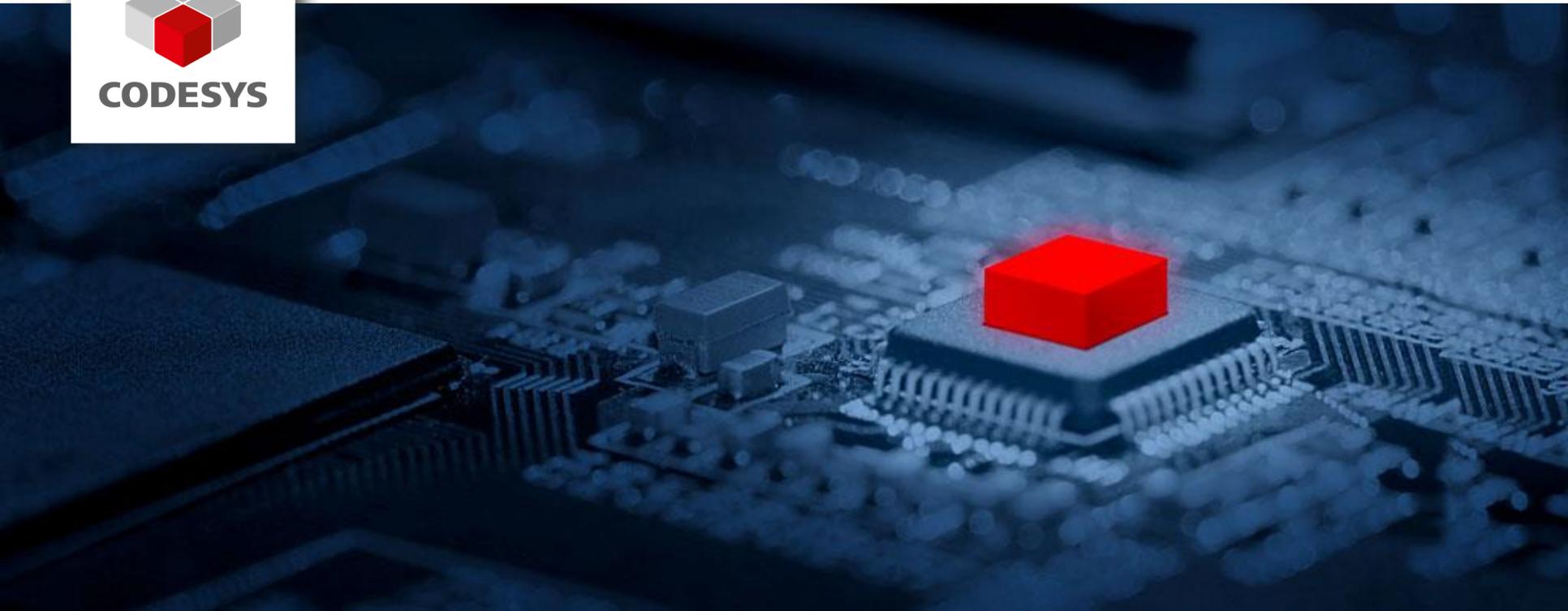




**CODESYS**



# Features & Improvements CODESYS V3.5 SP6

**1**

Runtime

**2**

Engineering

**3**

Visualization

**4**

Motion + CNC

**5**

Fieldbus

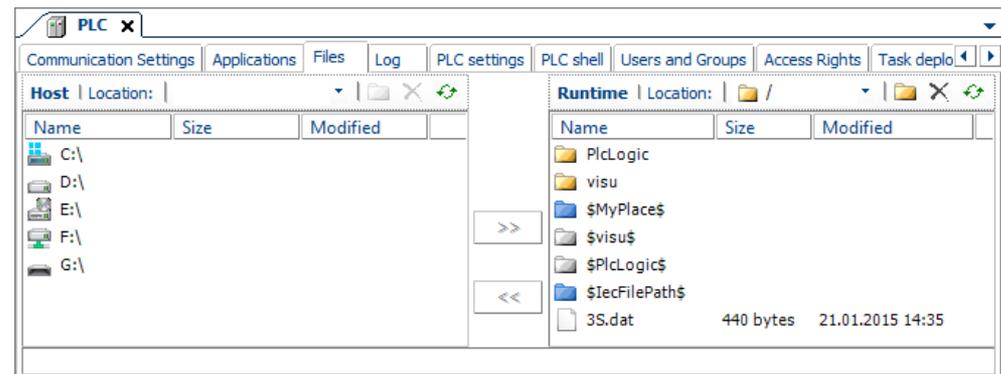
## Overview

- Symbolic paths for file system on the controller
- Structured Exception Handling (Try / Catch)
- Call stack of exceptions in external lib functions
- Secure leaving external lib functions
- Further new features



## Symbolic paths for file system on the controller

- Device manufacturer predefines general directories for files on the controller (e.g. for boot projects, visualization files, IEC files)
- Linking to different memory media possible (internal flash, external flash, USB, etc.)
- Redirection of file access out of the IEC 61131-3 application into a determined directory
- Placeholder concept:
  - \$visu\$ → ./visu
  - \$PlcLogic\$ → ./plclogic
  - \$usb\$ → /usb
- Configuration of standard directories by the device manufacturer





## Structured Exception Handling (Try / Catch)

- Preconditions:
  - CODESYS Control V3.5 SP6 implemented on the device
  - Generic implementation by the device manufacturer for embedded devices
- Prevents deadlocks due to mutually exclusive events (mutexes) / semaphores
- Exception handling e.g. in
  - called functions by system event handler (IEC callbacks, e.g. AfterReadingInputs, PrepareOnlineChange etc.)
  - (generated) IEC code coming from the context of the communication
- Basis of the `__TRY / __CATCH` operators in IEC

## Structured Exception Handling (Try / Catch)

### Example:

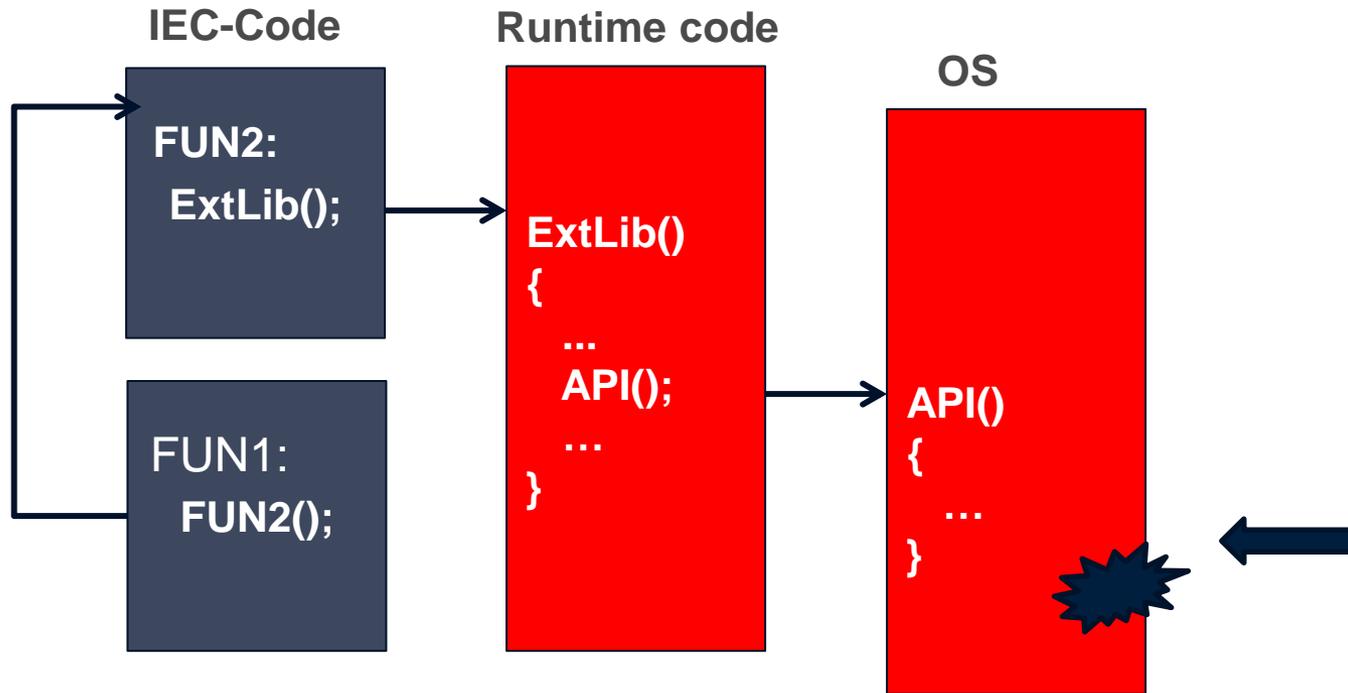
```
__TRY  
MyFunc := tryFun(count := count, testcase := g_testcase);  
__CATCH(exc)  
HandleException(exc, strExceptionText => strExceptionText);  
__FINALLY  
GVL.g_count := GVL.g_count + 2;  
__ENDTRY
```

