



**User Documentation:
CoDeSys V3.x, Installation and First Start**

Document Version 7.0

CONTENT

1	WELCOME	3
2	SYSTEM REQUIREMENTS	3
3	INSTALLATION	3
4	START	3
5	HELP	3
6	CREATE AND RUN A PROJECT	4
6.1	Start CoDeSys and create a project	4
6.2	Write a PLC program	6
6.3	Define the Resource objects for running and controlling the program on the PLC	9
6.4	Run and watch the application on the PLC	12
6.5	Debug an application	16
7	DEINSTALLATION, MODIFY, REPAIR	17
	CHANGE HISTORY	18

1 Welcome

Welcome to the CoDeSys V3.x Development System by 3S - Smart Software Solutions GmbH !

CoDeSys is a device-independent PLC-programming system. Matching the IEC 61131-3 standard it supports all standard programming languages but also allows including C-routines and supports object orientated programming. In combination with the CoDeSys SP runtime system it allows multi-device and multi-application programming. The component-based structure makes possible a customer-specific configuration and extension of the user interface.

Before you are going to install regard the following on system requirements, installation and how to get further information.

2 System Requirements

Required:

- Operating systems: Windows 2000 (Windows XP or Windows Vista also possible)
- RAM: 512MB
- Harddisk: 200MB
- Processor: Pentium V, Centrino > 1,8 GHz, Pentium M > 1,0GHz

Recommended:

- Operating systems: Windows XP
- RAM: 1024MB
- Harddisk: 1GB
- Processor: Pentium V, Centrino > 3,0 GHz, Pentium M > 1,5GHz


3 Installation

Perform the Setup_CoDeSysV<Version>.exe to start the installation assistant which will guide you.

4 Start

Start CoDeSys from the Start menu: Per default the path is Programs – 3S CoDeSys – CoDeSys – CoDeSys V<version>.



Alternatively use the CoDeSys icon  which is available on the desktop after installation.

5 Help

Online help is provided via the 'Help' menu. There you find the commands to open a Contents view of the help, the Index view for searching by keywords or the Search view for full-text search.

6 Create and run a project

See in the following a description of how to create a simple project containing a PLC program, further how to load this program via a Gateway Server to the PLC (target device) and to get it run and monitored. The PLC runtime system used for this example project per default is provided with the CoDeSys setup.

The sample program will be written in Structured Text language and consist of a program *PLC_PRG* and a function block *FB1*. *PLC_PRG* will contain a counter variable *ivar* and call function block *FB1*; *FB1* will get input "in" from *PLC_PRG*, will add "2" on this input and will write the result to an output *out*; *out* will be read by *PLC_PRG*.

(Regard that the following descriptions refer on the default configuration of the user interface provided with the currently installed version of the programming system.)

6.1 Start CoDeSys and create a project

(1) Start CoDeSys

From the Start menu on your PC choose

Programs > 3S CoDeSys > CoDeSys > CoDeSys V<version>

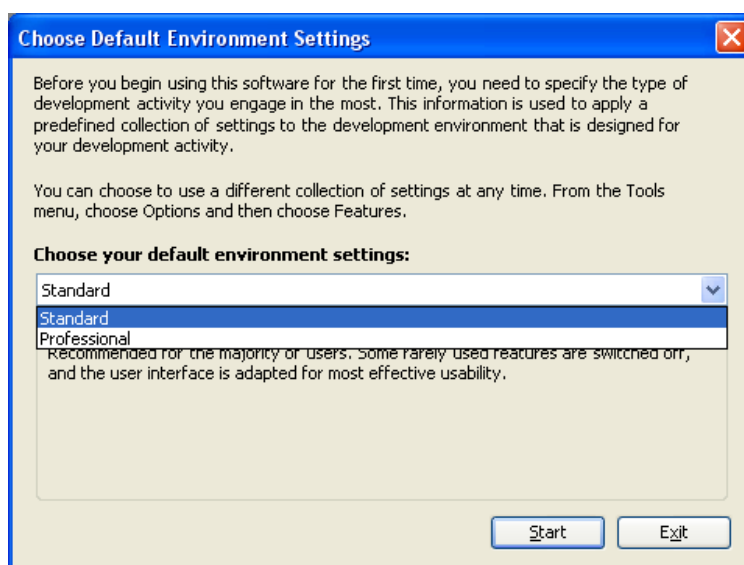


Alternatively you can start via the CoDeSys Icon that will be available on the desktop after installation.

(Future: You will be asked to select a Profile and after pressing **Continue** the CoDeSys user interface will open.)

The programming system will be automatically started with a predefined profile. So currently you do not to have select a profile.

When you start the programming system the first time after first installation on the system, you will be asked to choose the default collection of settings and features:

