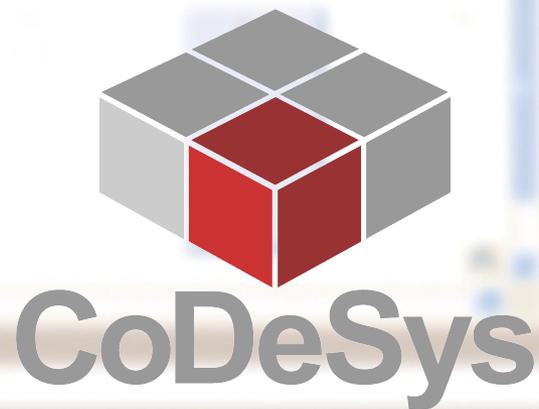




Features and improvements V3.4 SP3



1. Engineering: Batch Mode/Scripting
2. Engineering: PLCopen XML
3. Engineering: User interface and editors
4. Compiler: Online change
5. Runtime: Transfer of system config as XML file
6. Runtime: Run from flash for embedded systems
7. Runtime: CoDeSys message box
8. Runtime: Operation control
9. Visualization: ActiveX control element
10. Visualization: Visu element „histogram“ available
11. Fieldbus technology: Device update during opening of a project
12. Fieldbus technology: Ethernet over EtherCAT (EoE)
13. Motion: New motion drivers
14. Motion: CAM editor



- CoDeSys V3 now includes a scripting language based on Python.
- This means you have the same functionality as in CoDeSys V2.3 plus the advantages of a real programming language combined with the vast possibilities of the Python standard library.
- Applications:
 - Automatic import of project parts
 - Automatic compilation and download of PLC applications

Advantages

- Easy syntax
 - Existing CoDeSys V2.3 batches can easily be converted
- Support of complex program structures
 - structured, object-oriented and functional programming is possible
 - comprehensive scripts and script libraries can be realized
- Widespread use
- Link for more information: <http://www.python.org/>



Comparison between V2 and V3

- CoDeSys V2.3 batch mode – very common - but
 - No standard
 - No variables
 - No control structures (except of subroutines)
 - No synchronous online functions
- CoDeSys V3
 - Python as scripting language
 - Powerful programming structures (variables, loops)
 - Comprehensive API functions
 - Access to CoDeSys objects
 - Optional – synchronous online functions
 - Python standard library (file access etc.)



System

PLC environment:
Trace
Activate user interface

User requests

Wait

Project

Open, save, close

Search objects and list

Print, export and import

Delete, rename and
move

Properties

LibMan

List of libs

Add or
delete
libraries

Device

Add, insert
or plug
device

Device
properties:
Communi-
cation
Simulation

Online

Login, logout,
start, stop

Download,
Boot application

Read, write and force
variables

Status query:
PLC status
Communication abort



- Via task menu



- Via command line

```
CoDeSys --profile="CoDeSys V3.4 SP3" --runscript="path_to_sample_script.py"  
[--enablescriptracing] [--noUI]
```

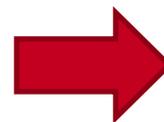
File generation of a fieldbus configuration based on a description file

```

Master (EtherCatMaster)
  KopplerLinks (EK1100)
    outventile (EL2008)
    InLichtschranke (EL1004)
  KopplerRechts (EK1100)
    outwarnlicht (EL2008)
    InInduktionsschalter (EL1004)
  
```

```

print "Current project."
proj = projects.primary
for dev in proj.get_children():
    if dev.is_device():
        f = open("C:/Users/d.hess/Desktop/Konfiguration.txt", "r")
        for line in f:
            tok = line.split(' ')
            print tok[1], tok[0]
            if "(EtherCatMaster)" in tok[1]:
                dev.add(tok[0], DeviceID(64, "0000 0001", "3.4.2.0"))
                subnodes = dev.get_children()
                master = subnodes[len(subnodes) - 1]
                elif "(EK1100)" in tok[1]:
                    master.add(tok[0], DeviceID(65, "2_044C2C5200110000",
"Revision=16#00110000"))
                    subnodes = master.get_children()
                    coupler = subnodes[len(subnodes) - 1]
                    elif "(EL2008)" in tok[1]:
                        coupler.add(tok[0], DeviceID(65, "2_07D8305200100000",
"Revision=16#00100000"))
                    elif "(EL1004)" in tok[1]:
                        coupler.add(tok[0], DeviceID(65, "2_03EC305227080000",
"Revision=16#27080000"))
            print "script finished."
  
```



Definition of PLCopen XML

- Exchange of function blocks, libraries and projects between different tools, e.g.
 - Debugging Tools
 - Simulators
 - Documentation tools
 - Modelling tools
- Transport without losing information
 - Logical information
 - Graphical information
 - Manufacturer specific information
- Source of the XML codes unimportant
 - XML description is complete
 - Filtering of the required information via the importing tool



A “native“ import/export in CoDeSys already exists, but

- it is hardly human readable
- it includes constraints, which have to be observed
- it is only “documented“ by the export
- it can be different from version to version

... on the other hand it is always...

- complete

... but then again not suitable for all applications.

```
<Single Name="TypeGuid" Type="System.Guid">6f9dac99-8de1-4efc-8465-68ac443b7d08</Single>
  <Array Name="EmbeddedTypeGuids" Type="System.Guid">
    <Single Type="System.Guid">a9ed5b7e-75c5-4651-af16-d2c27e98cb94</Single>
    <Single Type="System.Guid">3b83b776-fb25-43b8-99f2-3c507c9143fc</Single>
  </Array>
  <Single Name="Timestamp" Type="long">633439547036984297</Single>
</Single>
<Single Name="Object" Type="{6f9dac99-8de1-4efc-8465-68ac443b7d08}"
Method="IArchivable">
  <Single Name="SpecialFunc" Type="{0db3d7bb-cde0-4416-9a7b-ce49a0124323}">None</Single>
  <Single Name="Implementation" Type="{3b83b776-fb25-43b8-99f2-3c507c9143fc}" Method="IArchivable">
    <Single Name="TextDocument" Type="{f3878285-8e4f-490b-bb1b-9acbb7eb04db}" Method="IArchivable">
      <Array Name="TextLines" Type="{a5de0b0b-1cb5-4913-ac21-9d70293ec00d}">
        <Single Type="{a5de0b0b-1cb5-4913-ac21-9d70293ec00d}"
Method="IArchivable">
          <Single Name="Id" Type="long">22</Single>
```



Therefore we now have a new PLCopen XML import/export which is ...

- standardized
- human readable
- defined for text and graphical languages
- with CoDeSys specific enhancements (e.g. devices, interfaces, methods)
- also available in other IEC 61131-3 tools

... and even though it is...

- not really complete

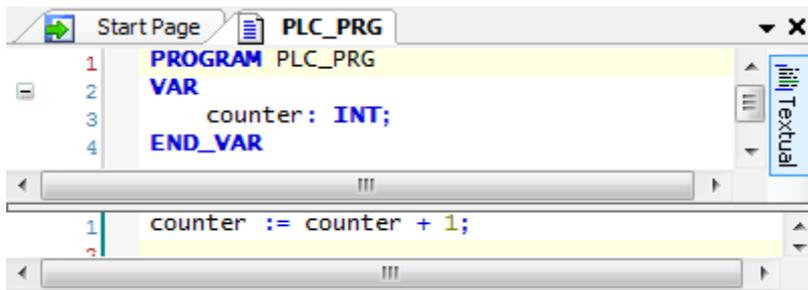
... it is suitable for almost all applications.

