Working with Automation Studio

TM210

Perfection in Automation
www.br-automation.com

Requirements

Trainingsmodule:	TM201 – The B&R product range
Software	Automation Studio 3.0.90 Automation Runtime 3.0.8
Hardware	X20CP1485

TABLE OF CONTENTS

1 INTRODUCTION

Automation Studio is a programming environment for the B&R automation components, which include the controller, motion control and visualization. The clear structure of projects and the ability to manage a wide range of configurations and machine variations makes working in teams that much easier.

The user can choose between programming languages, diagnostic tools and editors to create and manage projects with maximum efficiency. The use of standard libraries provided by B&R and IEC programming languages that are integrated in the system enable a highly efficient workflow.

The Automation Studio splash screen

This training module will use examples to demonstrate how to use the great many tools available in Automation Studio. The extensive Automation Studio Help system provides the basis for the examples and exercises in this training modules.

1.1 Training guide objectives

This training module will use examples to demonstrate how to use the great many tools available in Automation Studio. The extensive Automation Studio Help system provides the basis for the examples and exercises in this training modules.

You will learn how to…

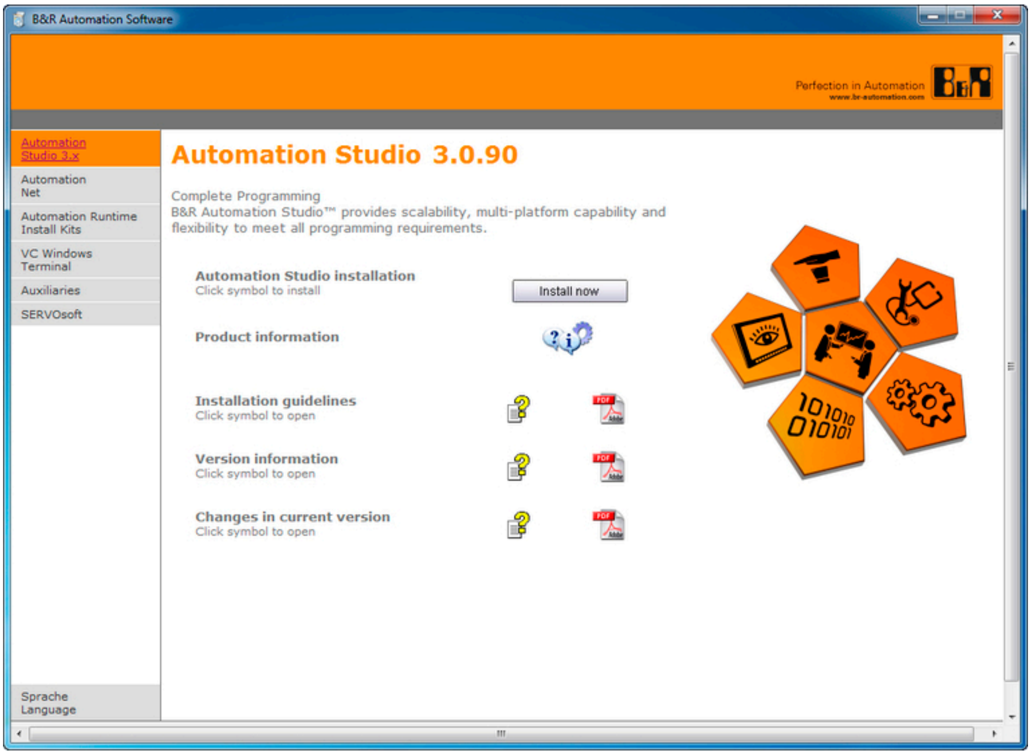
- Construct simple and complex machines and systems
- Use the tools for programming software elements
- Create simple programs
- Simulate and test sections of program

2 INSTALLATION

The Automation Studio installation is started by Autorun after inserting the DVD or by running **"Install.exe"** in Explorer.

2.1 Installation wizards

An Installation Wizard guides you through the installation of the necessary components. After selecting the language, the installation can be started in the subsequent dialog box. The version information can be opened from this page.