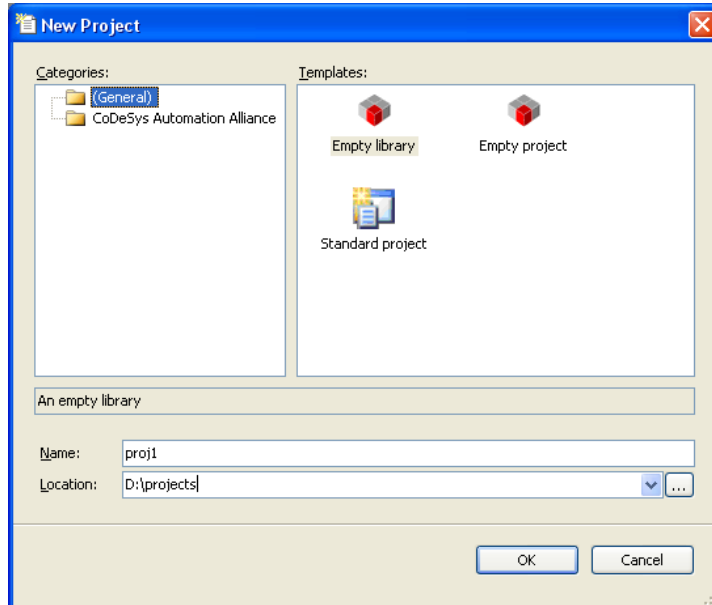


Choose "Standard" for the first steps described here, you always can switch to the "Professional" settings (via the [Project Options](#) dialog) if necessary . In the "Standard"

environment the user interface is adapted for most effective usability and some rarely used features are left out.

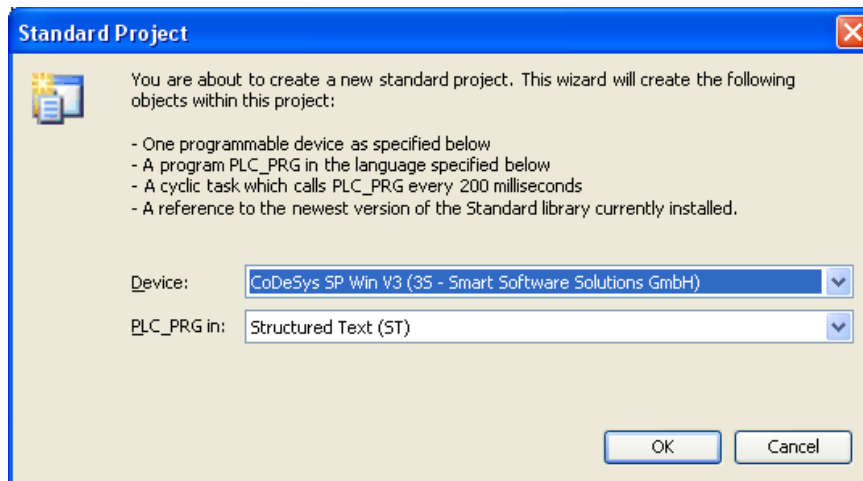
(2) Create a project

To create a new project, choose command **New project** from the **File** menu:



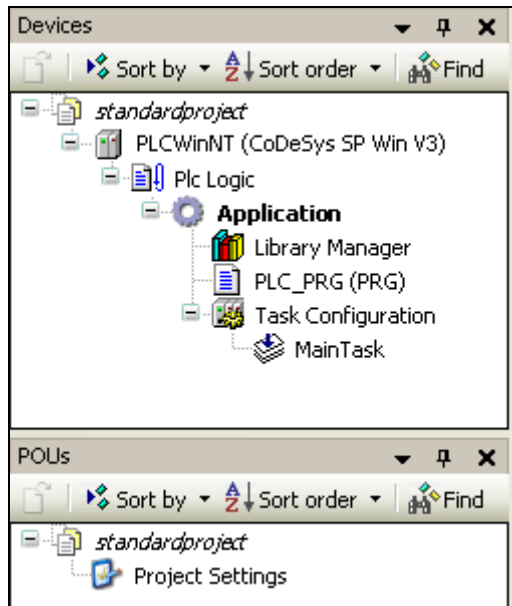
In the **New Project** dialog select **Standard project** in the 'Templates' field and enter a **Name** and a **Location** path for the project file. Press **OK** to confirm.

A wizard dialog will open:



Choose device CoDeSys SP Win V3 and programming language Structured Text for PLC_PRG. Press OK to open the new project.

The project name now will appear in the title bar of the CoDeSys user interface and as a symbolic root node in the POU's and the Devices view windows.



Per default one program POU, the Project Settings and some resource objects are available:

The POUs window contains the Project Settings.

The Devices window shows tree with a device "Device (CoDeSys SP Win V3)" of type CoDeSys SP Win V3 with an application adjoined below. The latter includes the program "PLC_PRG" to be edited in Structured Text and the obligatory Task Configuration defining a "MainTask" that controls PLC_PRG.

Furthermore there is a Library Manager, automatically including the library "IoStandard.library" necessary for IO-configurations and the Standard.library providing all functions and function blocks which are required matching IEC61131-3 as standard POUs for an IEC programming system.

(At the moment do not care about the additional node "Plc Logic" inserted below the device node "Device (CoDeSys SP Win V3)". It is just a symbolic node indicating that the device is a "programmable" one.)

The symbolic device name can be renamed by selecting this entry, opening an edit field with the <Space> key and entering another name instead of "Device". Do this and enter PLCWinNT" instead.