Applications

- Exchange format between IEC programming tools
 - Data exchange between different development platforms
 - Parallel usage of different programming environments
- Interface for “suppliers“ of graphical or logical information
 - From higher-level engineering tools, generating IEC code / data
 - Possibility to keep the connection to some elements
- Interface for “consumers“ of graphical or logical information
 - Examples: Validation tools, compilers, SCADA/HMI tools, documentation tools, translation tools
 - Filters the required information from the complete XML file
- Distribution format for function block libraries



Example of a PLCopen XML export

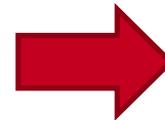


```

1 PROGRAM PLC_PRG
2 VAR
3   counter: INT;
4 END_VAR

1 counter := counter + 1;

```



```

<pou name="PLC_PRG" pouType="program">
  <interface>
    <localVars>
      <variable name="counter">
        <type>
          <INT />
        </type>
      </variable>
    </localVars>
  </interface>
  <body>
    <ST>
      <xhtml xmlns="http://www.w3.org/1999/xhtml">counter := counter + 1;
    </xhtml>
    </ST>
  </body>
</pou>

```

IEC program: Simple counter

PLCopen XML file

- Realized object types:
 - POU (incl. object-oriented enhancements)
 - Interfaces (CoDeSys-specific enhancement)
 - Actions
 - Methods (CoDeSys-specific enhancement)
 - Properties (CoDeSys-specific enhancement)
 - Transitions (CoDeSys-specific enhancement)
 - Global variables
 - Data types
 - **Tasks (incl. Unions)**
 - Devices (not complete, CoDeSys-specific enhancement)
 - Applications (not complete, CoDeSys-specific enhancement)
 - **Project information (CoDeSys-specific enhancement, only export)**
- Realized programming languages:
 - ST
 - **FBD**
 - **CFC**



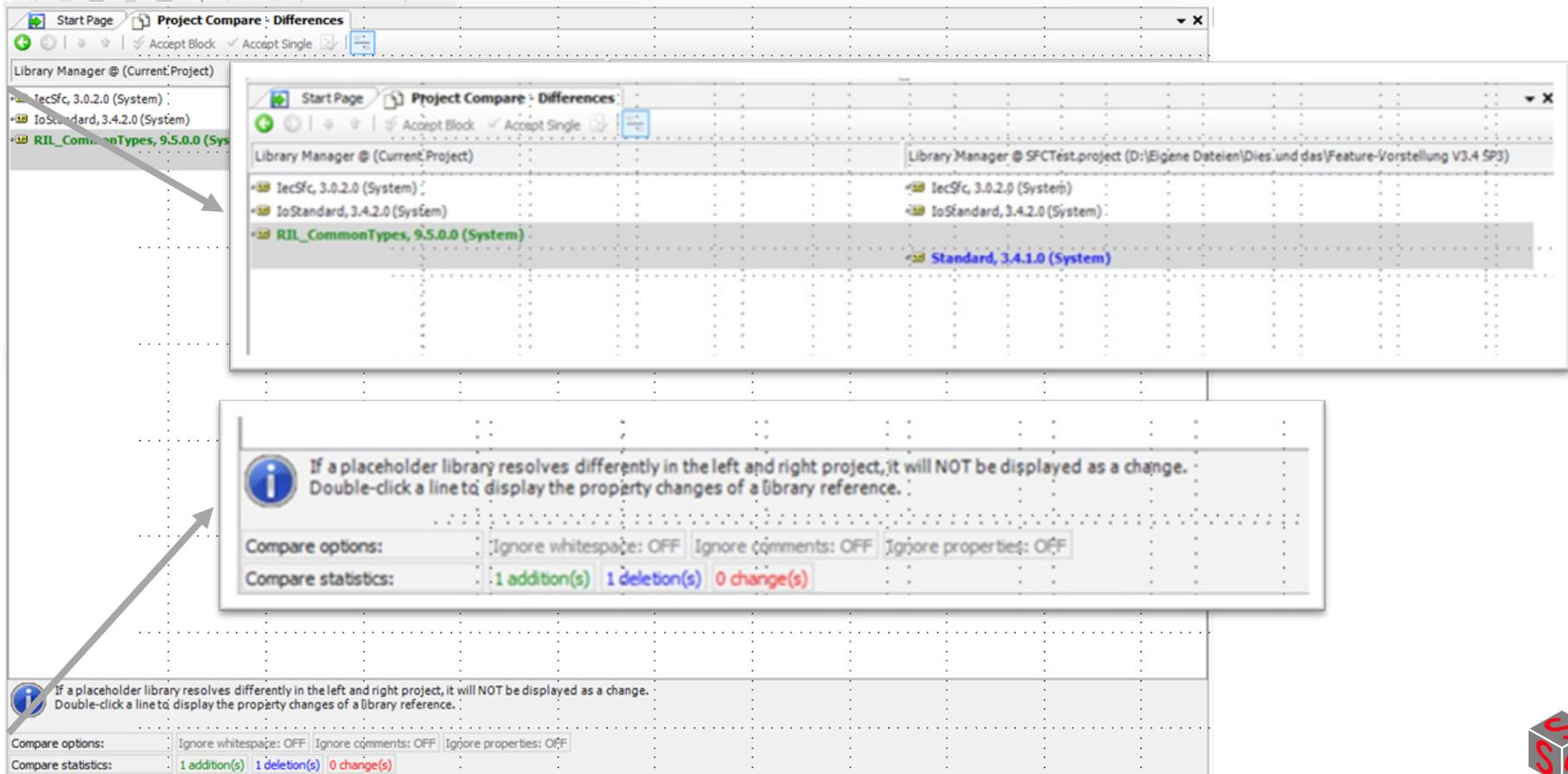
PLCopen XML - comparison

	“Native” XML	PLCopenXML
Human readable?	no	yes
Usable for external tools?	yes, only with automation platform	yes
Loss free?	yes	no
Complete?	yes	Enhancements in the project format are not directly available in PLCopen XML

PLCopenXML is not suitable for an exact storage of a CoDeSys project.



Project comparison: Graphical diff editor for library manager



Project Compare : Differences

Library Manager @ (Current Project)

- IecSfc, 3.0.2.0 (System)
- IoStandard, 3.4.2.0 (System)
- RIL_CommonTypes, 9.5.0.0 (System)

Project Compare : Differences

Library Manager @ (Current Project) | Library Manager @ SFCTest.project (D:\Eigene Dateien\Dies und das\Feature-Vorstellung V3.4 SP3)

- IecSfc, 3.0.2.0 (System)
- IoStandard, 3.4.2.0 (System)
- Standard, 3.4.1.0 (System)

Information: If a placeholder library resolves differently in the left and right project, it will NOT be displayed as a change. Double-click a line to display the property changes of a library reference.

Compare options: Ignore whitespace: OFF | Ignore comments: OFF | Ignore properties: OFF

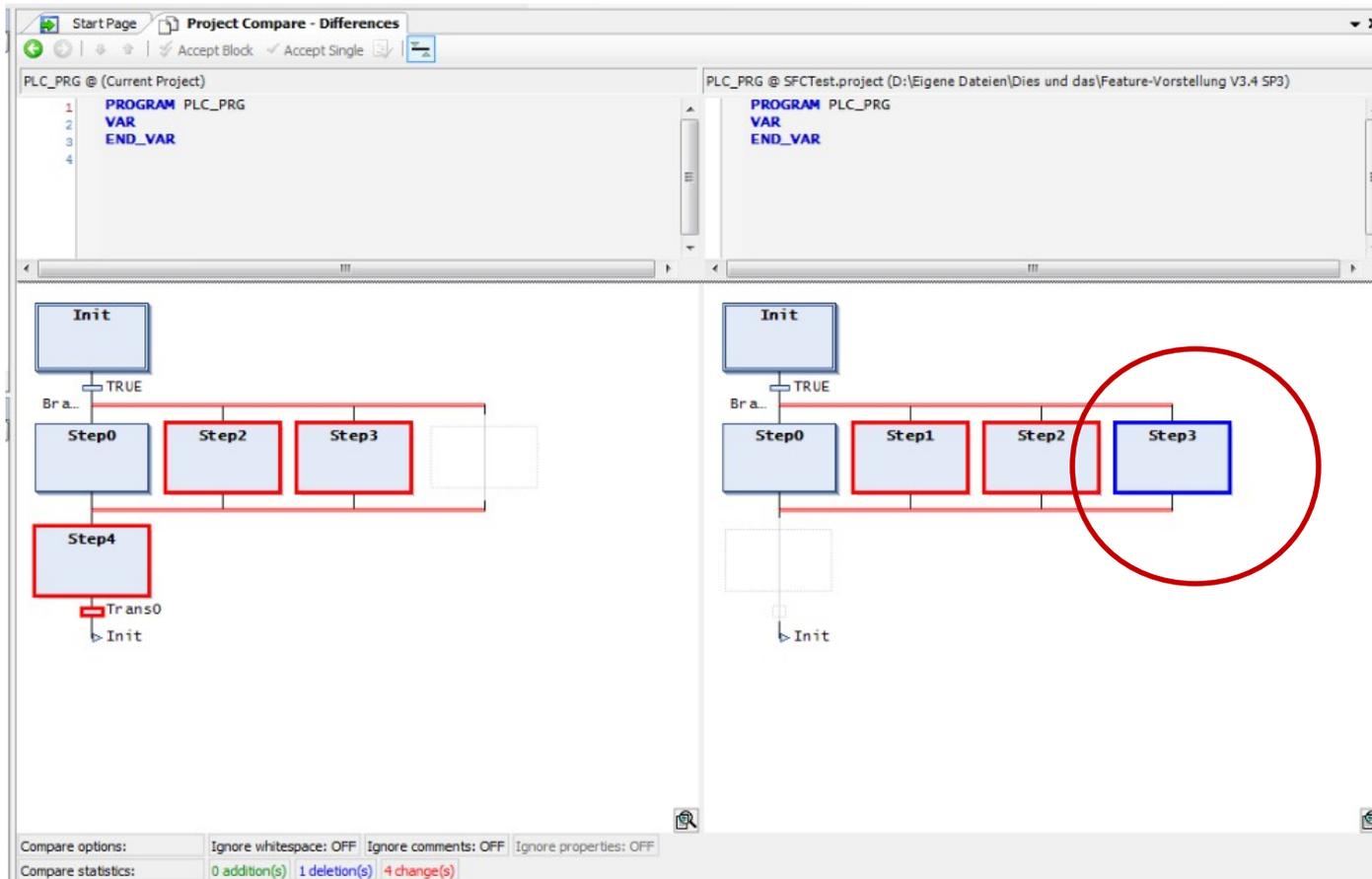
Compare statistics: 1 addition(s) | 1 deletion(s) | 0 change(s)