Automation Studio 3.0.90

Complete Programming
B&R Automation Studio™ provides scalability, multi-platform capability and flexibility to meet all programming requirements.

Automation Studio installation
Click symbol to install

Product information

Installation guidelines
Click symbol to open

Version information
Click symbol to open

Changes in current version
Click symbol to open

Automation Studio installation: Selecting the desired language

The system requirements for Automation Studio are listed in the installation guidelines, which can be opened by clicking on the corresponding icon.

> Depending on your PC and the Windows environment, installation of the selected components may take some time.

2.2 Licensing

Automation Studio license sticker

Each full version must be licensed after the initial installation.

This is done by entering the license or serial number in the licensing dialog box. The 11-digit serial number (beginning with 8986.. or A555..) can be found on a paper CD in the CD case for each full version.

The licensing window is displayed every time Automation Studio is started, and can also be opened later on from the 'About' window The full versions can each be licensed 5 times per serial number. After that, the serial number cannot be used for further licensing.

The full version for an unlimited number of workstations can be licensed an unlimited number of times.

Once the serial number has been entered in the licensing dialog box, the licensing can be automatically completed online if an Internet connection is present. If an Internet connection is not present, licensing can also be done manually (e.g. over the telephone by calling the respective B&R support hotline).

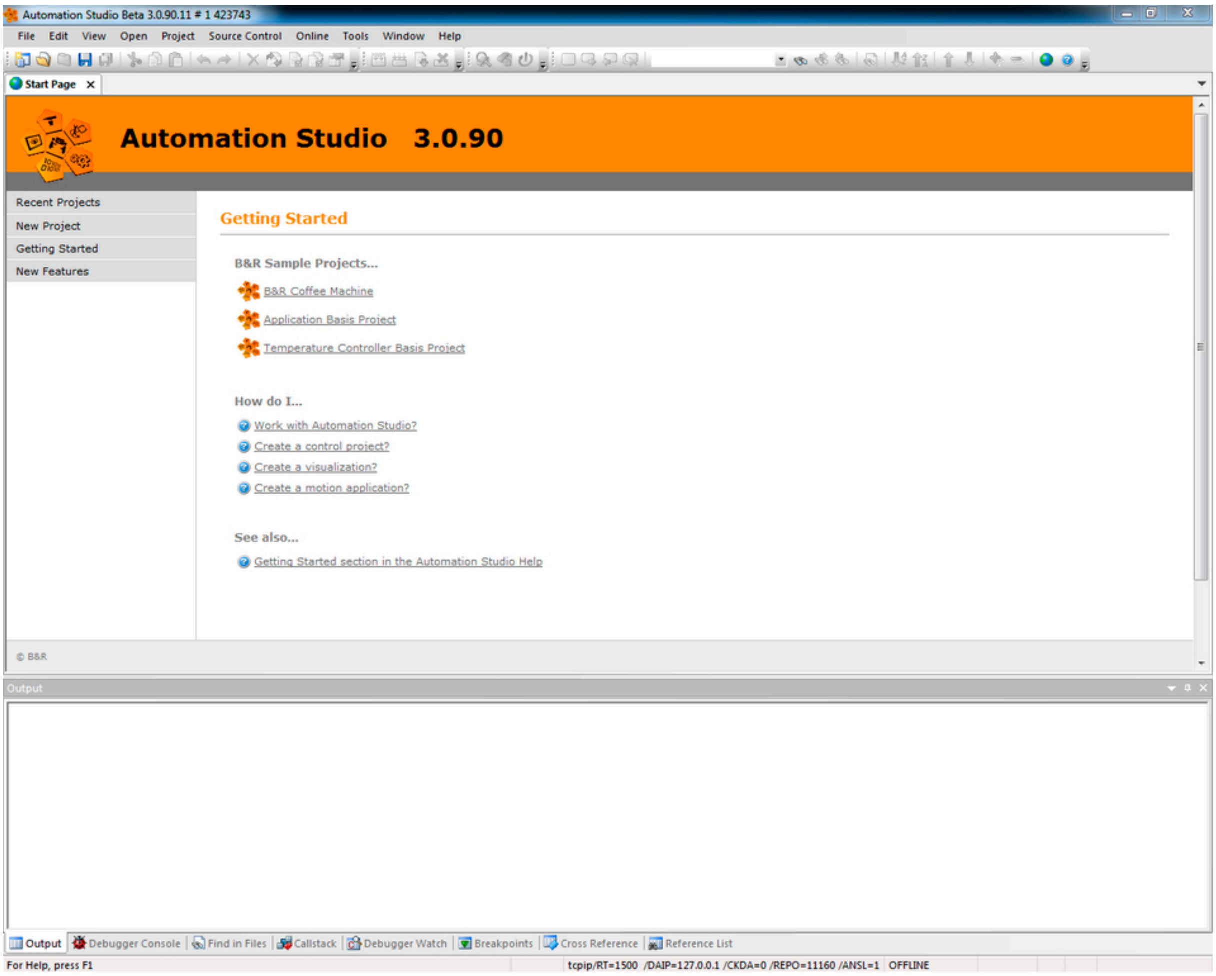
If the full version is not licensed, the program will be locked after 30 days of use.