6.2 Write a PLC program

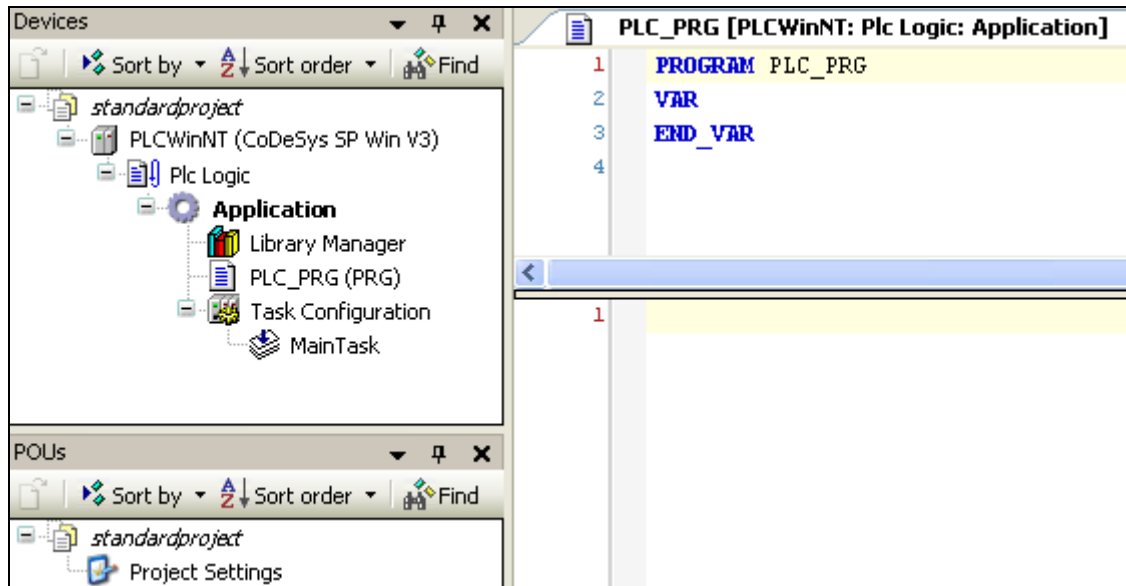
(3) Declare variables in PLC_PRG

Select the default POU "PLC_PRG" which is already available in the POUs window.

In the center part of the window the **ST language editor** window for 'PLC_PRG' will open. The editor consists of a declaration part (upper) and a "body" (lower part), separated by a screen divider.

The declaration part shows line numbers at the left border, the POUs type and name ("PROGRAM PLC_PRG") and the embracing keywords "VAR" and "END_VAR" for the variables declaration.

The body is empty, only line number 1 is displayed:



In the declaration part of the editor put the cursor behind VAR and press the Return-key. A new empty line will be displayed where you enter the declaration of variables *ivar* and *erg* which are of type INTEGER and *fbinst* of type FB1:

```
PROGRAM PLC_PRG
VAR
    ivar: INT;
    fbinst: FB1;
    erg: INT;
END_VAR
```

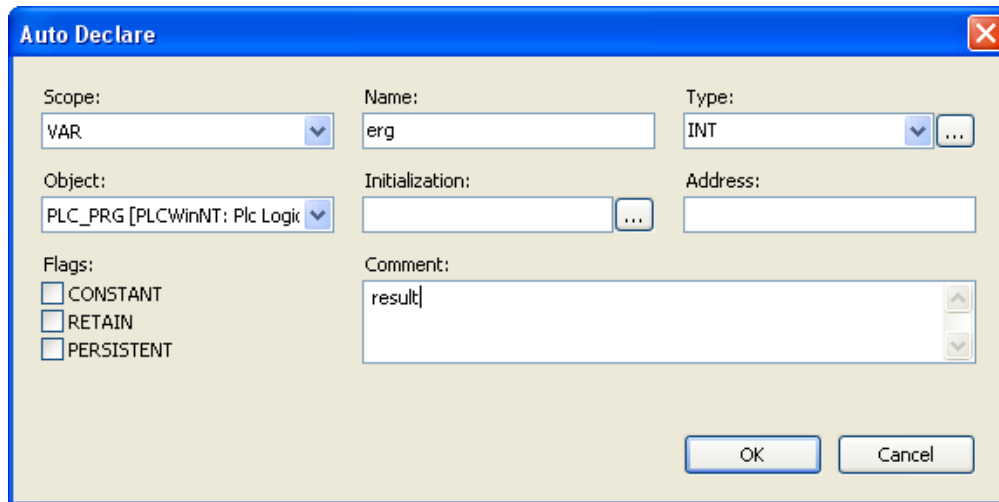
Instead of this you could directly type a line in the implementation part of the editor (body) and use the Autodeclare function, see (4).

(4) Enter programming code in the body of PLC_PRG

In the body part of the PLC_PRG editor put the cursor in line 1 and enter the following lines:

```
ivar := ivar+1;           // counter
fbinst(in:=11, out=>erg); // call function block FB1, input parameter
                          // with input parameter "in"
                          // output is written to "erg"
```

Instead of steps (3) and (4) you could use the Auto Declaration feature: Immediately enter an implementation line in the body of the program, then press the <Return>-key. For each not yet declared variable found in the new line the Auto Declare dialog will open, where you can do the declaration settings:



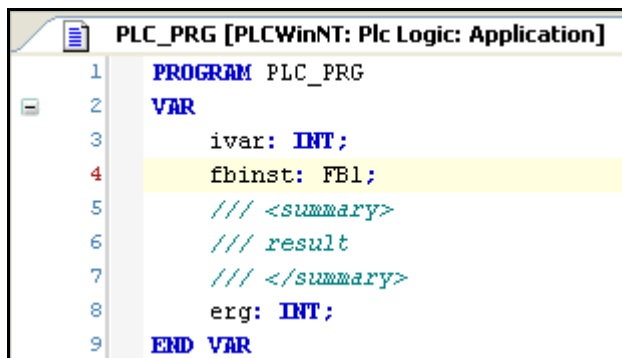
The 'Auto Declare' dialog box is shown with the following fields and options:

- Scope: VAR
- Name: erg
- Type: INT
- Object: PLC_PRG [PLCWinNT: Plc Logic]
- Initialization: (empty)
- Address: (empty)
- Flags: CONSTANT, RETAIN, PERSISTENT
- Comment: result

Buttons: OK, Cancel

The variables name and scope as well as the current POU (Object) will be filled in automatically. Enter the desired type and initialization value according to the declaration described in (3). Regard that comments if you define them here in the autodeclaration dialog, will not be inserted indicated by `///` like shown above, but as an xml-description in the declaration part and will later be usable for documentation purposes.