Code, that is tried to be executed

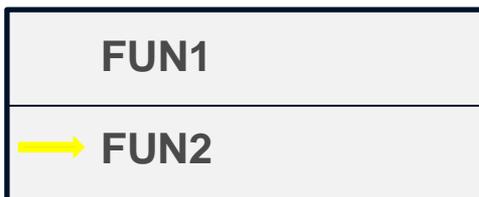
Code, that is executed at detected exception

Code, that is executed before leaving the try/catch execution

## Call stack of exceptions in external lib functions



Call stack in CODESYS:



## Call stack of exceptions in external lib functions

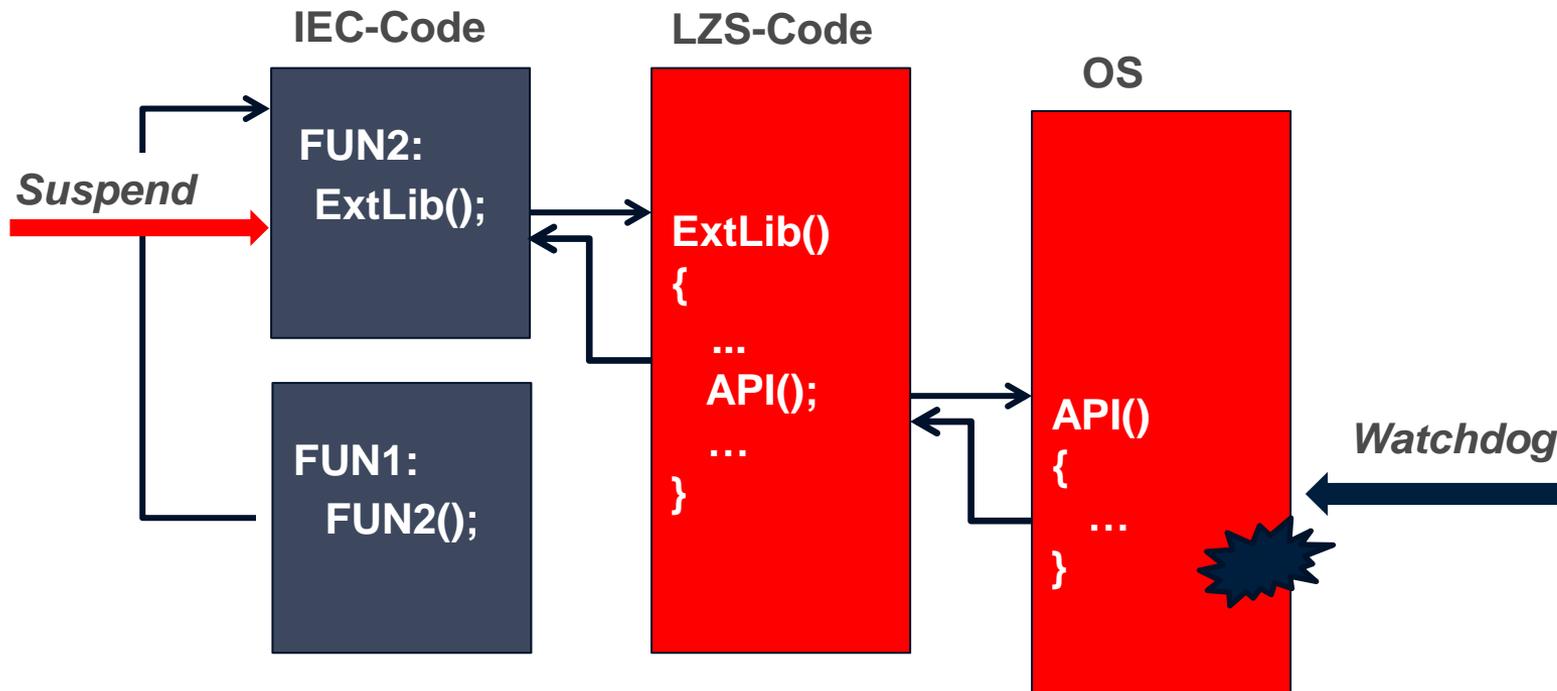
- So far:  
Call stack for an exception in an external library function (e.g. in the runtime system or operating system API) could not be determined in the IEC 61131-3 IDE, as the C call stack is not available
- New in CODESYS V3.5 SP6:  
Generic mechanism for the detection of the exit position on the stack out of the IEC 61131-3 application
- ➔ Localization of the caller and the whole call stack in the IEC 61131-3 IDE possible
- ➔ Further benefit: detection of watchdog exceptions within library function calls with longer duration (e.g. file accesses)



## Secure leaving external lib functions

- **Problem:**  
Watchdog exceptions in external lib functions may cause unpredictable states (e.g. occupied semaphores), as tasks are so far immediately suspended
- **Solution:**  
Watchdog exceptions will be served immediately, but task leaves the external lib function and will suspended afterwards
- **Precondition:**  
Detection of the caller out of the IEC code (see previous feature)

## Secure leaving external lib functions after watchdog





## Further Features

- New Plc shell command: listing of all PCI cards on the controller
- Hilscher CIFX for 64Bit Windows driver on SoftPLC CODESYS  
ControlWin for PROFINET / PROFIBUS
- Hilscher CIFX for BigEndian systems with PROFINET



1

Runtime

2

Engineering

3

Visualization

4

Motion + CNC

5

Fieldbus

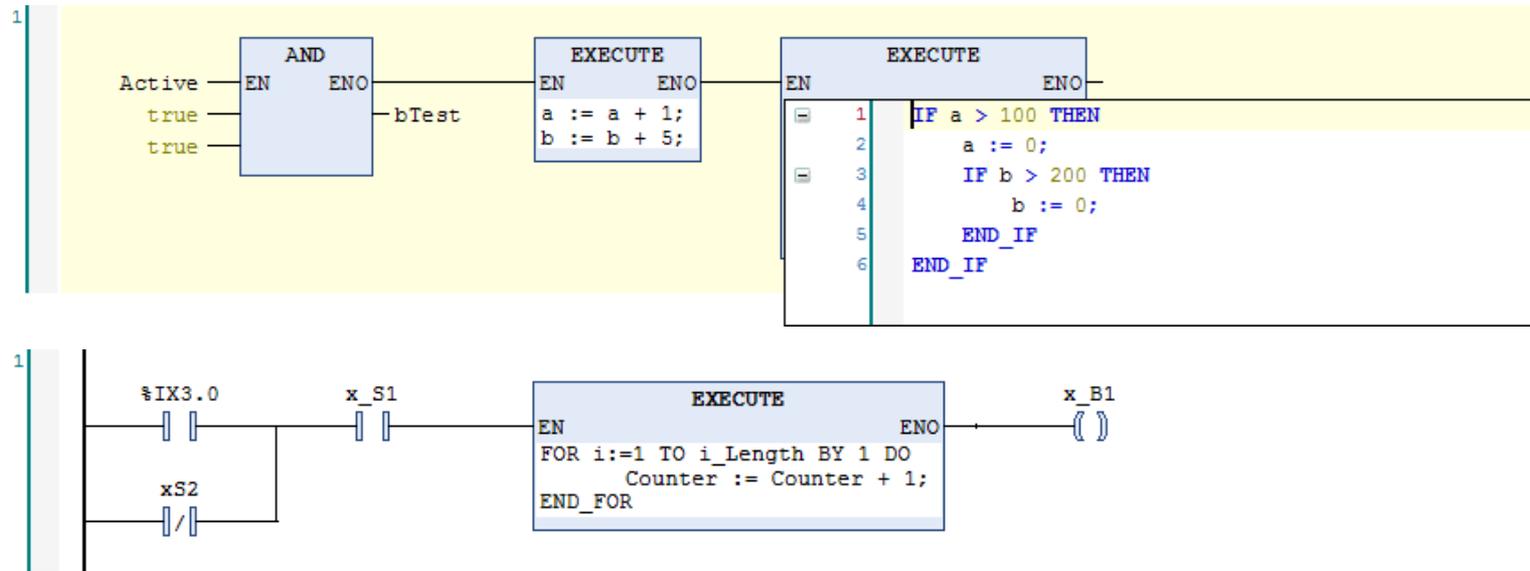
## Overview

- Inline ST code in the FBD / LD editors
- Simplified license handling on devices
- Watch list improvements
- Memory view
- CODESYS Application Composer improvements
- Further improvements



