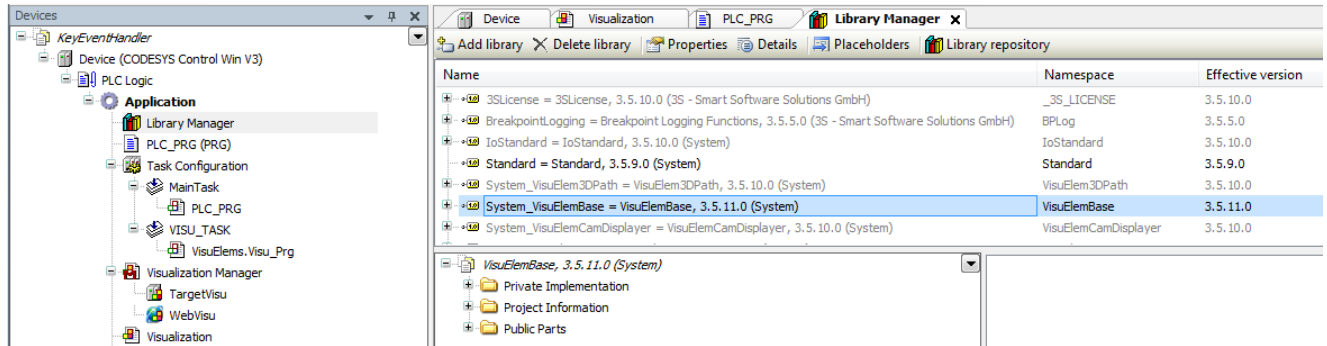


Including an Interface (Example "IKeyEventHandler")

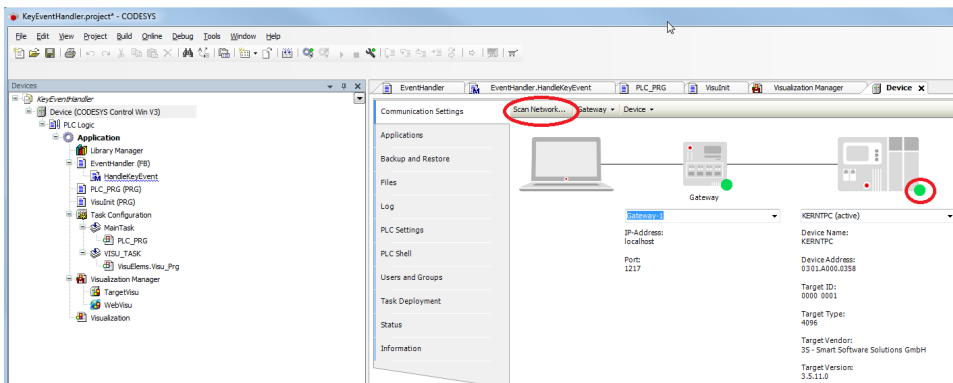
Requirement

- Create a "Standard project" and select "CODESYS ControlWin V3" as the device.
- Add a "Visualization" to the application.

A "Visualization Manger" is added to the project automatically.
Add the library "VisuElemBase" as a top-level library in the "Library Manager".

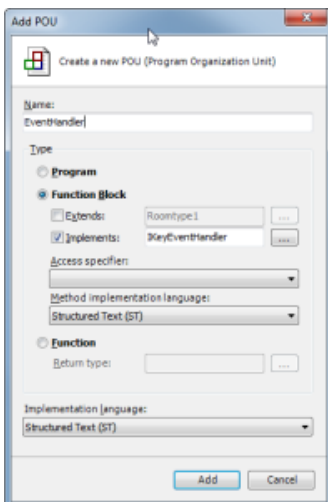


- Define the target system by means of the [Network scan](#).

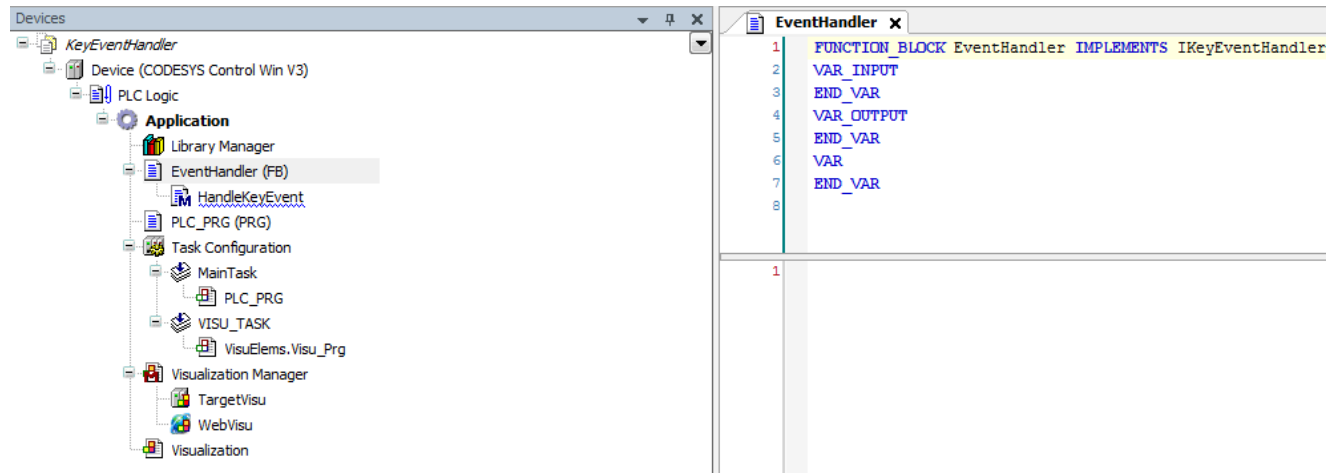


Creating the EventHandler

- Create a new FB named "EventHandler" and implement the interface "IKeyEventHandler".