Confirm the dialog with OK. This will enter the declaration of `erg` in the declaration part of the POU with the comments ahead:



```
1 PROGRAM PLC_PRG
2 VAR
3     iver: INT;
4     fbinst: FB1;
5     /// <summary>
6     /// result
7     /// </summary>
8     erg: INT;
9 END_VAR
```


(5) Create a further programming POU (ST function block FB1)

We add a function block which will add "2" on the input given by variable "in". The result will be written to output "out":

Choose command **Add object** from the **Project** menu.

Select 'POU' in the left part of the 'Add Object' dialog. Enter a Name "FB1" for the POU, activate option **Function Block** in the Type section.

Choose 'Structured Text (ST)' for the **Implementation language**.

Press button **Open** to confirm the object settings.

A further editor window will open for the new function block FB1. Declare there the following variables, in the same way as having done for PLC_PRG:

```
FUNCTION_BLOCK FB1
VAR_INPUT
    in: INT;
END_VAR
VAR_OUTPUT
    out: INT;
END_VAR
VAR
    ivar: INT:=2;
END_VAR
```



In the implementation part of the editor enter the following:

```
out:=in+ivar;
```

6.3 Define the Resource objects for running and controlling the program on the PLC



(6) Start Gateway Server and PLC

Start Gateway Server:

The Gateway Server is started automatically at system start as a service. Make sure that there is an icon  in the system tray, indicating that the gateway is running. If the icon is looking like , the gateway currently is stopped.

(This icon is part of the GatewaySysTray program which is available for controlling and monitoring the Gateway service. It provides a menu with a start and a stop command, thus allowing the user to stop or restart the service manually. NOT YET IMPLEMENTED: Also a Gateway Inspector function is available. The menu also includes the command Exit Gateway Control, which just terminates the GatewaySysTray program, not however the Gateway service. The GatewaySysTray program is started automatically when Windows is started, however it also can be started manually via the Programs menu.)

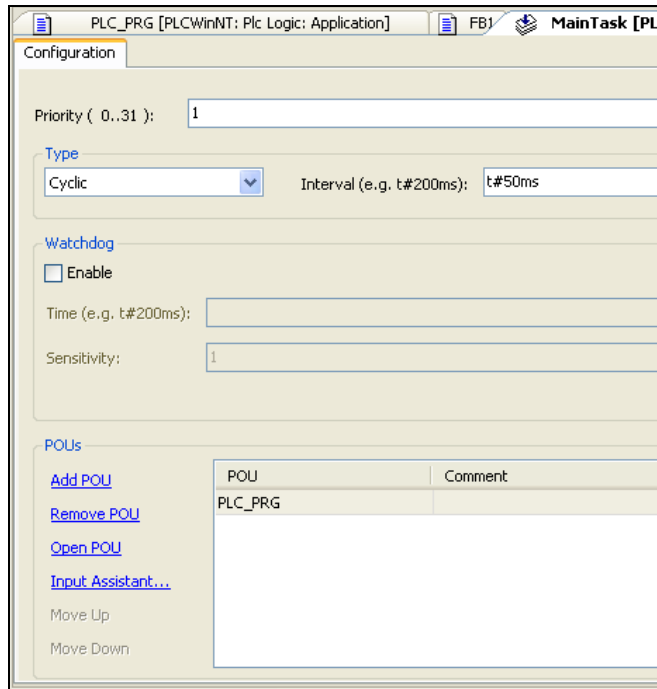
Start PLC:

The PLC (CoDeSys SP Win) is available as a service at system start. It is represented by an icon in the system tray:  for status 'stopped',  for status 'running'. If allowed by the system, the PLC service will be automatically started at system start. Otherwise you have to start it manually by command 'Start PLC' from the menu you open by a mouse-click on the icon.

(This icon is part of the CoDeSysSPSysTray program which is available for controlling and monitoring the CoDeSys SP Win service. It provides a menu with a start and a stop command, thus allowing the user to stop or restart the service manually. The menu also includes the command Exit PLC Control, which just terminates the CoDeSysSPSysTray program, not however the PLCservice. The CoDeSysSPSysTray program is started automatically when Windows is started, however it also can be started manually via the Programs menu.)