## Inline ST code in the FBD / LD editors

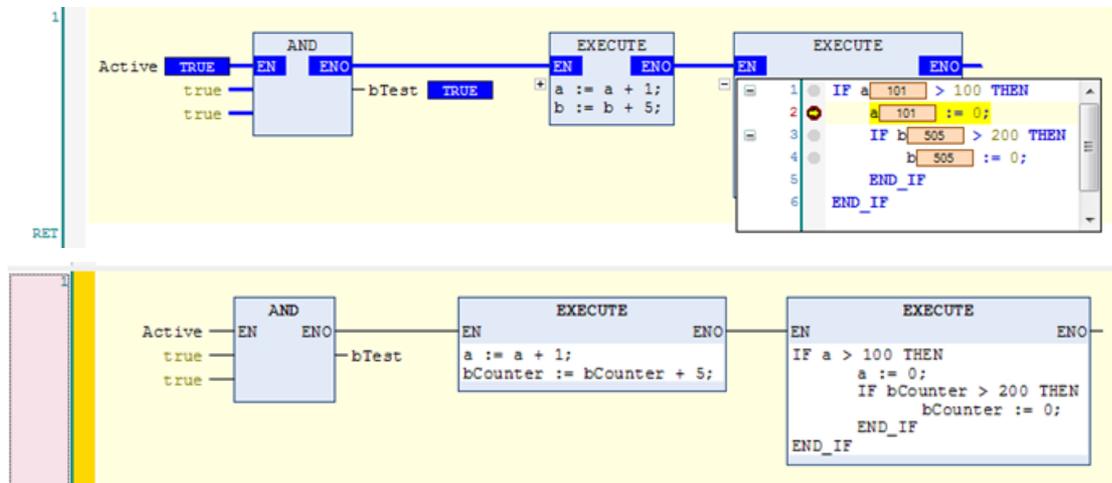
- New box EXECUTE:  
Integration of the IEC text editor in FBD / LD
- ST code is transferred into the language model of the FBD / LD editor



## Inline ST code in the FBD / LD editors

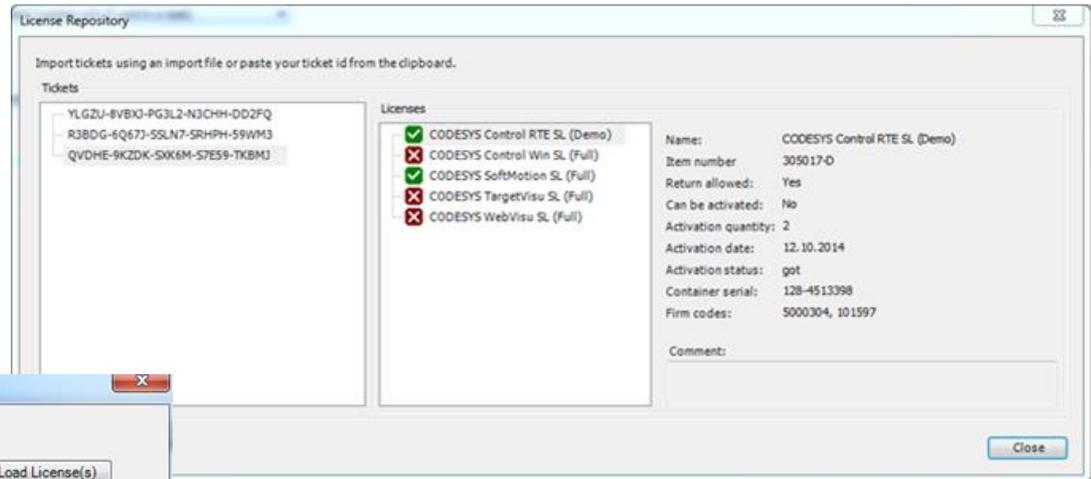
- Support of
  - Auto Declare
  - Monitoring of precompile errors
  - Online monitoring
  - Flow Control
  - Debugging (Breakpoints / Bookmarks)
  - Refactoring
  - Project compare

EXECUTE	
EN	ENO
1	IF a > 100 THEN
2	a := 0;
3	IF b > 200 THEN
4	b := 0;
5	END_IF
6	END_IF

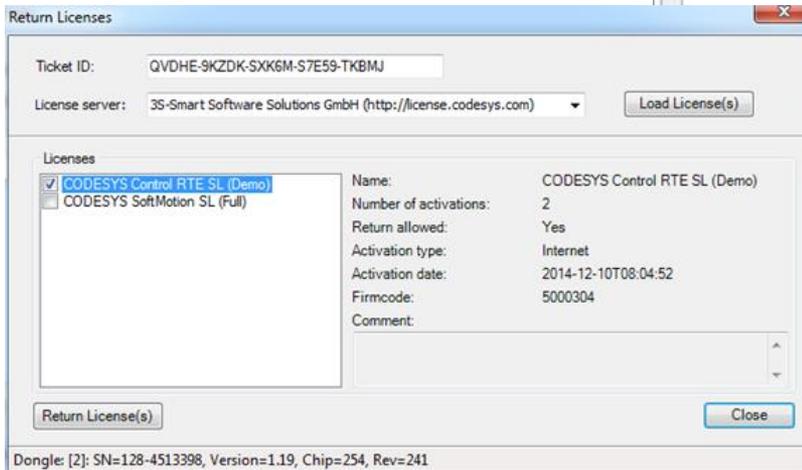


## Simplified license handling on devices

- Management of license tickets in the license repository

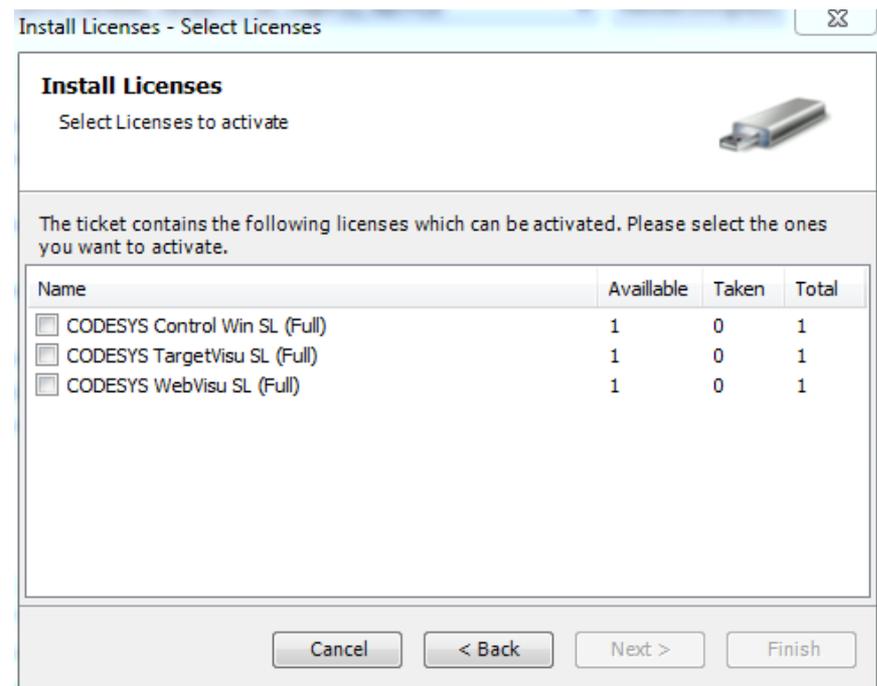
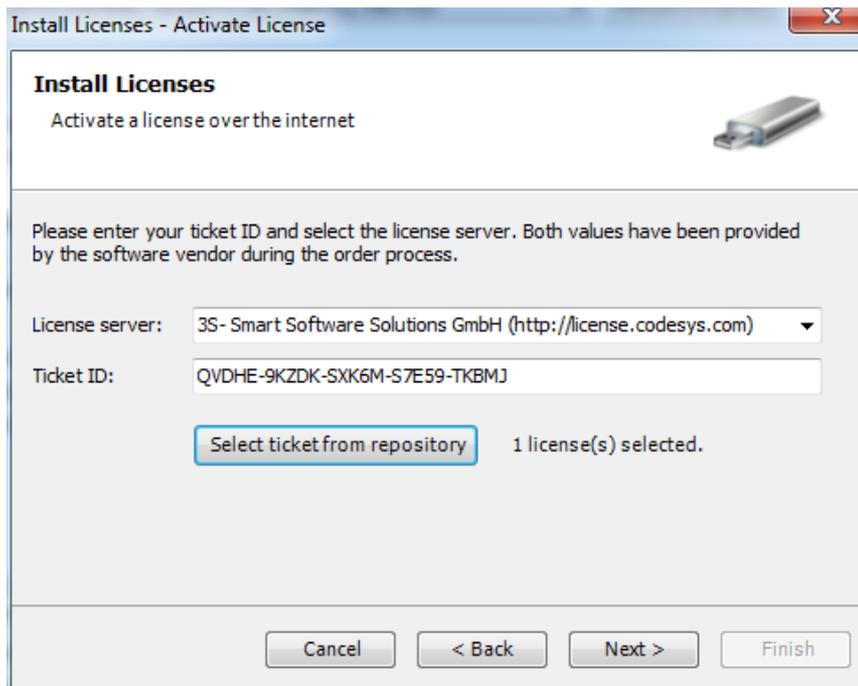