Information: If a placeholder library resolves differently in the left and right project, it will NOT be displayed as a change. Double-click a line to display the property changes of a library reference.

Compare options: Ignore whitespace: OFF | Ignore comments: OFF | Ignore properties: OFF

Compare statistics: 1 addition(s) | 1 deletion(s) | 0 change(s)

Project comparison: Graphical diff editor for SFC



The screenshot shows a graphical diff editor for SFC (Sequential Function Chart) comparing two projects. The window title is "Project Compare - Differences". The left pane shows the "Current Project" and the right pane shows the "SFCTest.project (D:\Eigene Dateien\Dies und das\Feature-Vorstellung V3.4 SP3)".

Both panes show the same code at the top:

```
PROGRAM PLC_PRG
VAR
END_VAR
```

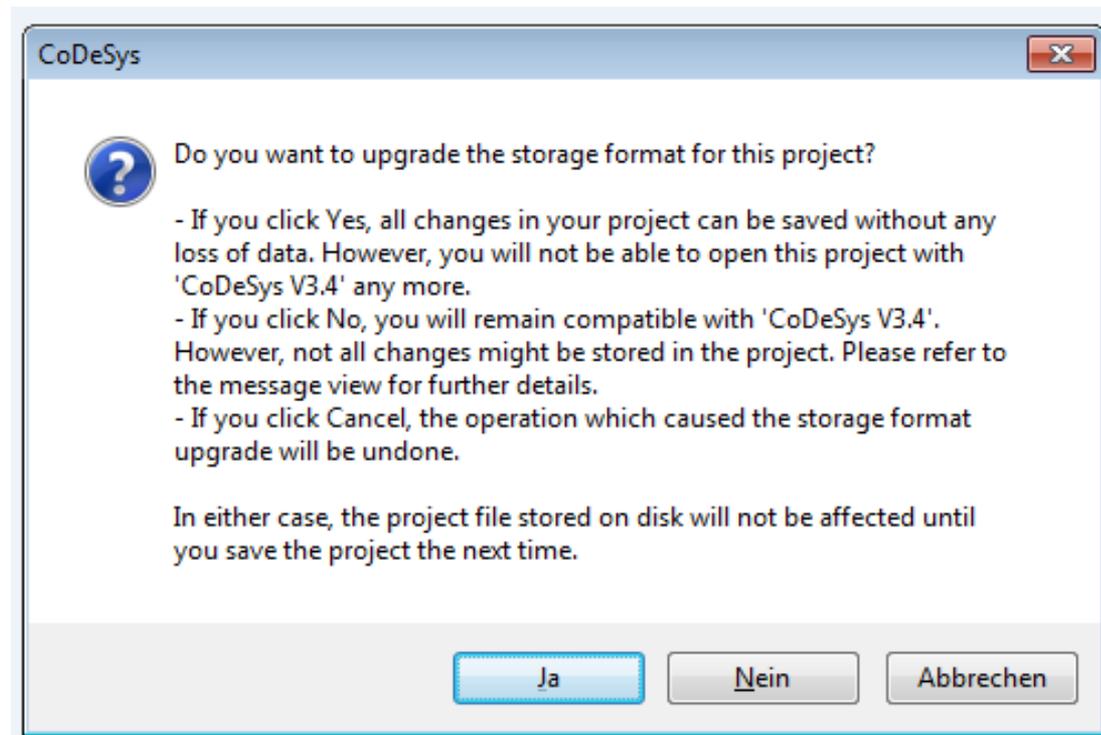
The SFC diagrams below show the differences:

- Left Pane (Current Project):** Shows a sequence starting with "Init" (blue), followed by a branch "Br a..." leading to "Step0", "Step2", and "Step3" (all red). "Step4" (red) is shown below, with a transition "Trans0" leading back to "Init".
- Right Pane (SFCTest.project):** Shows a sequence starting with "Init" (blue), followed by a branch "Br a..." leading to "Step0", "Step1", "Step2", and "Step3" (all red). "Step3" is circled in red. A transition leads back to "Init".

At the bottom, the "Compare options" are: Ignore whitespace: OFF, Ignore comments: OFF, Ignore properties: OFF. The "Compare statistics" are: 0 addition(s), 1 deletion(s), 4 change(s).

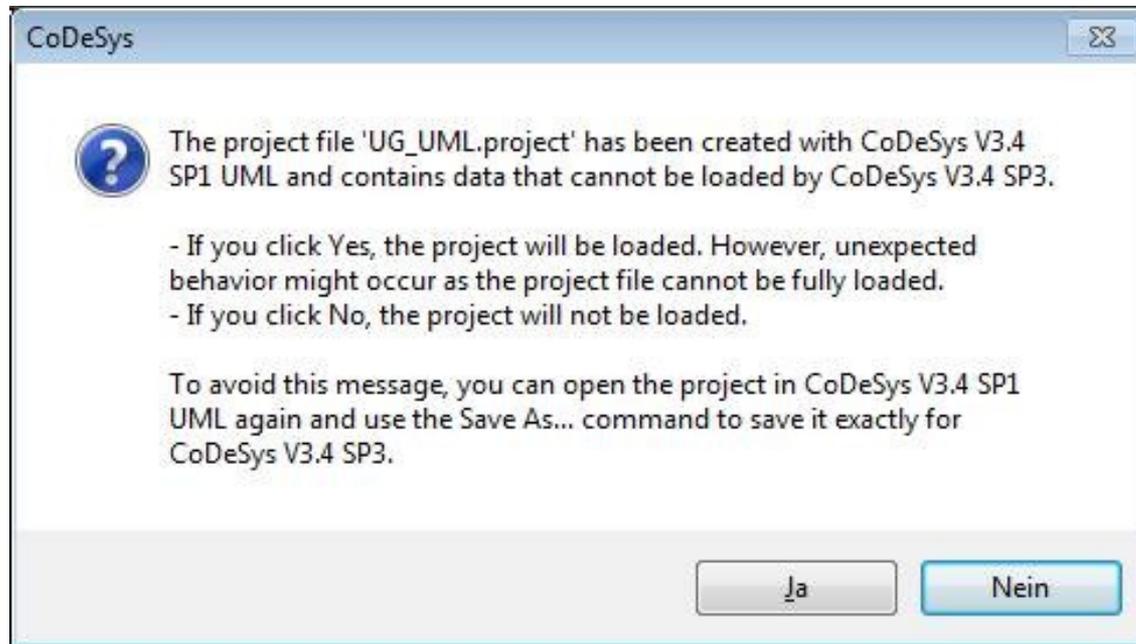
Project compatibility: Improved usability

- Edit operations which would require a project storage format upgrade can be undone with the function “Cancel”.



Project compatibility: Improved usability

- Projects created with a new CoDeSys version can no longer be opened without having to confirm a warning message. In previous versions, the user was often unaware of a possible data loss under certain circumstances.