(7) Set the "Active Application"

In consequence of a mouse-click on 'MainTask' in the devices window of "standardproject" an editor view containing the configuration of the task opens:



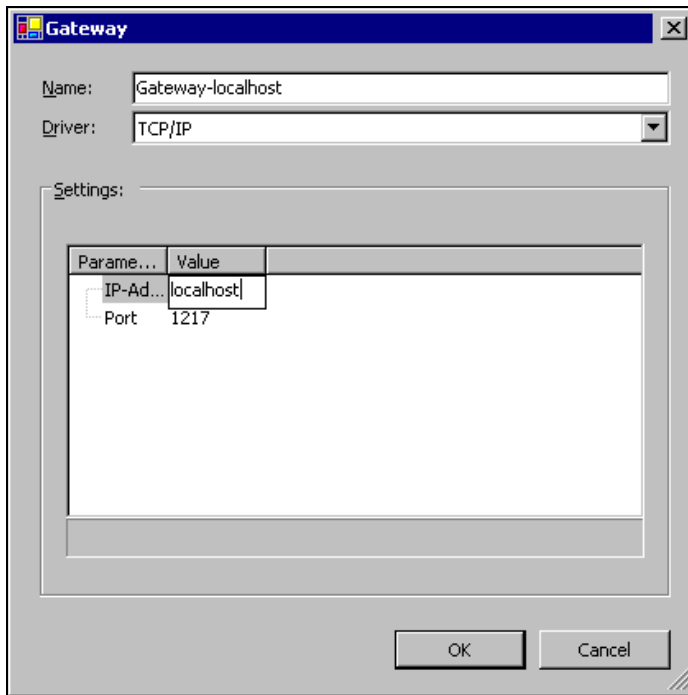
In the Devices window the name 'Application' is displayed in bold letters. This means that this application is set as "active application". Thus all commands and actions concerning the communication with the PLC will refer to this application. In order to set an application as "active one" select the application entry in the Devices window and choose command **Set Active Application** from the context menu.

(8) Configure a communication channel to the PLC

Perform a double-click on entry "PLCWinNT (PLCWinNT)" in the devices window. The dialog "PLCWinNT" will open with subdialog "**Communication**". Here you have to set up the connection between the PLC (target, device) and the programming system according to the following steps. The resulting connection will finally be entered in the line below **Select network path to the controller**:

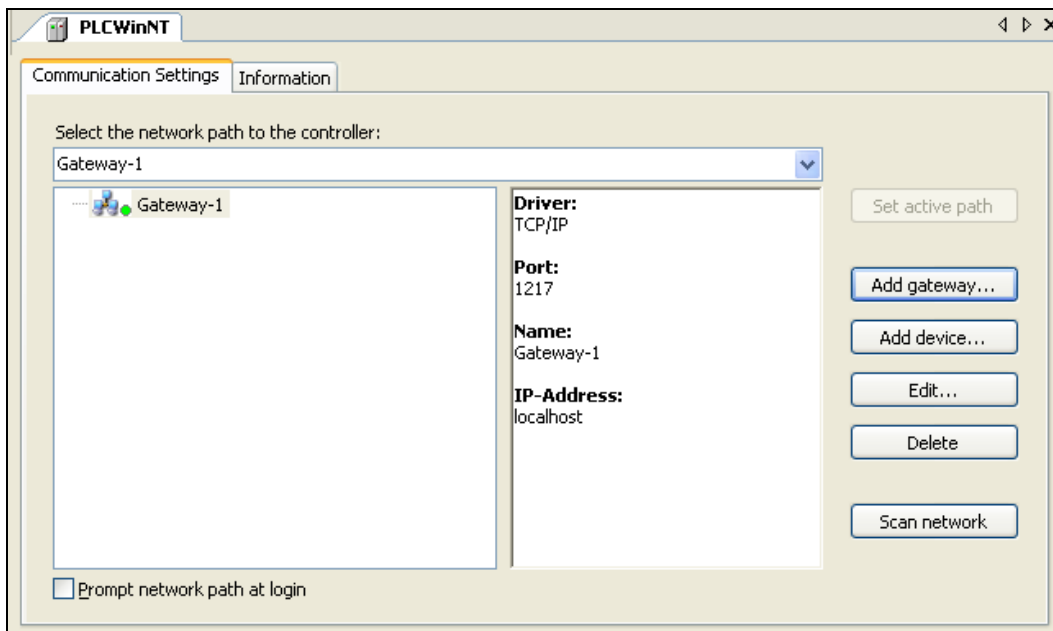
If this is your first communication setup with CoDeSys V3.x, you now have to define the local Gateway Server. (If you have defined the server already in previous sessions, it will be displayed in the communication settings dialog as shown in the figure on the next page. In this case you can skip this step and continue with defining the communication channel to the target, see also on the next page.)

The server is provided with the CoDeSys setup. Press button **Add gateway** to open the Gateway dialog:



Enter a symbolic **Name** for the Gateway, define the Driver type "TCP/IP" and enter IP-address "localhost" (Perform a double-click on the column field to open an edit frame). Leave the setting for the Port. Confirm with **OK**.