- Return licenses



## Simplified license handling on devices

- Selection of the tickets out of the license repository



## Watch list improvements

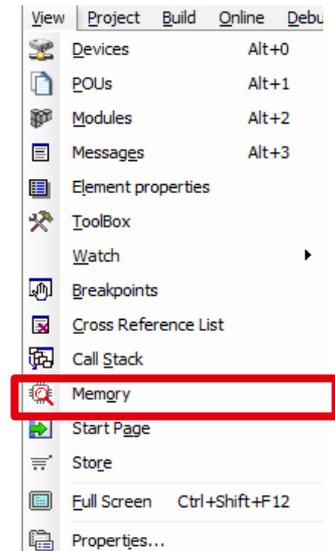
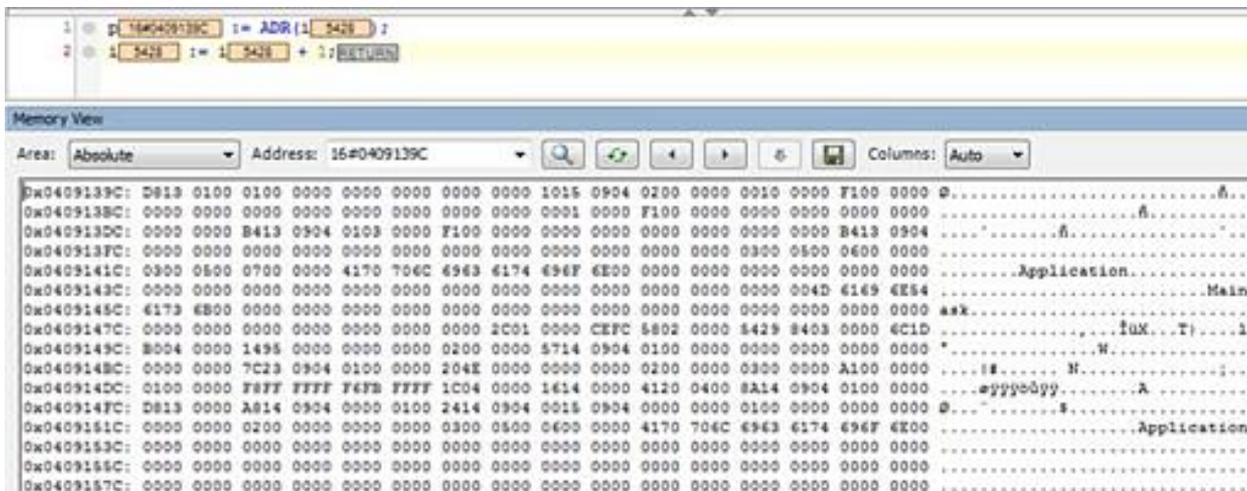
- Improved multi application handling by an additional column for the selection of device and application

Watch 1					
Expression	Application	Type	Value	Prepared value	Executionpoint
POU_FBD.b	Device.Application	INT	10		Cyclic Monitoring
POU.hugo	Device.Application2	INT	4820		Cyclic Monitoring

- Range check for integer data types during writing of values

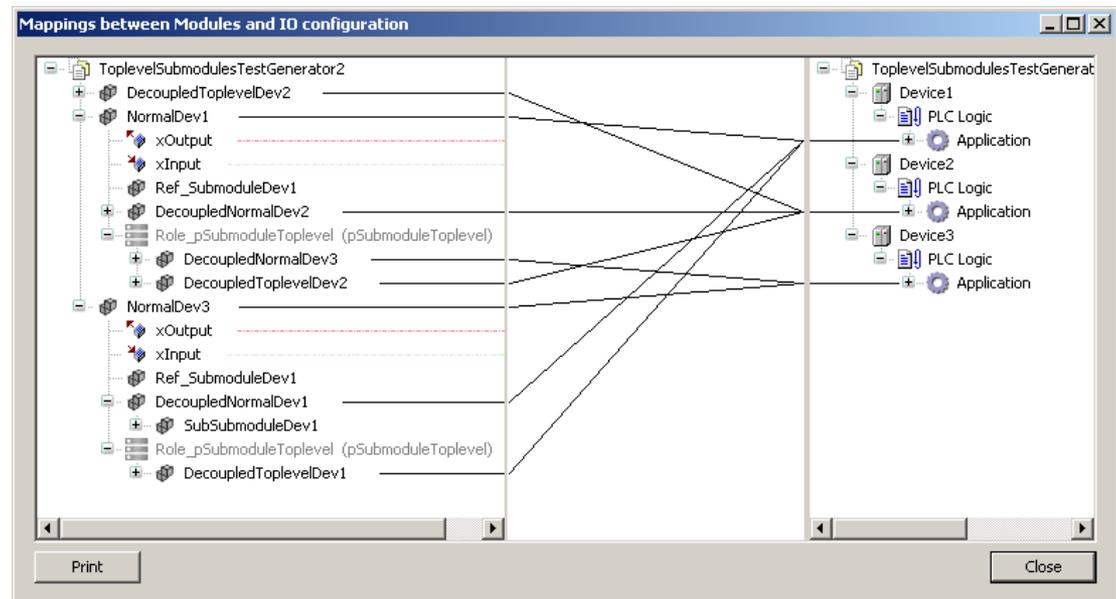
## Memory view

- Additional debugging window showing the application memory area e.g. of variables or pointers
- Hex and ASCII view on the memory
- Memory areas may be browsed and saved into files



## CODESYS Application Composer: Decoupled Modules

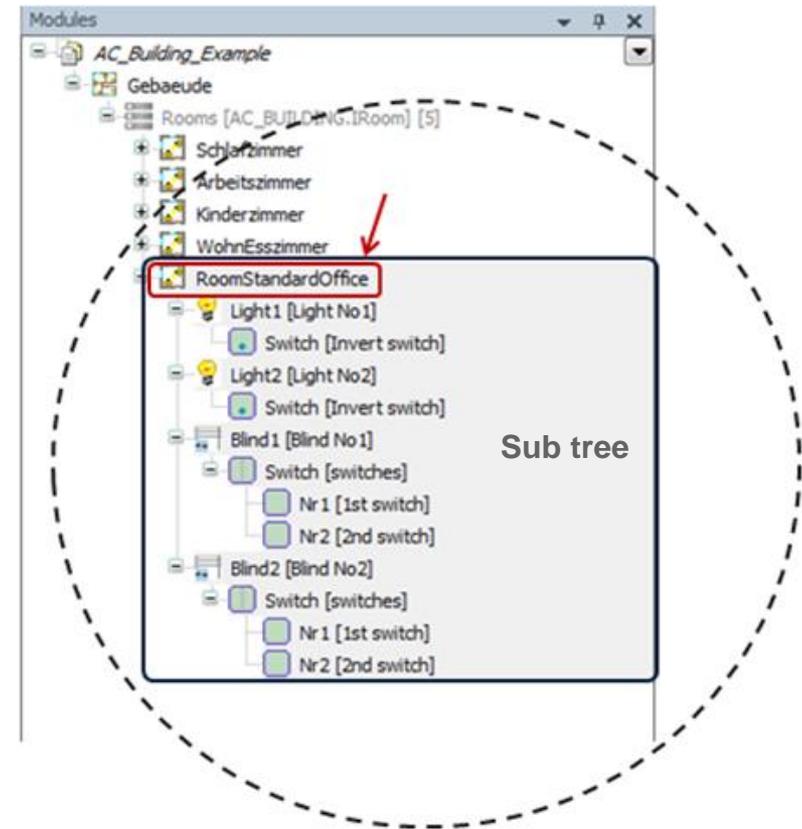
- Motivation:
  - support fine-granular deployment of modules to different PLCs
- Before decoupled modules, only top level modules could be deployed to different PLCs
- Now modules can be sub modules and be deployed to different PLCs