Package Manager:

- Library profiles are now supported
- Setting of options is now supported

Minor usability improvements:

- Project archives can now be extracted even if a second instance of CoDeSys is running. Library update messages are now displayed in the Message view instead of in a dialog box.
- The editor caption now shows *POU.Name* instead of *Name* for actions, methods, properties of transitions.



New functionality

- Online change with optimized jitter
 - Reduction of jitter through optimized change procedure
 - Display of changes
- Alternative storage allocation algorithm with a complete download during online change:
 - No additional loading of the boot project
 - No memory fragmentation
 - CRC (cyclic redundancy check) of the program code is possible

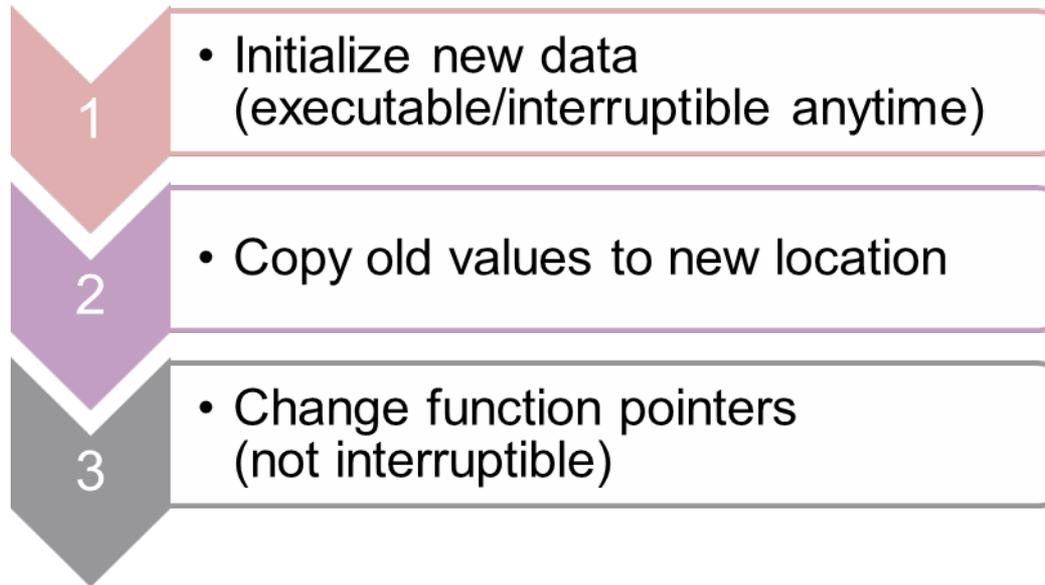


Requirements:

- In order to use this new feature a runtime update is required and the device description settings "runtime_features//optimized_online_change" have to be configured.



Process simplified:



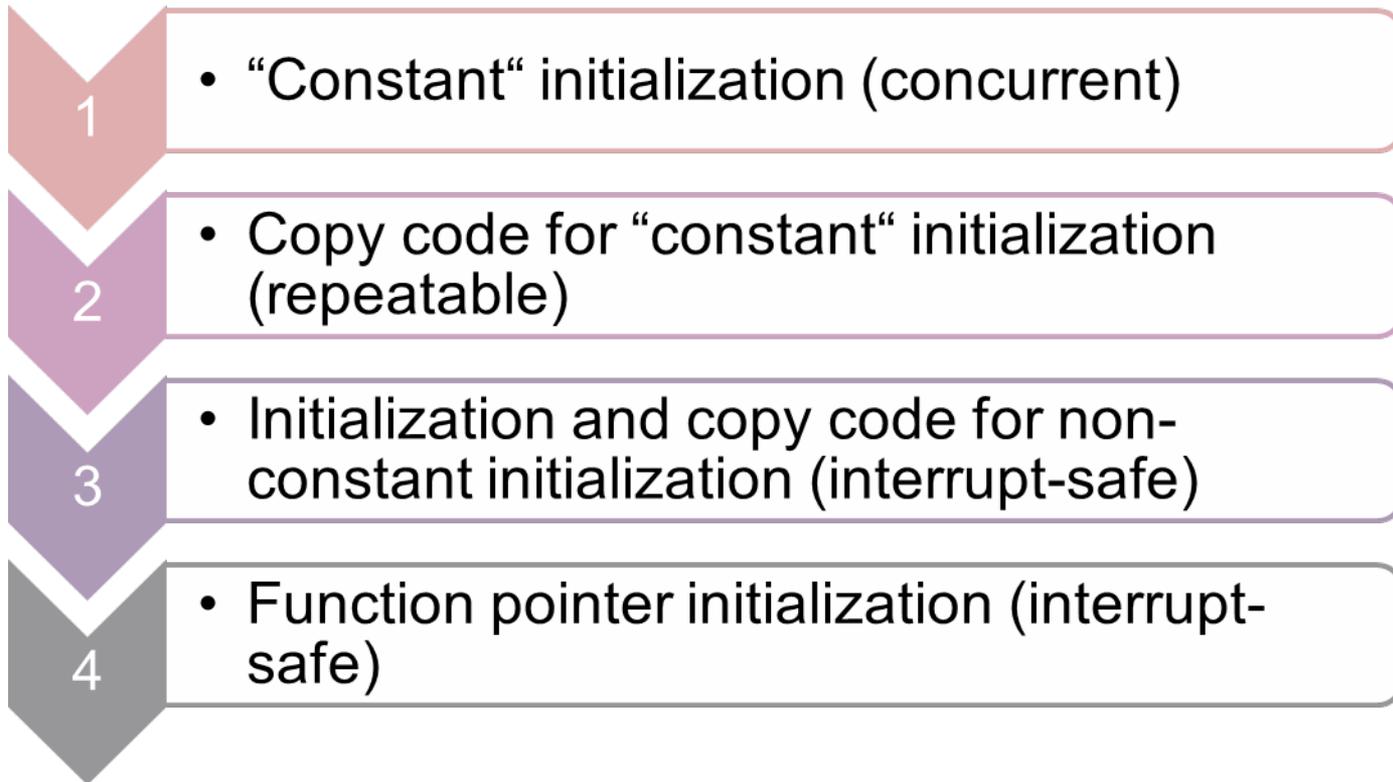
Idea:

- 1 Easy call up
 - 2 Call up in blanking interval; repeat if interrupted
 - 3 Stop task scheduling in blanking interval and then execute
- Minimum jitter

Restrictions:

- Initialization, cannot be done while the task is running, if an
 - initialization is not constant
 - FB_EXIT is necessary
 - FB_INIT contains cross references to moved code
 - FB_INIT is called in the IEC-Cycle
 - FB_INIT contains virtual method calls

Real process



Conclusion:

- All online change procedures are divided into executable, interruptible and non-interruptible steps.
- The different function block instances will be differentiated, and external references will be verified. An external reference is an access to global variables and external calls.
- Then the runtime system can call up the different parts.



Examples for online change events **without** jitter:

- Change size of existing arrays (even really big ones)
- Add new data of any volume
- Call new functions (a complete new call tree will have no effect on the running IEC-tasks)
- Changes in old style function blocks without FB_Init or FB_Exit-methods.