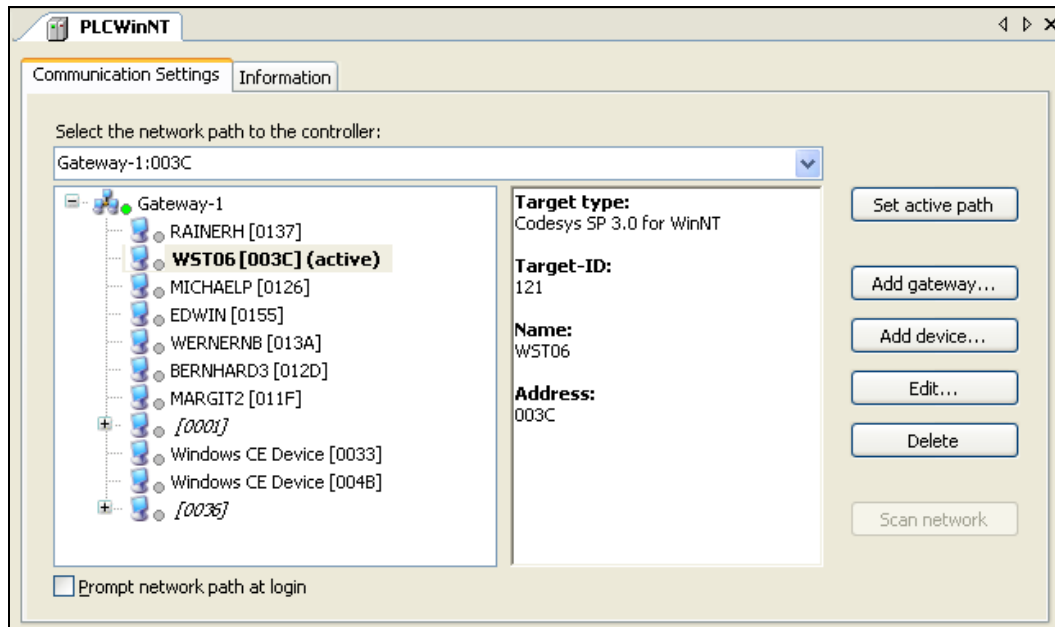
The gateway will be entered in the field on the left part of the Communication dialog and the name of the Gateway will be added to the selection list under 'Select the network path to the controller'. When the gateway is properly running, a green bullet is displayed before the entry, otherwise a red one:



Now define a channel to the target device which should be connected via the set gateway:

Press button **Scan network** to search for available devices in your local network.

You should at least find the PLC installed with the CoDeSys setup: It will be displayed indented below the gateway: instead of "WST06 [003C]" shown in the picture below you should find the name and address of your computer.



Now select the PLC (device) entry and press button **Set active path**.

This will set this communication channel as the active one, which means that all actions concerning communication will exactly refer to this channel. Regard this later when you may have set up several communication channels in your project.

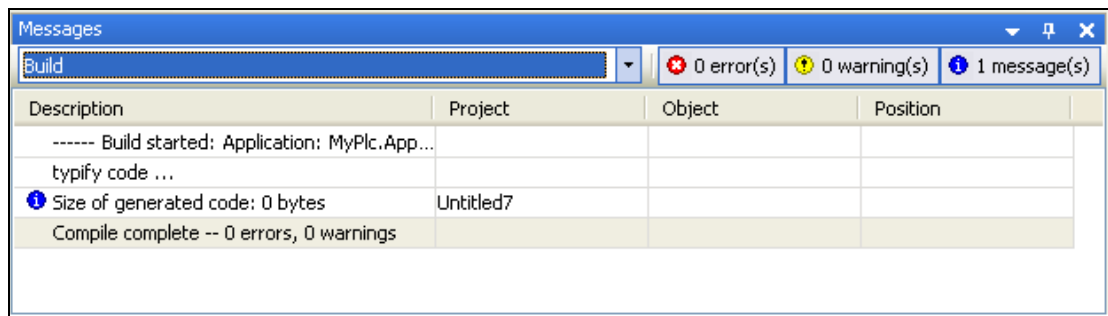
The settings will be applied automatically, you can leave the Communication dialog now.

6.4 Run and watch the application on the PLC

(9) Compile and load application to the PLC

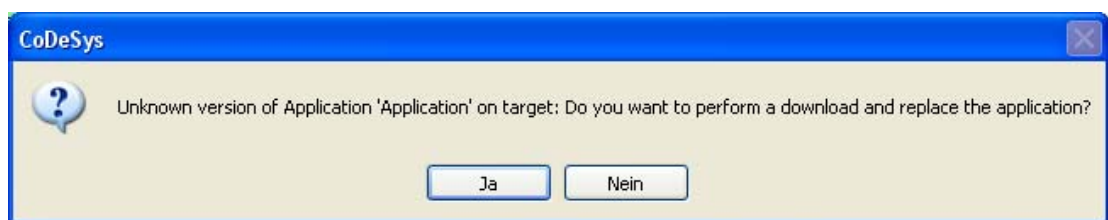
If you just want to check your "active" application program for syntactic errors, perform command 'Build' (Context menu when application object is selected, Build menu) or use <F11>.

Attention: No code will be generated in this case. Error messages will be displayed in the Messages window which is placed at the lower part of the user interface per default.



Even if this syntactical check has not be done before, you can log into the PLC. (Therefore make sure, that the PLC is running, that is the symbol in the system bar is colored).

Use command **Login** (context menu when the application object is selected). If the communication settings have been configured as described in (8) the following message box will appear (otherwise you will be asked to correct the communication settings):



Confirm with Yes to start the compilation and download of the application.

The compile messages will be displayed in the Messages window. If the project has been created correctly, no compilation errors are to be expected, so that the application can now be started, see (11).

(10) Start and monitor application

Having created a 'standardproject' and downloaded the error-free application 'Application' as described in the previous steps, this application can now be started on the device CoDeSys SP Win ('PLCWinNT').

Starting the application on the PLC:

Perform command **Start**, which is in the context menu per default, when the application object is selected.

In consequence the program starts running.