## CODESYS Application Composer: Macro Modules

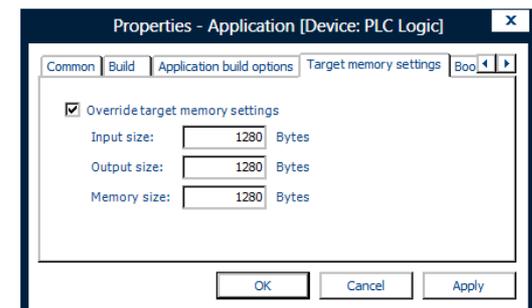
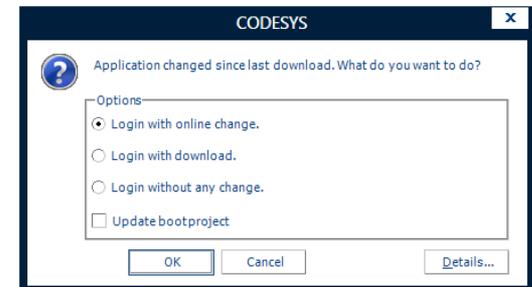
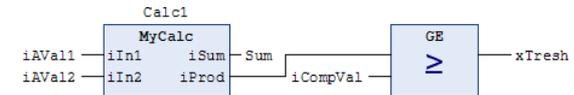
- Macro functionality for the CODESYS Application Composer
- Editor for easy creation of macro modules from sub trees
- Possibility to create macros directly using a textual description
- Macro modules are “first class”, no difference in use to regular modules
- Can be shipped in libraries

Macro module



## Further improvements

- Automatic saving before compile
- CFC/LD/FBD: Connection corrections using drag and drop
- LD/FBD: Any box input/output can now be connected for data flow
- Try – Catch (see Runtime)
- Online Change:  
update of boot project possible
- Size of output, input and marker may be configured





1

Runtime

2

Engineering

3

Visualization

4

Motion + CNC

5

Fieldbus

## Overview

- New visualization style “flat”
- New visualization element web browser
- Semitransparent elements
- Native control interface for CODESYS WebVisu
- Duplicate elements
- Further improvements



Runtime



Engineering



Visualization



Motion + CNC



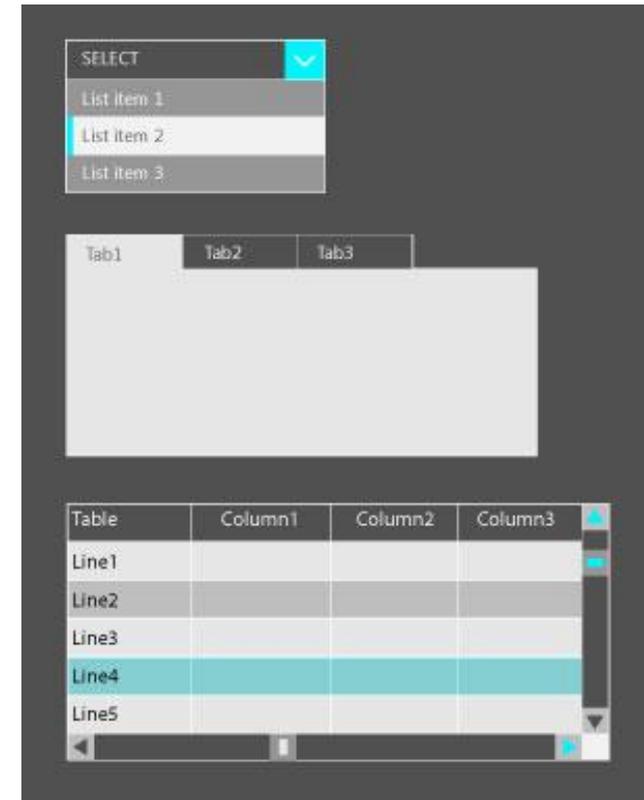
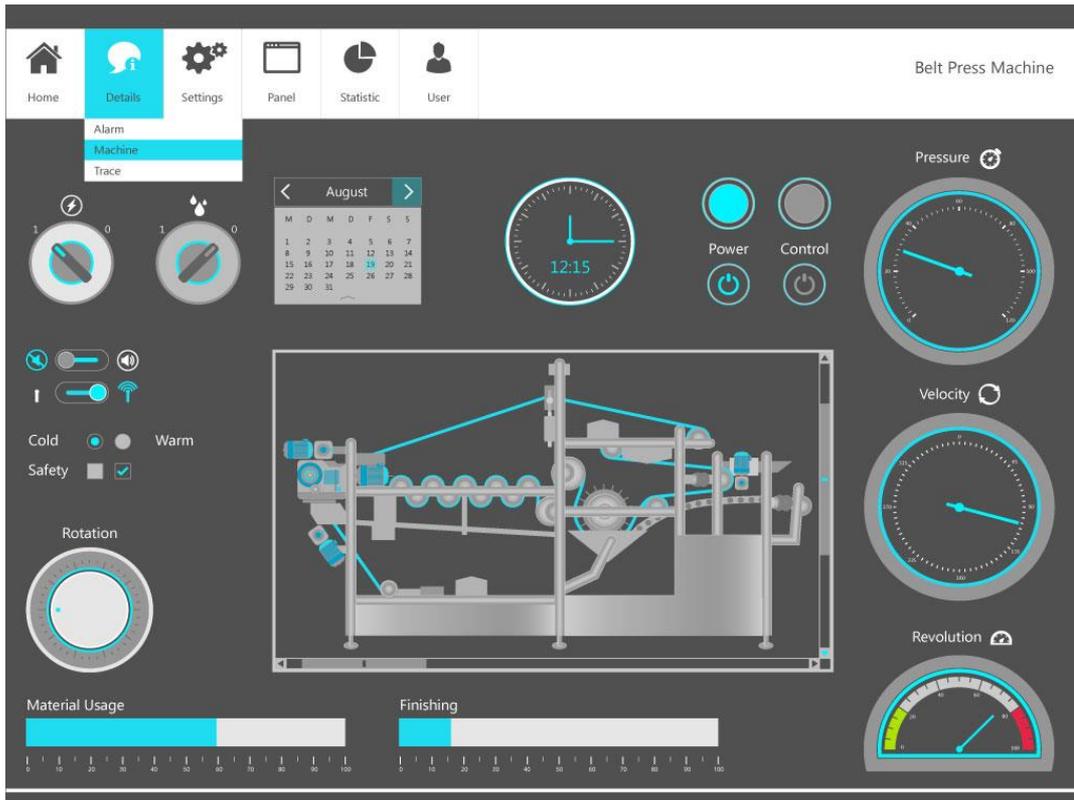
Fieldbus



Safety

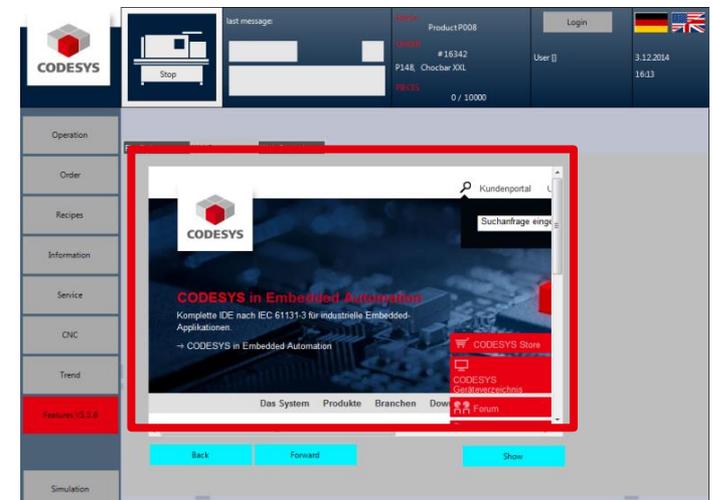
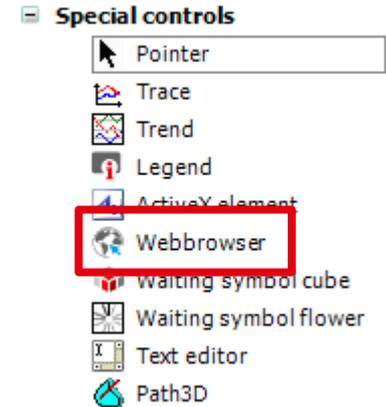
## New Visualization style “Flat”

New Look & Feel of the visualization elements by selecting a new style.