The method "HandleKeyEvent" is created automatically with the FB:



- Add an output variable of type "UDINT" to the **EventHandler**:

Declaration

```
FUNCTION_BLOCK EventHandler IMPLEMENTS IKeyEventHandler
VAR_INPUT
END_VAR
VAR_OUTPUT
    udiKeyDownCount    :    UDINT;
END_VAR
VAR
END_VAR
```

- Edit the method "**HandleKeyEvent**" so that the counter is incremented only when a key is pressed:

Declaration

```
{warning 'add method implementation '}
(* This method will be called after a key event is released.
RETURN:
TRUE - When the handler has handled this event and it should not be handled by someone else
FALSE - When the event is not handled by this handler*)
METHOD HandleKeyEvent : BOOL
VAR_INPUT
    (* The event type. The value is true if a key up event was released.*)
    bKeyUpEvent    : BOOL;
    (* The key code*)
    dwKey          : DWORD;
    (* The modifiers. Possible values are:
VISU_KEYMOD_SHIFT :          DWORD := 1;
VISU_KEYMOD_ALT   :          DWORD := 2;
VISU_KEYMOD_CTRL  :          DWORD := 4;*)
    dwModifiers     : DWORD;
    (* A pointer to the client structure where the event was released*)
    pClient        : POINTER TO VisuStructClientData;
END_VAR
```

Implementation

```
IF bKeyUpEvent THEN
    THIS^.udiKeyDownCount := THIS^.udiKeyDownCount + 1;
END_IF
```

Instantiating the EventHandler

- Create an FB instance in PLC_PRG, as well as a variable for reading the current value:

Declaration

```
PROGRAM PLC_PRG
VAR
    instEvHandler    :    EventHandler;
    udiCurValue      :    UDINT;
END_VAR
```

Implementation

```
udiCurValue := instEvHandler.udiKeyDownCount;
```

Assigning the EventHandler to the visualization



Versions < V3.5.SP10

In old versions, the following approach is not possible, because the assignment of a program from the visualization manager was not possible: For these versions, an initialization must take place in the program code.

- Create a new POU of type "Program" and name it "VisuInit" for example.
- Write the following program code:

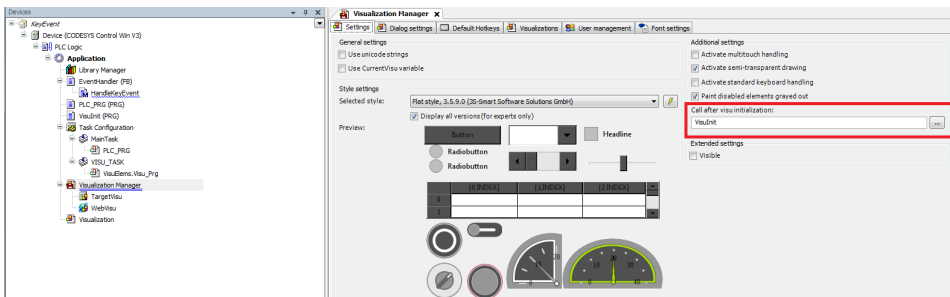
Declaration

```
PROGRAM VisuInit
VAR
END_VAR
```

Implementation

```
VisuElems.VisuElemBase.g_VisEventManager.SetKeyEventHandler(PLC_PRG.instEvHandler);
```

- Assign the program in the "Visualization Manager".



Downloading and starting the project

- Download the project to the controller and start the application.
- The visualization starts automatically.
Click the visualization window to make sure that it is the active window.

- The variable "udiCurValue" is incremented by one each time a key is pressed on the keyboard.

The screenshot displays the SIMATIC Manager interface. On the left, the 'Devices' tree shows the project structure, including 'KeyEventHandler', 'Device [connected] (CODESYS Control Win V3)', and 'PLC Logic'. The 'Application [run]' folder is expanded, showing various components like 'Library Manager', 'Event-Handler (FB)', 'HandleKeyEvent', 'PLC_PRG (PRG)', 'Visuinit (PRG)', 'Task Configuration', 'MainTask', 'PLC_PRG', 'VISU_TASK', 'VisuElem.Visu_Prg', 'Visualization Manager', 'TargetVisu', 'WebVisu', and 'Visualization'.

The main window shows the 'Device.Application.PLC_PRG' variable table and a ladder logic network.

Expression	Type	Value	Prepared value	Address	C
InstEvHandler	EventHandler				
udiCurValue	UDINT	6			

Below the table, a ladder logic network is shown. Network 1 contains the following logic:

```
1 | udiCurValue[6] := instEvHandler.udiKeyDownCount[6] | RETURN
```