Monitoring the application:

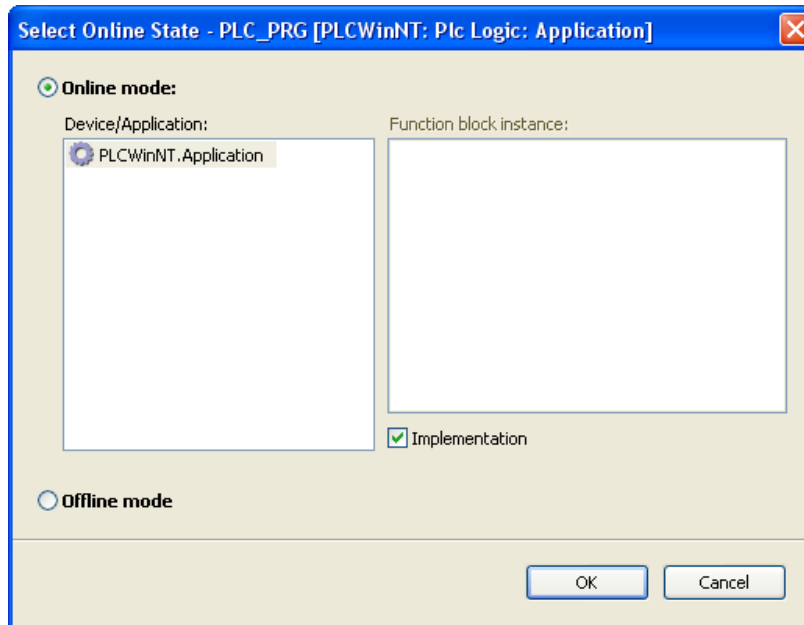
There are three possibilities for watching the variables of the application program:

1. Watch views with defined watch lists
2. Writing and forcing variables
3. Online views of the particular POUs.

1. Open an instance window of the program

The instance view of a POU provides all watch expressions of that instance in a table view in the declaration part and – if activated – as “inline monitoring” also in the implementation part.

In order to open the online view, perform a double-click on "PLC_PRG" in the Devices window or select this entry and choose command 'Edit object' (context menu). The following dialog will open, showing all instances of PLC_PRG (only one in our example):



Here you can choose whether the POU should be viewed in online or offline mode. Online mode will be activated per default and thus just confirm with **OK** for the current example.

The online view of PLC_PRG will be opened: In the lower part you see the code lines as entered in offline mode, supplemented by the little inline monitoring windows behind each variable, showing the actual value. In the upper part a table shows the watch expressions of the POU, i.e. the current values of the respective variables in application "Application" on the PLC.

Expression	Type	Value	Prepared value
ivar	INT	43	
fbinst	FB1		
erg	INT	13	

```

1  ivar 43 :=ivar 43 +1;           // counter
2  fbinst(in 11 :=11, out 13 =>erg 13); // call function block of type FB1
3
4                                     // with input parameter "in"
5                                     // output is written to "erg" RETURN

```

2. Writing and forcing variables:


You can write or force a "Prepared value" to variable ivar on the PLC, which means that ivar will be set to this value at the beginning of the next cycle. Perform a double-click on the field in column **Prepared value**, enter a desired integer value and leave the field by <Return> or by a mouse-click outside of the field. Perform command 'Write values' resp. 'Force values'

(Debug menu) to write or force this value to the PLC. You will see the result at once in column Value.

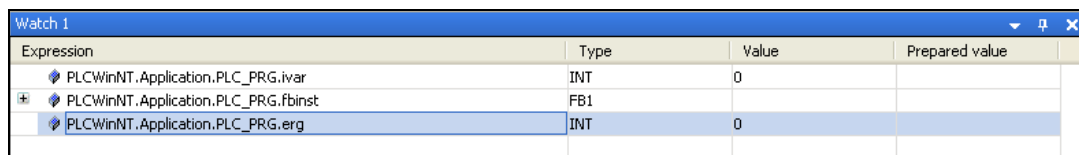
3. Use the watch views:

Watch view windows can be used to configure specific sets of watch expressions of the application, e.g. for the purpose of debugging.

From the **View** menu use command 'Watch' -> 'Watch1'. The watch window will open.

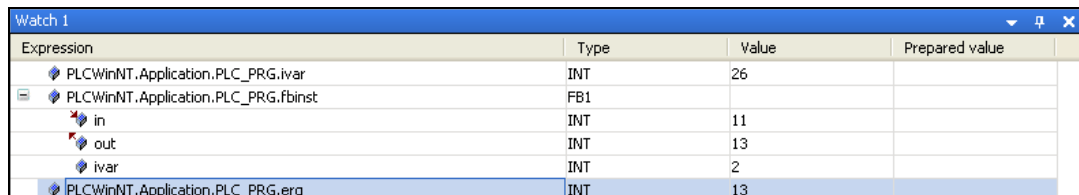
In column **Expression** perform a mouse-click in the first line of the table to open an edit frame. Enter the complete path for variable ivar which should be monitored: "Device.Application.PLC_PRG.ivar". It is recommended to use the input assistant via button  for this purpose. Close the edit frame with <Return>. The type will be added automatically.

Do the same for the further variables. The watch list shown in the next picture just contains expressions of PLC_PRG, but of course you might create a set of any variables of your project. Regard that for instance variables, e.g. for the FB1-instances, it is sufficient to enter the expression "Device.Application.PLC_PRG.fbinst". The particular variables will be entered automatically and the corresponding lines can be opened via the plus-symbol: The current value of a variable is displayed in the 'Value' column:



Expression	Type	Value	Prepared value
PLCWinNT.Application.PLC_PRG.ivar	INT	0	
PLCWinNT.Application.PLC_PRG.fbinst	FB1		
PLCWinNT.Application.PLC_PRG.erg	INT	0	

If not yet done, now perform command 'Start Application' from the Online menu. The application will be started on the PLC and the current value will be displayed in column Value:



Expression	Type	Value	Prepared value
PLCWinNT.Application.PLC_PRG.ivar	INT	26	
PLCWinNT.Application.PLC_PRG.fbinst	FB1		
in	INT	11	
out	INT	13	
ivar	INT	2	
PLCWinNT.Application.PLC_PRG.erg	INT	13	

Writing and forcing of values is possible like described above in 2.