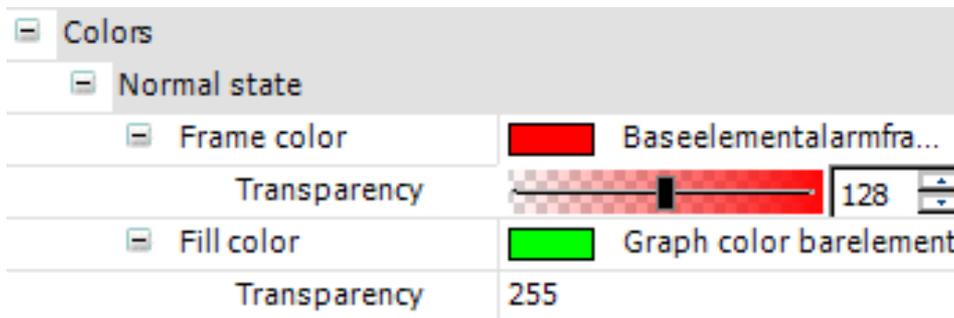
## New Visualization element web browser

- Web browser element can be used in visualization screens as often as desired
- Display e.g. of HTML help pages or camera pictures
- OS specific restrictions
  - Windows / Linux: HTML5 including camera pictures supported
  - WinCE: HTML supported
- Element has to be enabled in the device description by OEM



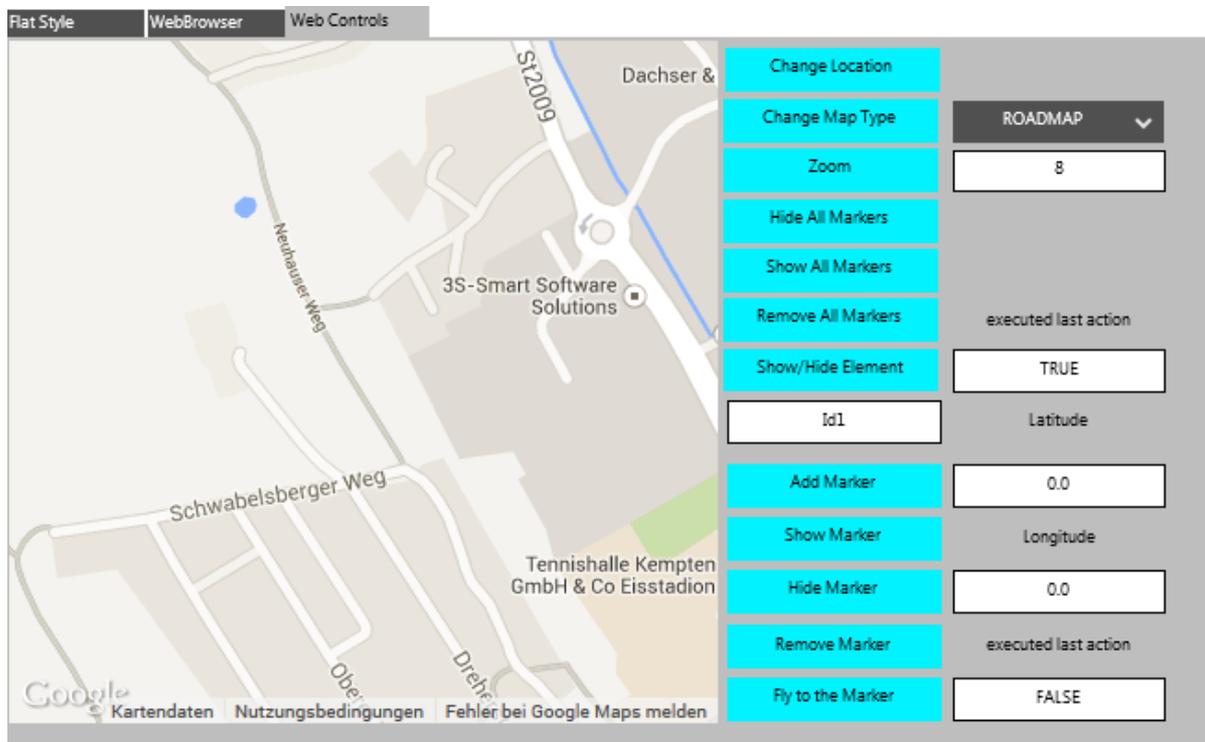
## Semitransparent elements

- Element color may be semitransparent
- Degree of transparency may be configured for all elements with color configuration
- Feature has to be enabled in the device description by the device manufacturer



## Native control interface for CODESYS WebVisu

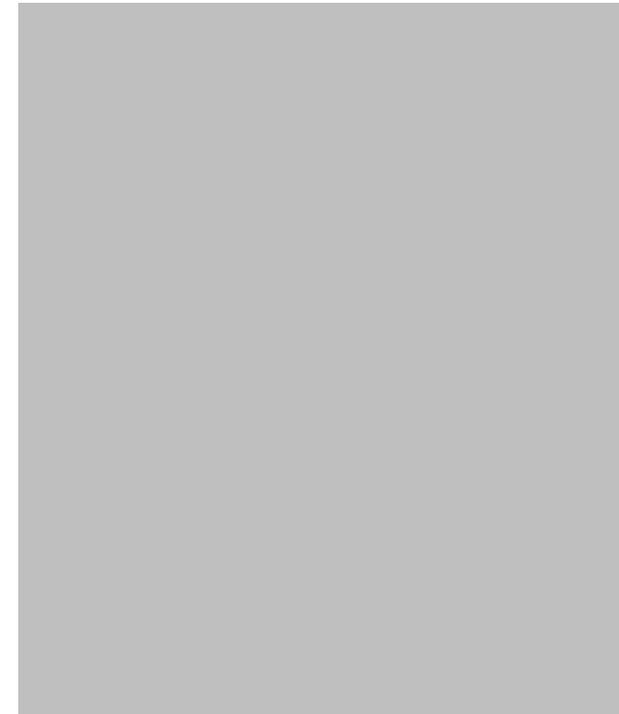
- Implementing web controls in CODESYS WebVisu
- Example:  
Implementation of Google Maps and controlling via IEC variables



The screenshot displays a web browser window with a 'Web Controls' tab active. On the left, a Google Map shows a street view of an industrial area with labels like 'Dachser & St 2009', 'Neuhauer Weg', '3S-Smart Software Solutions', 'Schwabelberger Weg', 'Tennishalle Kempten GmbH & Co Eisstadion', and 'Ober Drehe'. A blue marker is placed on the map. On the right, a control panel with cyan buttons allows for map manipulation:

