaceLogin := EventRegisterCallback(hEventLogin, itfEvtCallback, iecResult);
```

---

- Add the method *FB_Exit* to the *FB_EventHandler* and adapt the code as follows:

---

Declaration

```
METHOD FB_Exit: BOOL
VAR_INPUT
    bInCopyCode: BOOL;  // TRUE: the exit method is called in order to leave the instance which
will be copied afterwards (online change).
END_VAR
```

---

Implementation

```
EventUnregisterCallback(hEventStart, hInterfaceStart);
EventUnregisterCallback(hEventStop, hInterfaceStop);
EventUnregisterCallback(hEventLogin, hInterfaceStop);
EventClose2(hEventStart);
EventClose2(hEventStop);
EventClose2(hEventLogin);
```

## Instantiating the event handler and test the functionality

- Adapt the POU *PLC_PRG* as follows:

Declaration

```
VAR
    udiCntStart     : UDINT;
    udiCntStop      : UDINT;
    udiCntLogin     : UDINT;
    fbEventHandler  : FB_EventHandler;
END_VAR
```

Implementation

```
fbEventHandler(udiStartDone => udiCntStart, udiStopDone => udiCntStop, udiLogin => udiCntLogin);
```

- Download the project to the controller and start it.
When logging in/out and starting the project, the counters are incremented.