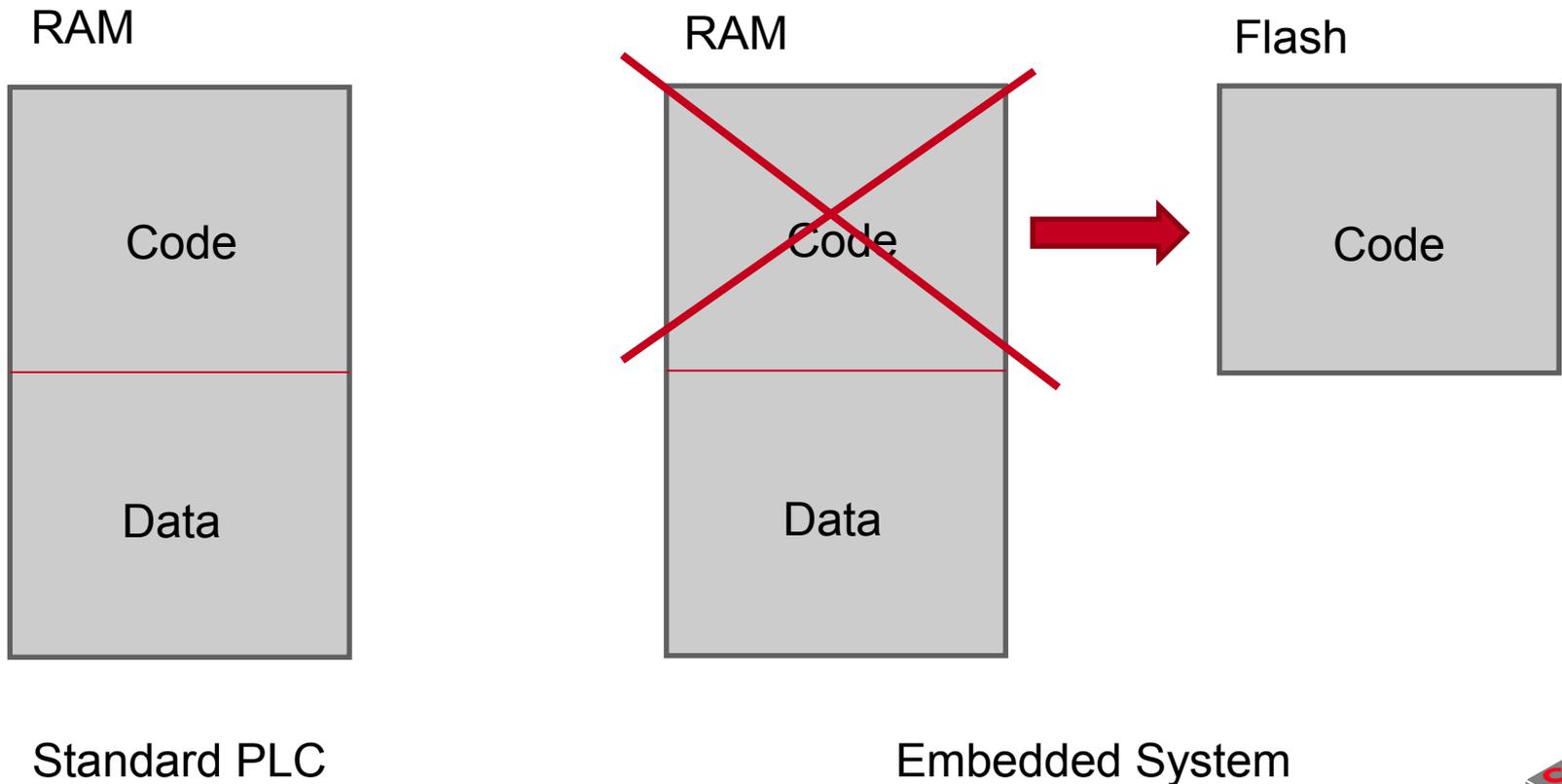
Transfer of symbol config as XML file

- Transfer of the “symbol configuration“ as XML file onto the controller (now contains area/offset/bit and name of direct addresses)
- Symbol configuration is required for external symbol access to IEC variables
- Advantage:
 - no code generation on PLC
 - no use of internal memory



Run from Flash for embedded systems

- Run from Flash for embedded systems



Standard PLC

Embedded System

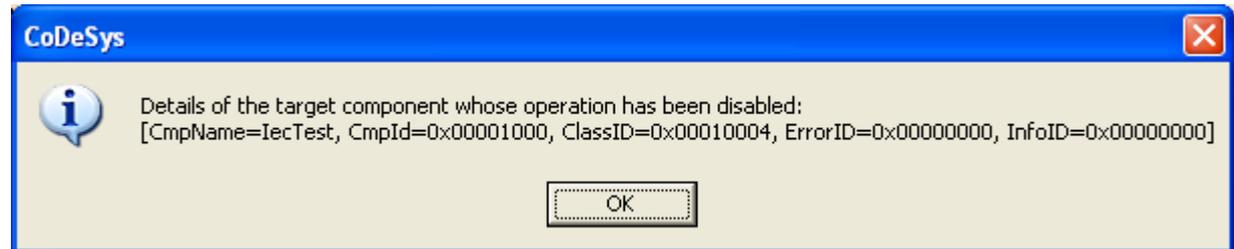


Run from Flash for embedded systems

- Advantages:
 - Especially for embedded systems, in respect of the reduced memory requirements for the RAM
- Disadvantages:
 - No debugging possible
 - No online change possible

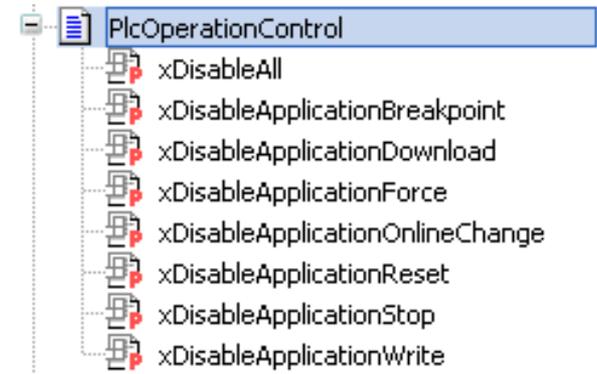


- Generation of message boxes through logger entries or IEC applications during online operation



Operation control to increase security

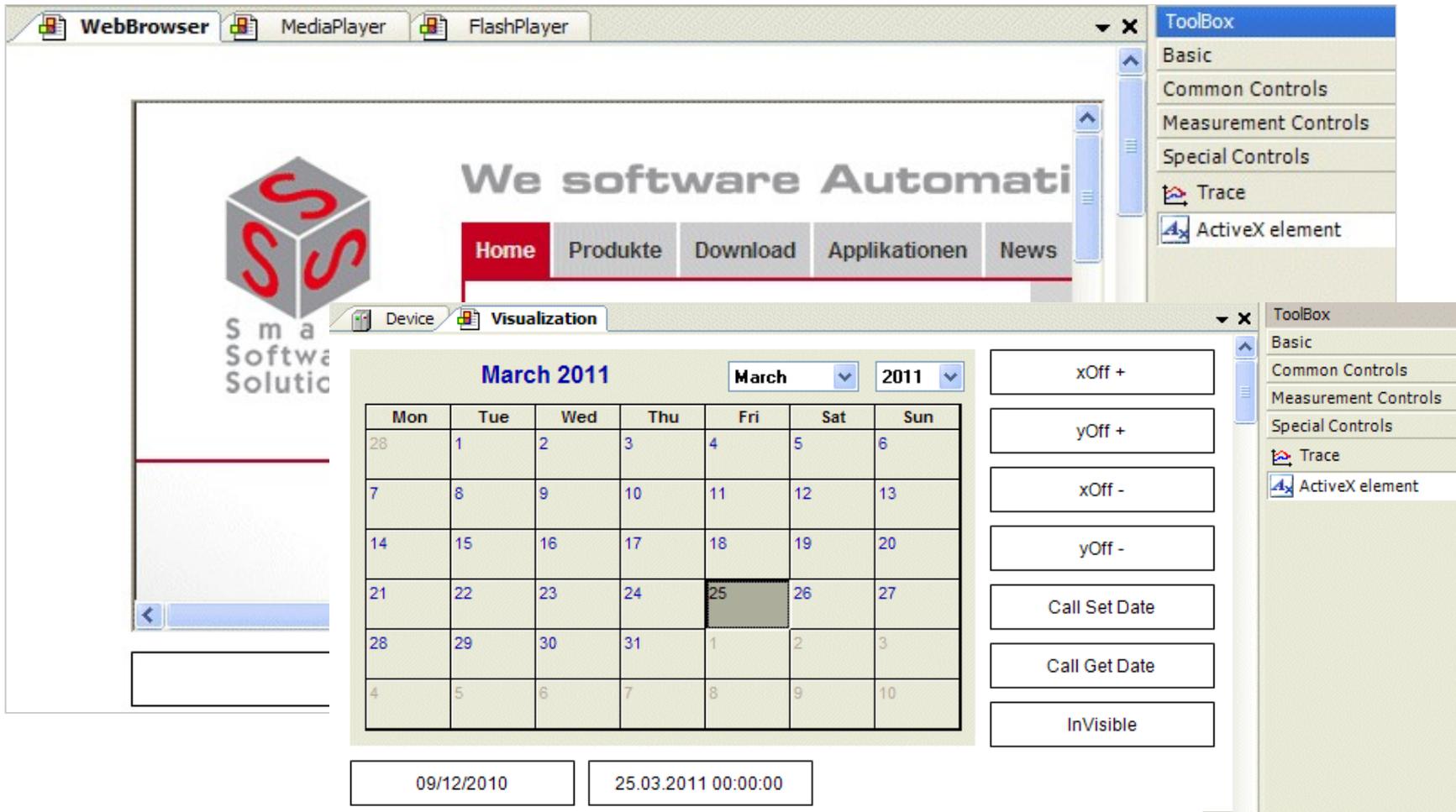
- Access to the PLC can be controlled via application in case of a critical condition to ensure a safe plant operation.
- Application is able to prevent following operations, depending on the status
 - Online Change
 - Force
 - Breakpoints
 - Reset
 - Stop
 - Download
- Operations can be deactivated directly from IEC or C
- Delete of application and reset origin is always possible
- A message will be displayed in CoDeSys