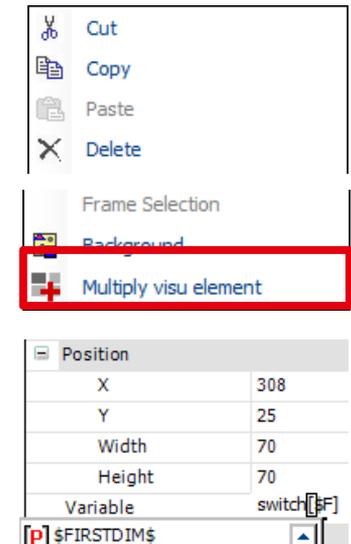
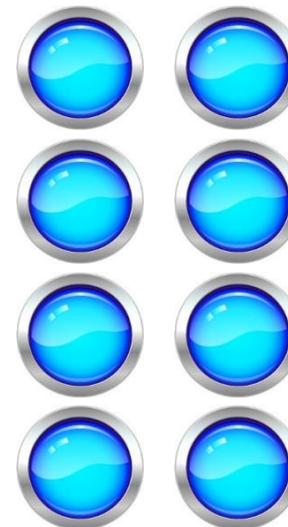
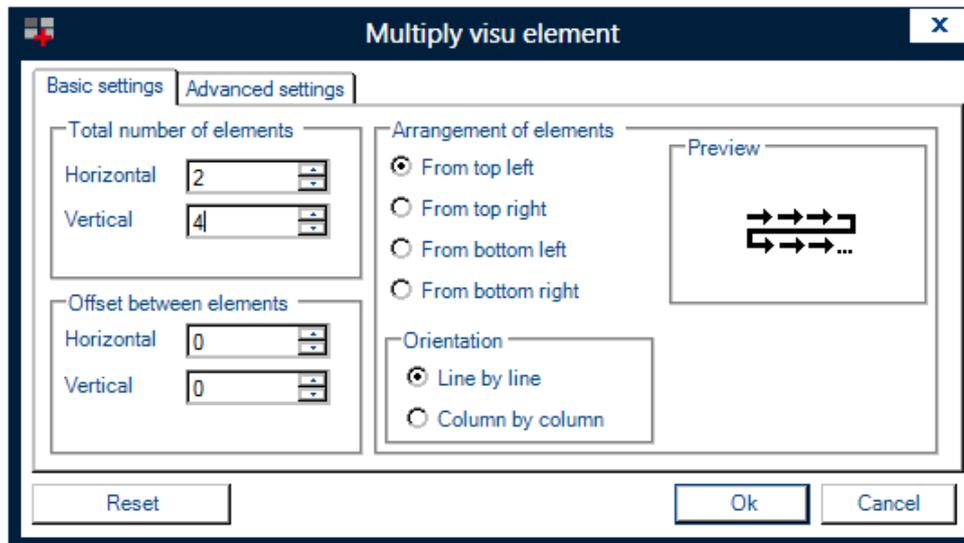
- Change Location
- Change Map Type: ROADMAP (dropdown)
- Zoom: 8 (input field)
- Hide All Markers
- Show All Markers
- Remove All Markers: executed last action
- Show/Hide Element: TRUE (input field)
- Id1: (input field)
- Latitude: (input field)
- Add Marker: 0.0 (input field)
- Show Marker: Longitude (input field)
- Hide Marker: 0.0 (input field)
- Remove Marker: executed last action
- Fly to the Marker: FALSE (input field)

At the bottom of the map, there are links for 'Google Kartendaten', 'Nutzungsbedingungen', and 'Fehler bei Google Maps melden'.



## Duplicate elements

- Duplication of the following elements possible: Rounded rectangle, rectangle, ellipse, line, polygon, polyline, Bezier, image, frame, button, pie, spin control, text field, checkbox, switches and lamps
- Comfortable generation of one or two dimensional element arrays for displaying data arrays





## Further improvements

- Secure Online Change for the visualization
  - Concurrent re-initialization of the complete visualization
  - ➔ Jitter considerably reduced
- Style preview in the visualization manager
- Hierarchical styles: styles may inherit properties from other styles
- User management visualization:  
One user may be assigned to different groups (roles)
- Multitouch for CODESYS WebVisu

1

Runtime

2

Engineering

3

Visualization

4

Motion + CNC

5

Fieldbus

## Overview

- String literals in G-Code
- Infrastructure Improvements



Runtime



Engineering



Visualization



Motion + CNC



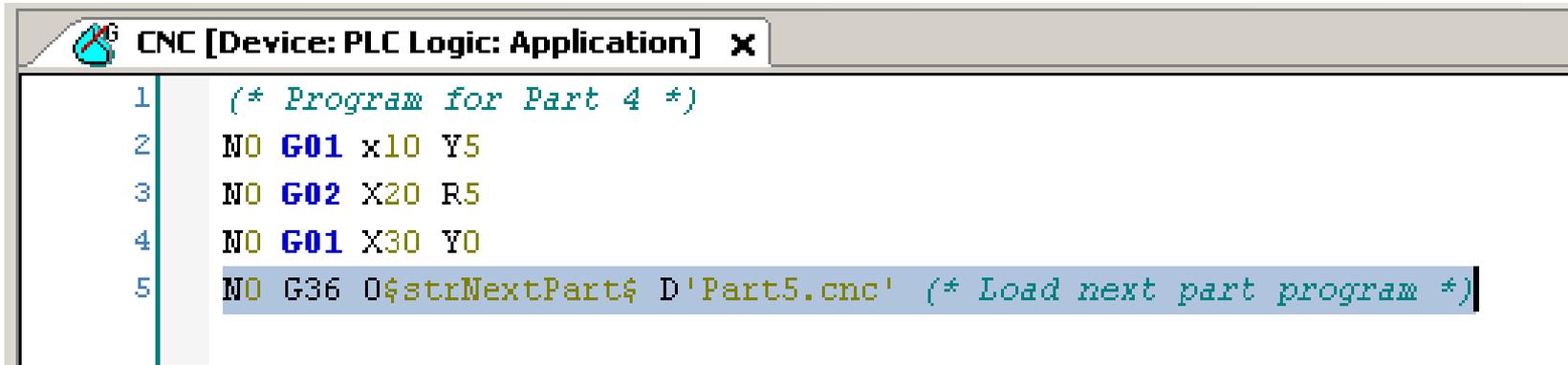
Fieldbus



Safety

## String literals in G-Code

- Allows one part program to reference the next program to load
- G36 sets a string variable, G37 appends to a string variable



```
CNC [Device: PLC Logic: Application] X
1  (* Program for Part 4 *)
2  NO G01 x10 Y5
3  NO G02 X20 R5
4  NO G01 X30 Y0
5  NO G36 0$strNextPart$ D'Part5.cnc' (* Load next part program *)
```

## Infrastructure Improvements

- Consistent logging of all kinds of errors (important for Support)
- Improved Stöber driver: EoE support, improved velocity pilot control during switch from controller mode velocity to position, additional drive variables
- New library SM3\_Debug with function blocks to read CNC queues from files and write them to files.  
Important to reproduce CNC problems that customers experience
- New parameter for generic DSP402 drives to ignore bit 12 of status word (as many drives don't set this bit correctly)