3 STARTING AUTOMATION STUDIO

Installation creates an entry for Automation Studio in the Start menu. Automation Studio can now be launched from the Start menu or using the shortcut on your desktop.

3.1 The start page

When Automation Studio is started the first time, the start page is opened in the programming interface.


Automation Studio Beta 3.0.90.11 # 1 423743
File Edit View Open Project Source Control Online Tools Window Help
Start Page
Automation Studio 3.0.90
Recent Projects
New Project
Getting Started
New Features
Getting Started
B&R Sample Projects...
B&R Coffee Machine
Application Basis Project
Temperature Controller Basis Project
How do I...
Work with Automation Studio?
Create a control project?
Create a visualization?
Create a motion application?
See also...
Getting Started section in the Automation Studio Help
© B&R
Output
Output
Debugger Console
Find in Files
Callstack
Debugger Watch
Breakpoints
Cross Reference
Reference List
For Help, press F1
tcpip/RT=1500 /DAIP=127.0.0.1 /CKDA=0 /REPO=11160 /ANSL=1 OFFLINE

Automation Studio - Start Page

From the Start Page, you can create a new project or open an existing one. You can also jump to the Getting Started chapter of the Help system for an introduction to Automation Studio.

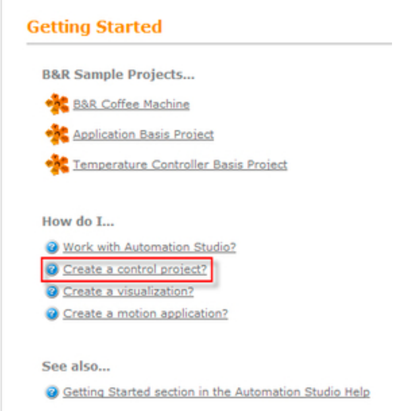
The start page also opens when you close a project, or you can open it any time by selecting **<Help> / <Show Start Page>** from the menu.

4 THE FIRST PROJECT

In this section we will use the Automation Studio Help system to create a new project, transfer it to the „PC based Runtime Simulation ARsim“ , and then test the program using Automation Studio.

Create your first project using the Help system

You can open the right chapter of the Automation Studio Help system directly from the start page.



Getting Started

B&R Sample Projects...

- B&R Coffee Machine
- Application Basis Project
- Temperature Controller Basis Project

How do I...

- Work with Automation Studio?
- Create a control project?
- Create a visualization?
- Create a motion application?

See also...

- Getting Started section in the Automation Studio Help

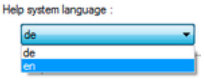
Automation Desktop start page

1) From the start page, open the chapter „**How do I create a controlproject?**“

2) Select the first sub-section „First project with PC based Simulation Runtime“

3) Work through each of the steps

Under **<Tools> / <Options>** in the menu you can set the language of the Automation Studio Online Help (German/English).