- Visualization element “NativeControl”:
For connection to ActiveX controls for Windows platforms or platform specific objects, like camera objects directly within a visualization.
- Platform-dependent controls in a visualization can be used via NativeControl (ActiveX element within the toolbox).
Examples are web browser, calendar, media player, flash player, ...
- IEC variables can be passed to the element and the element can send back a result via a common interface.



- ActiveX element in the toolbox



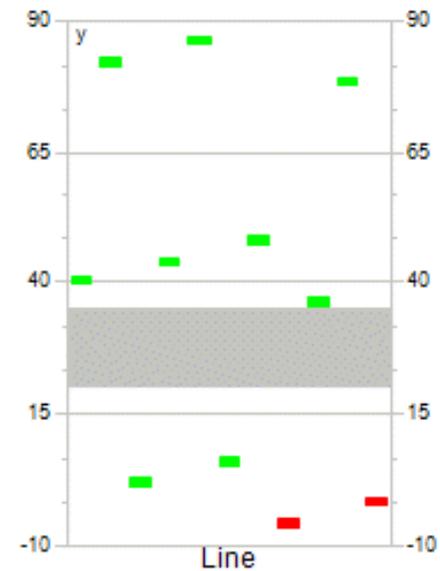
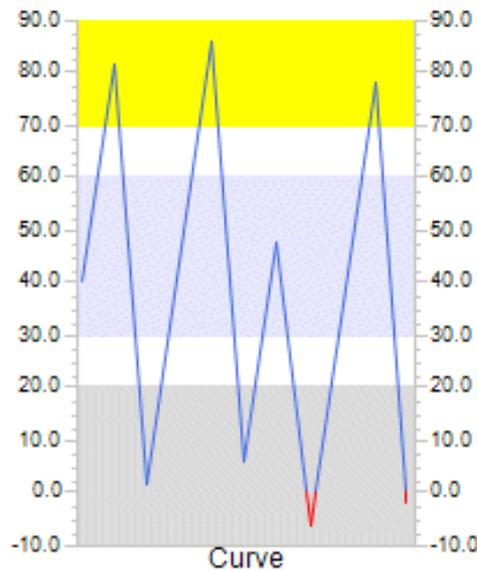
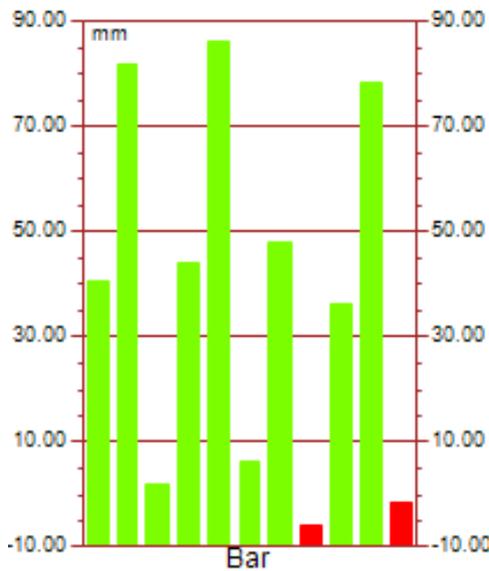
The screenshot shows two windows in the CoDeSys environment. The top window is a WebBrowser displaying a website for 'Smart Software Solutions' with a navigation menu (Home, Produkte, Download, Applikationen, News) and a logo. The bottom window is titled 'Visualization' and contains a calendar for March 2011, a set of control buttons (xOff +, yOff +, xOff -, yOff -, Call Set Date, Call Get Date, InVisible), and two date/time input fields.

March 2011

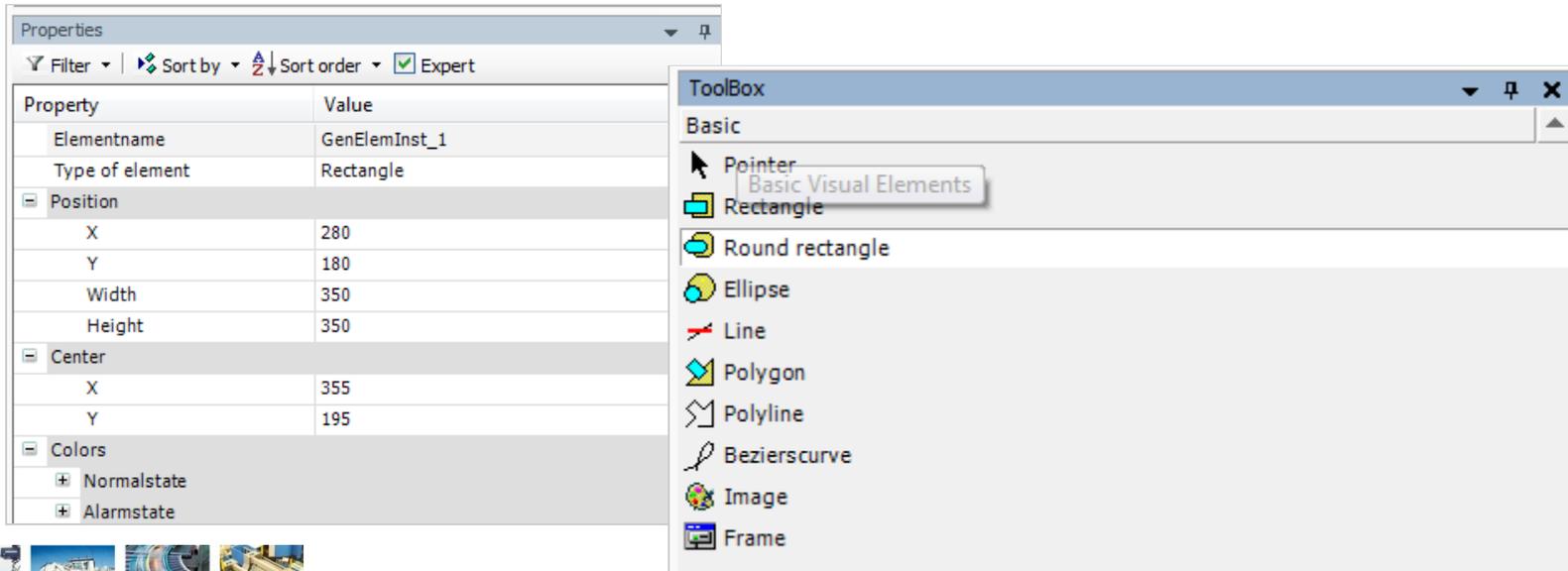
Mon	Tue	Wed	Thu	Fri	Sat	Sun
28	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

09/12/2010 25.03.2011 00:00:00

- Visualization element “Histogram” now available – display of a numerical array as bargraph, curve or line.



- When using the VisualElementToolkit to develop a visualization the texts of the different elements can now be supplied in different languages.
- The element properties as well as the toolbar are now available in German and English language.
- Other languages will follow.



The screenshot displays two windows from the CoDeSys software. The 'Properties' window on the left shows the configuration for a 'Rectangle' element. The 'ToolBox' window on the right lists various visual elements available for development.