To disconnect from the PLC perform command 'Logout' from the Online menu.

6.5 Debug an application

(11) Set breakpoint and step through the program

In online mode you can set breakpoints as defined break positions for the program execution.

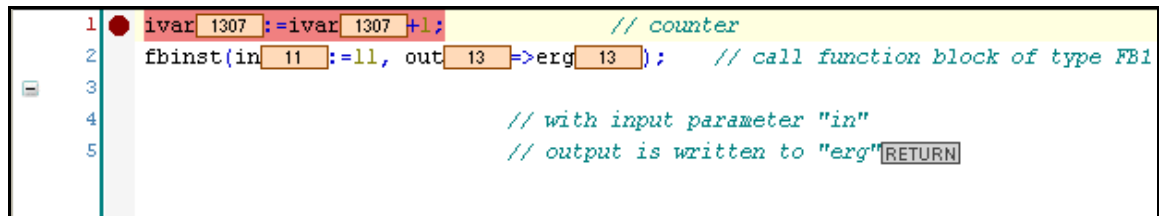
When the program has reached a breakpoint you can execute the program in single steps. At each halt position you see the current value of the variables in the monitoring views.

Try the following:

Select line 1 of PLC_PRG. Press key <F9>, which equals the command **'Toggle Breakpoint'** from the Debug menu.

The breakpoint will be indicated.

If the application currently is in STOP state, this will look like:

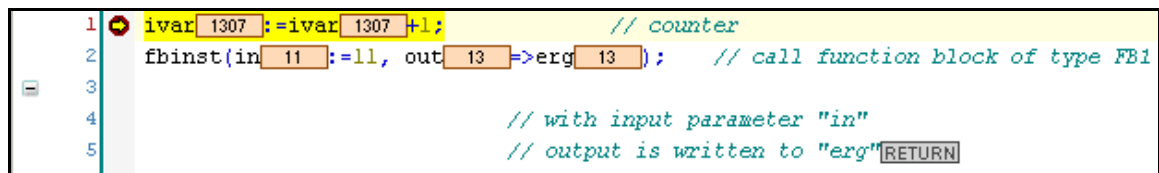


```

1  ● ivar 1307 :=ivar 1307 +1;           // counter
2  fbinst(in 11 :=11, out 13 =>erg 13 ); // call function block of type FB1
3
4                                     // with input parameter "in"
5                                     // output is written to "erg"RETURN

```

A running application will stop at a breakpoint:



```

1  ● ivar 1307 :=ivar 1307 +1;           // counter
2  fbinst(in 11 :=11, out 13 =>erg 13 ); // call function block of type FB1
3
4                                     // with input parameter "in"
5                                     // output is written to "erg"RETURN

```

Now you can step further by using <F8>, which represents command **Step Into** from the Debug menu and therefore will step also into the function block instance. To skip the steps of the function block use <F10> which equals the command **Step Over**. Each variable value currently read from the PLC will be displayed.

You might also have a look at the Breakpoints dialog to be opened via command **Breakpoints** from the View menu. Here, the breakpoints currently set can be viewed and edited and new breakpoints might be entered.

Regard also that the breakpoint positions will be remembered when you log out. They will be indicated by faded red bullets.

Now you have built and run your first project with CoDeSys 3.x. For further use of the programming system please see the online help.

We wish you successful programming. We appreciate each feedback.

7 Deinstallation, Modify, Repair

Deinstallation

Normally you can install new versions of CoDeSys without having to de-install the current version from your system. In case you want to de-install also execute Setup_CoDeSysV<version>.exe. In the start dialog of the installation wizard activate option 'Remove all installed features' and press 'Next'. Alternatively use the common deinstallation procedure via the System Control dialogs of your system.

Modify

If you want to modify the installation parameters, e.g. the selection of installed components, execute Setup_CoDeSysV<version>.exe and choose option 'Modify' in the installation wizard start dialog. Press Next to run through the installation dialogs and modify the settings, having made during the previous installation, as desired.

Repair

If for any reasons the installation is not proper, you might try to "repair" it. For this purpose execute Setup_CoDeSysV<version>.exe and choose option 'Repair' in the installation wizard start dialog. The installation will be done once more as configured during the previous run.

Change History

Version	Description	Editor	Date
0.1	Issued	MN	16.05.2006
0.2	Description of sample project	MN	26.01.2007
0.3	Update of sample project, Review	MN/MW	08.02.2007
0.4	Corrections acc. to Review	MN	09.02.2007
1.0	Release	MN	09.02.2007
2.0	Release after corrections in chap.6 concerning device name at several positions	MN	21.02.2007
3.0	Release after adaptations in chap.6.3 (default device) and 6.4 (monitoring possibilities)	MN	26.03.2007
4.0	Release after adaptations to CoDeSys V3.1.3.0 according to the German document version 4.0	MN	14.11.2007
4.1	Adaption to CoDeSys V3.2 acc. to online help	SE	20.02.2008
5.0	Formal Review, Release	MN	28.02.2008
6.0	Adaption to CoDeSys V3.2.0.4 acc. to online help, Release	MN	04.07.2008
7.0	Adaption to CoDeSys V3.4.0.0 acc. to online help, Release	MN	19.03.2010