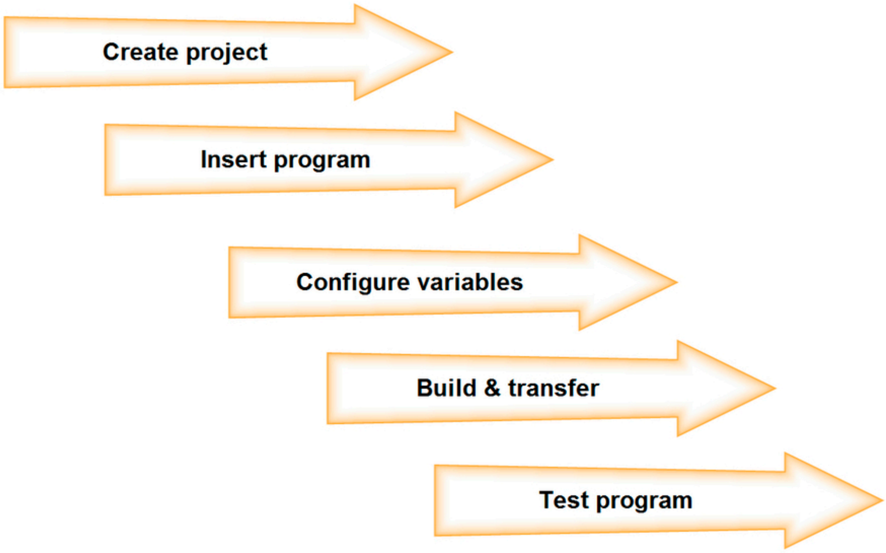


Help system language : de

de
en

Selecting the help language

With the aid of the Online Help system, you were able to create your first project, write a program, transfer it to a simulation system and test it. In the process of creating your first project, you already used several elements of Automation Studio. In the next few chapters, we will explain the structure of an Automation Studio project using a sample project.



Procedure for creating your first project

On PCs with a second monitor, the Online Help and Automation Studio can be displayed at the same time. If there is no second monitor available, you can use the shortcut **<ALT> + <TAB>** to switch back and forth between them.

5 AUTOMATION STUDIO

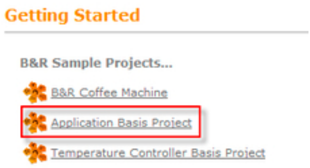
A sample project is used to illustrate the structure of an Automation Studio project.

5.1 The "CoffeeMachine" sample project

The installation of Automation Studio includes several sample projects. One of these sample projects will be used in this training module to illustrate the Automation Studio environment and its many functions.

Sample Automation Studio project - "CoffeeMachine"

The sample project can be opened from the start page.



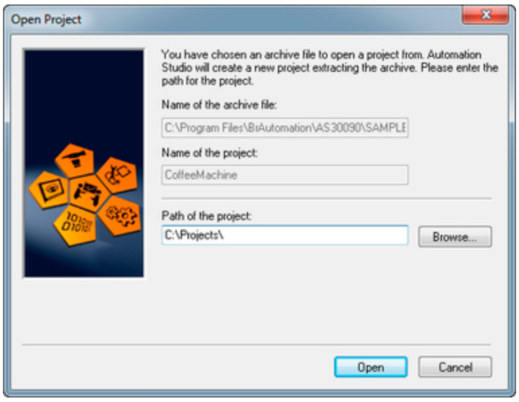
Getting Started

B&R Sample Projects...

- B&R Coffee Machine
- Application Basis Project
- Temperature Controller Basis Project

Selecting the sample project

1) Open the sample Automation Studio project from the start page.



Open Project

You have chosen an archive file to open a project from. Automation Studio will create a new project extracting the archive. Please enter the path for the project.