Property	Value
Elementname	GenElemInst_1
Type of element	Rectangle
Position	
X	280
Y	180
Width	350
Height	350
Center	
X	355
Y	195
Colors	
Normalstate	
Alarmstate	

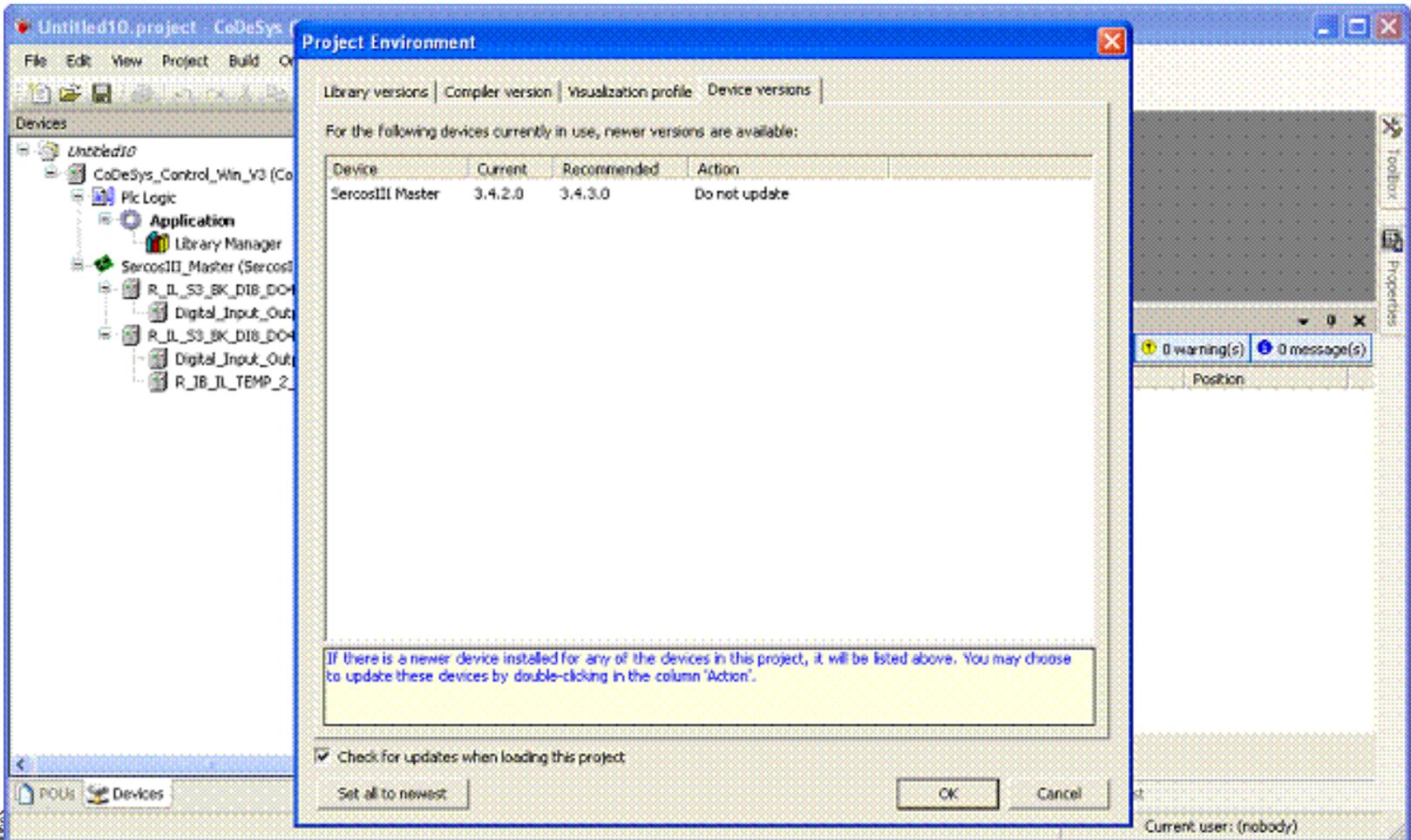
The ToolBox window lists the following elements:

- Basic
 - Pointer
 - Basic Visual Elements
 - Rectangle
 - Round rectangle
 - Ellipse
 - Line
 - Polygon
 - Polyline
 - Bezierscurve
 - Image
 - Frame

- When opening a project an update of all devices will be offered.
- Ethernet over EtherCAT (EoE)



- Device update when opening a project



The screenshot shows the 'Project Environment' dialog box in CoDeSys. The 'Device versions' tab is active, displaying a table of devices currently in use and their recommended updates. The table shows one device, 'SercosIII Master', with a current version of 3.4.2.0 and a recommended version of 3.4.3.0. The action for this device is 'Do not update'. Below the table, there is a note: 'If there is a newer device installed for any of the devices in this project, it will be listed above. You may choose to update these devices by double-clicking in the column 'Action''. At the bottom of the dialog, there is a checkbox for 'Check for updates when loading this project' which is checked, and a 'Set all to newest' button. The 'OK' and 'Cancel' buttons are also visible.

Device	Current	Recommended	Action
SercosIII Master	3.4.2.0	3.4.3.0	Do not update

0 warning(s) 0 message(s)

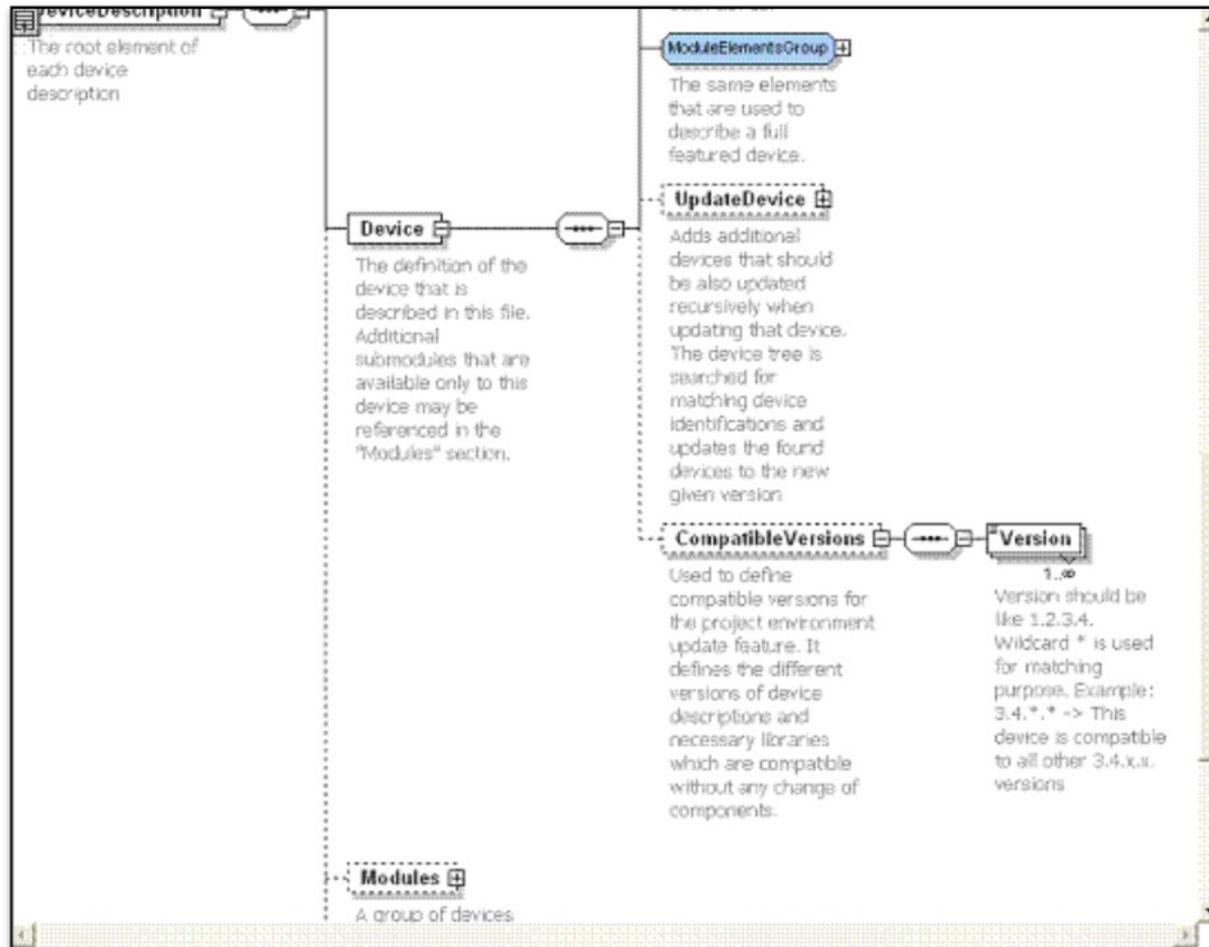
Position

Current user: (nobody)

- Device update when opening a project
 - Functionality is identical to functionality for libraries, visualizations and compilers.
 - Which one of the older versions is to be exchanged must be defined in the device description.
 - Only the supplier of the respective device can estimate the compatibility and can give a recommendation.