1

Runtime

2

Engineering

3

Visualization

4

Motion + CNC

5

Fieldbus

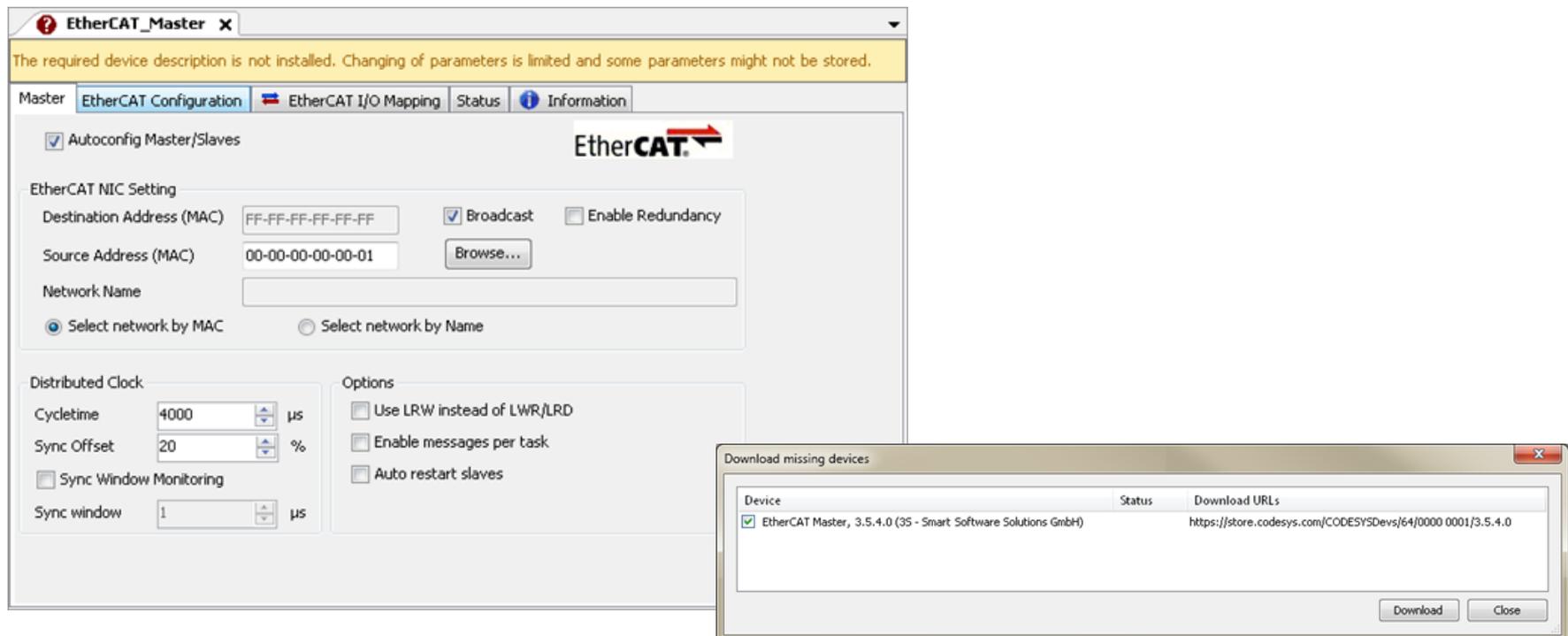
## Overview

- General improvements
- CANopen / J1939
- EtherCAT
- PROFINET
- Further improvements



## General improvements

- Device object: show information if device descriptions are not installed and therefore editing is limited
- Device Repository: download missing device descriptions from server



The screenshot shows the 'EtherCAT\_Master' configuration window. A yellow warning banner at the top states: 'The required device description is not installed. Changing of parameters is limited and some parameters might not be stored.' The 'EtherCAT Configuration' tab is active, showing settings for 'Autoconfig Master/Slaves', 'EtherCAT NIC Setting', 'Distributed Clock', and 'Options'. The 'EtherCAT NIC Setting' section includes fields for 'Destination Address (MAC)' (FF-FF-FF-FF-FF-FF), 'Source Address (MAC)' (00-00-00-00-00-01), and 'Network Name'. The 'Distributed Clock' section has 'Cycletime' set to 4000 µs and 'Sync Offset' set to 20%. The 'Options' section includes checkboxes for 'Use LRW instead of LWR/LRD', 'Enable messages per task', and 'Auto restart slaves'. A 'Download missing devices' dialog box is open in the foreground, displaying a table with the following data:

Device	Status	Download URLs
<input checked="" type="checkbox"/> EtherCAT Master, 3.5.4.0 (3S - Smart Software Solutions GmbH)		<a href="https://store.codesys.com/CODESYSDevs/64/0000 0001/3.5.4.0">https://store.codesys.com/CODESYSDevs/64/0000 0001/3.5.4.0</a>

The dialog box has 'Download' and 'Close' buttons at the bottom right.



## CANopen / J1939

- Simplified PDO Mapping editor and SDO editor
- CiA405 GetKernelState implemented
- CAN Network Scan implemented
- CANopenSlave: Adding / removing / editing objects on Object Dictionary and PDO configurator page now possible
- J1939 as a non-safe protocol stack in a Safety SIL2 environment now supported

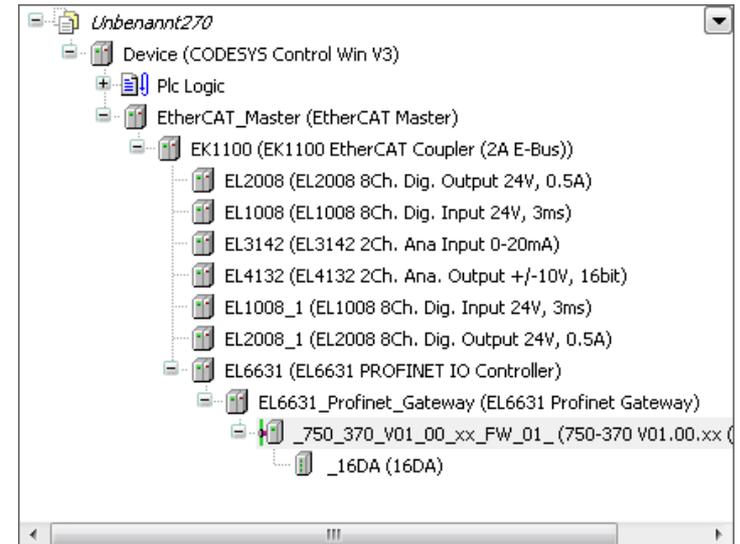
## EtherCAT

- Use station alias for changed order of devices
- EL6631 configuration now supported
- EtherCAT SDO Information service now supported

Slave Expert Process Data Process Data Startup parameters Online CoE Online EtherCAT Configuration

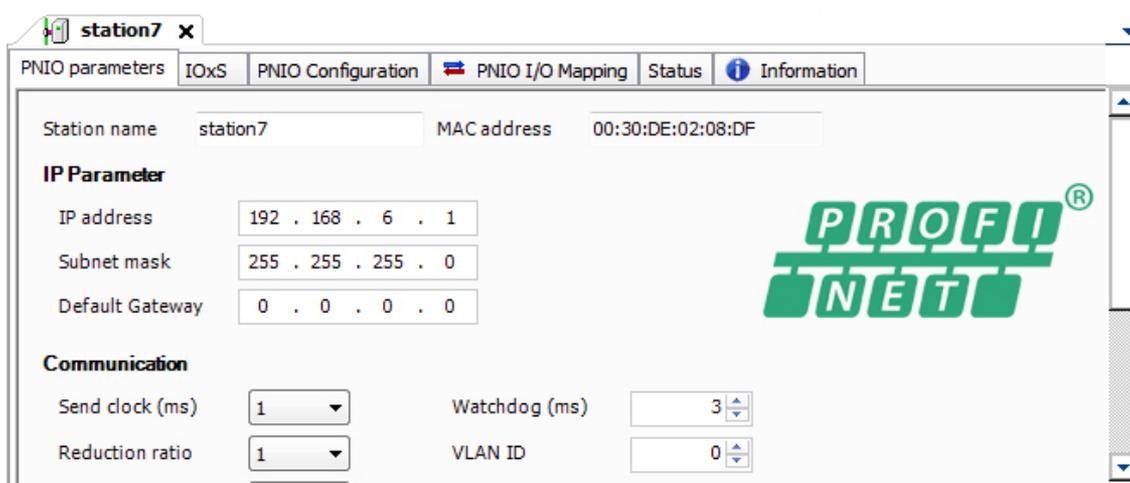
Read this page Auto Update Offline from ESI file Online from device

Index/Subindex	Name	Flags	Type	Value
16#1000:16#00	Device type	RO	UDINT	26219401
16#1008:16#00	Device name	RO	STRING(11)	'EL4132-0000'
16#1009:16#00	Hardware version	RO	STRING(2)	'13'
16#100A:16#00	Software version	RO	STRING(2)	'08'
16#1011:16#00	Restore default parameters	RO	USINT	1
16#1018:16#00	Identity	RO	USINT	4
16#10F0:16#00	Backup parameter handling	RO	USINT	1
16#1400:16#00	RxPDO 01 mapping	RO	USINT	6
16#1401:16#00	RxPDO 02 mapping	RO	USINT	6
16#1402:16#00	AO RxPDO-Par Ch.1	RO	USINT	6
16#1403:16#00	AO RxPDO-Par Ch.2	RO	USINT	6
16#1410:16#00	RxPDO 017 mapping	RO	USINT	6
16#1600:16#00	RxPDO 01 mapping	RO	USINT	1
16#1601:16#00	RxPDO 02 mapping	RO	USINT	1
16#1602:16#00	AO RxPDO-Map Ch.1	RO	USINT	1
16#1603:16#00	AO RxPDO-Map Ch.2	RO	USINT	1



## PROFINET

- PROFINET I/O Master Stack as CODESYS library now available
  - ➔ Using PROFINET with standard Ethernet ports now without additional hardware components possible
- Integrated PROFINET configurator
  - Suitable for the configuration of the CODESYS PROFINET I/O Master Stack
  - IRT Scheduling integrated





## Further improvements

- EtherNet/IP
  - Specification of the structure of I/O image now possible
- Modbus
  - TCP Function Block for Modbus client functionality
  - Support of communication settings of slaves at runtime



Inspiring Automation Solutions

Thank you for your attention.