- Entry in the .xml file:



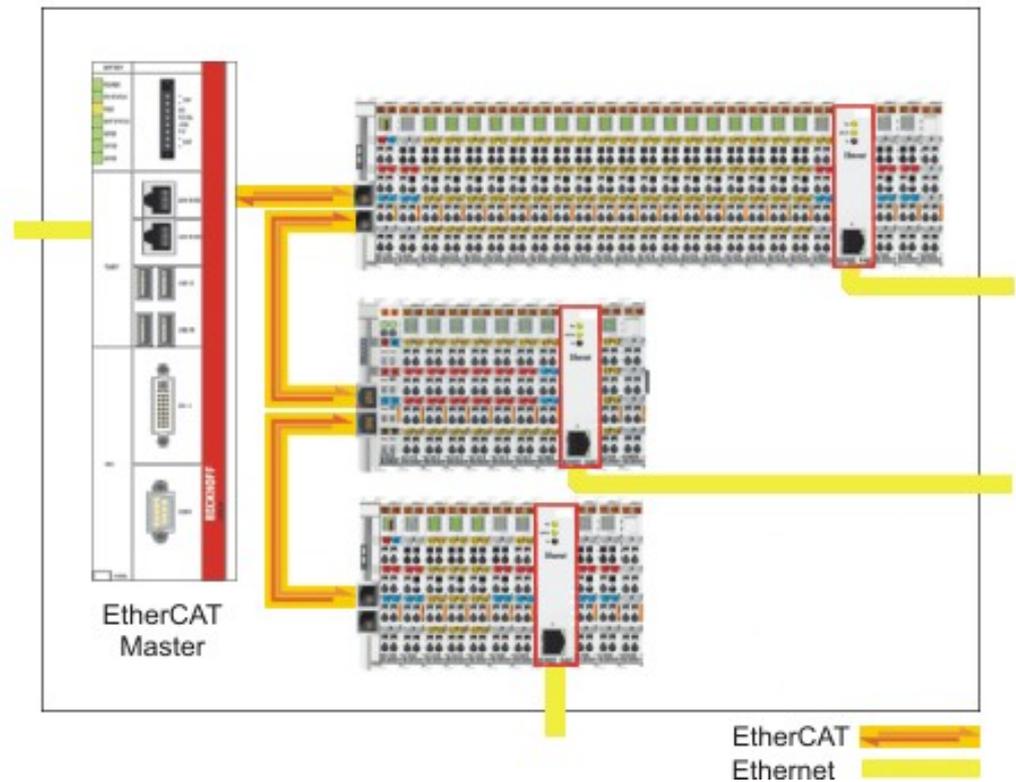
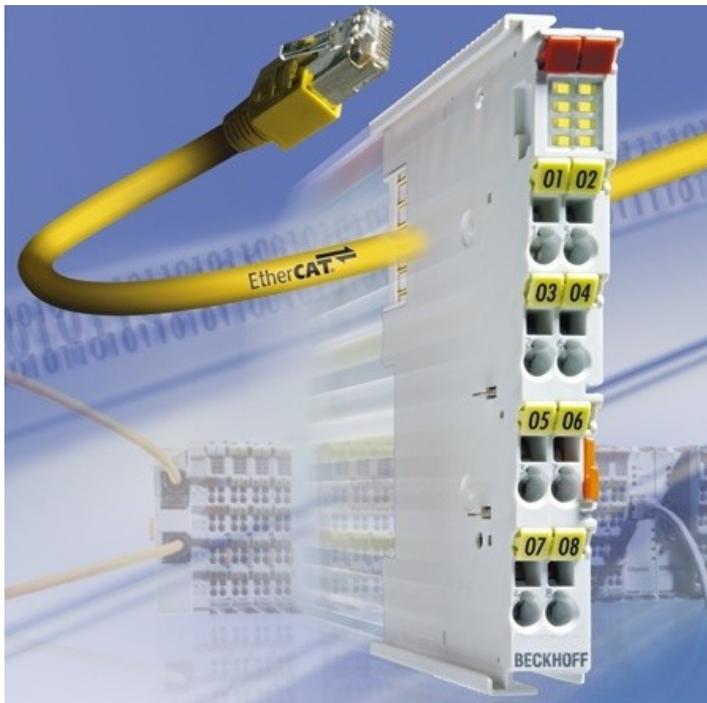
- **Definition: Ethernet over EtherCAT (EoE)**
- Any kind of Ethernet device can be connected within the EtherCAT segment via a switch port. The Ethernet frames are tunneled through the EtherCAT protocol, like it is known from internet protocols (e.g. TCP/IP, VPN, PPPoE (DSL))
- The EtherCAT network is completely transparent for the Ethernet devices and the EtherCAT realtime properties will not be affected.



- Extension for EtherCAT: EoE (Ethernet over EtherCAT)
- Special switch port terminal (EL6601) or devices with EoE protocol (e.g. Indradrive CS) will be supported
- It is possible to connect standard network devices to the switch port terminal, like printer, PC ...
- It is also possible to connect a Modbus TCP slave and to use our Modbus TCP stack without needing an additional network card.



- EL6601 – Switch port terminal for Ethernet from Beckhoff

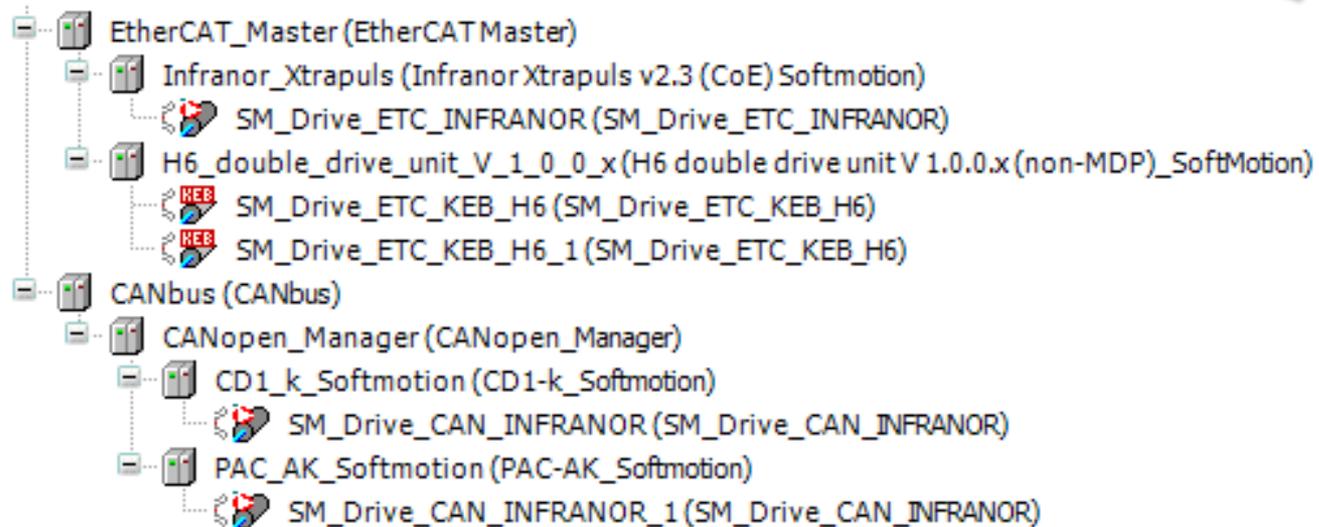


- Indradrive Cs uses EoE for configuration purposes and to download firmware.
- Indraworks can therefore be connected to the servo controller via EtherCAT and no serial data communication is necessary.



The following drives are supported from Version V3.4 SP3:

- KEB H6 Dual (ETC)
- Infranor Xtrapuls PAC (ETC)
- Infranor Xtrapuls PAC (CAN)
- Infranor cd1-k (CAN)



CAM editor

- This feature displays the exact initialization code of a CAM table.
- It helps the customer to understand the way CAM tables are defined in CoDeSys.
- CoDeSys V2.3 offers this feature too.