Name of the archive file: C:\Program Files\BrAutomation\AS30090\SAMPLE

Name of the project: CoffeeMachine

Path of the project: C:\Projects\

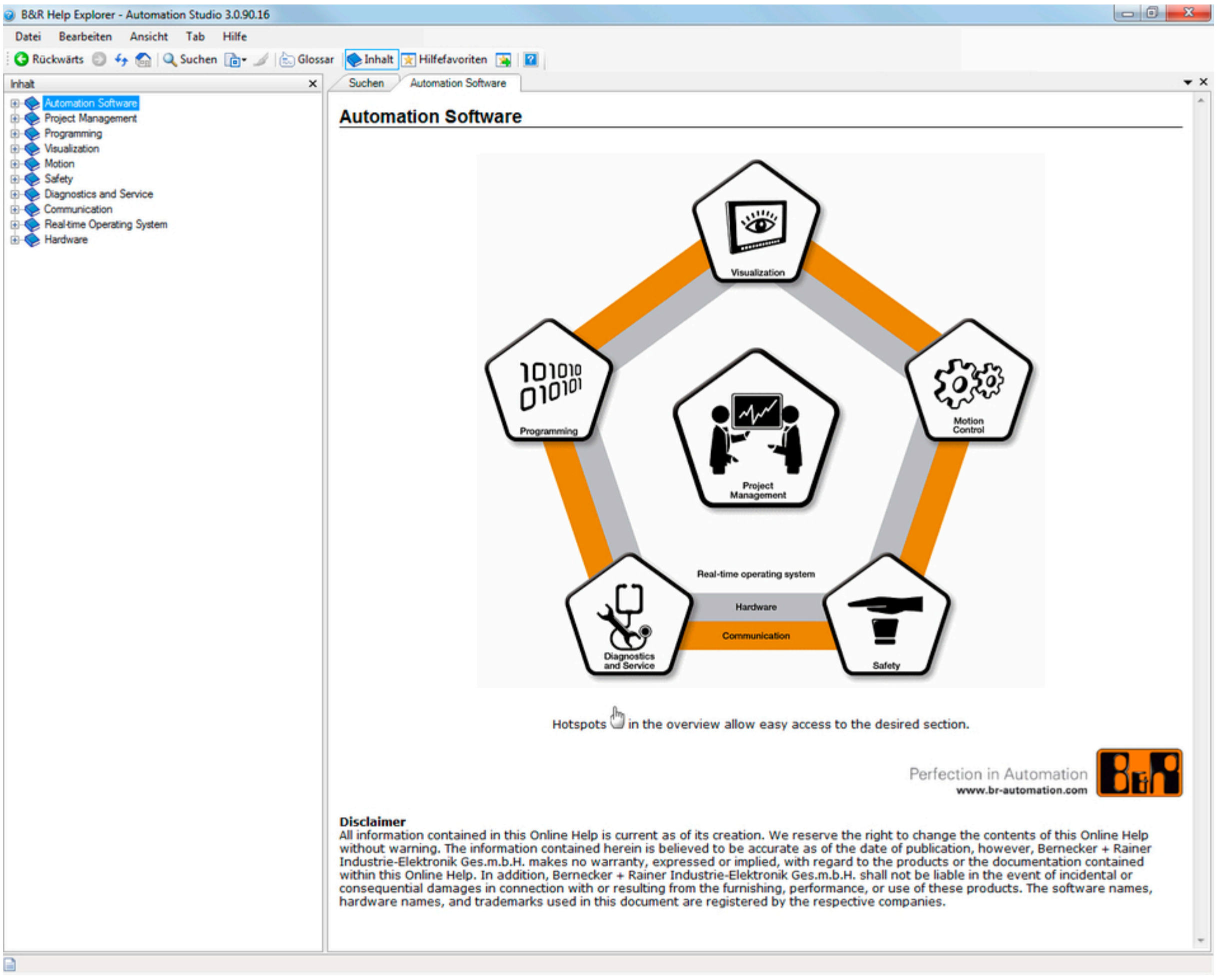
Open project with Automation Studio

5.2 Automation Studio Help

The Automation Studio Help system is your constant companion throughout the development, configuration and commissioning of a project. It serves as a reference guide for operating Automation Studio and its editors, for creating a program or visualization application, for configuring drives, and also provides access to all the hardware documentation.

Using the Automation Studio Help system

While working in Automation Studio, press the **<F1>** key to display help information on the selected element. Use the search function to find entries on a specific topic. The functions of the Help system are described in the chapter **Automation Software – How do I use the Help system**.



Automation Software

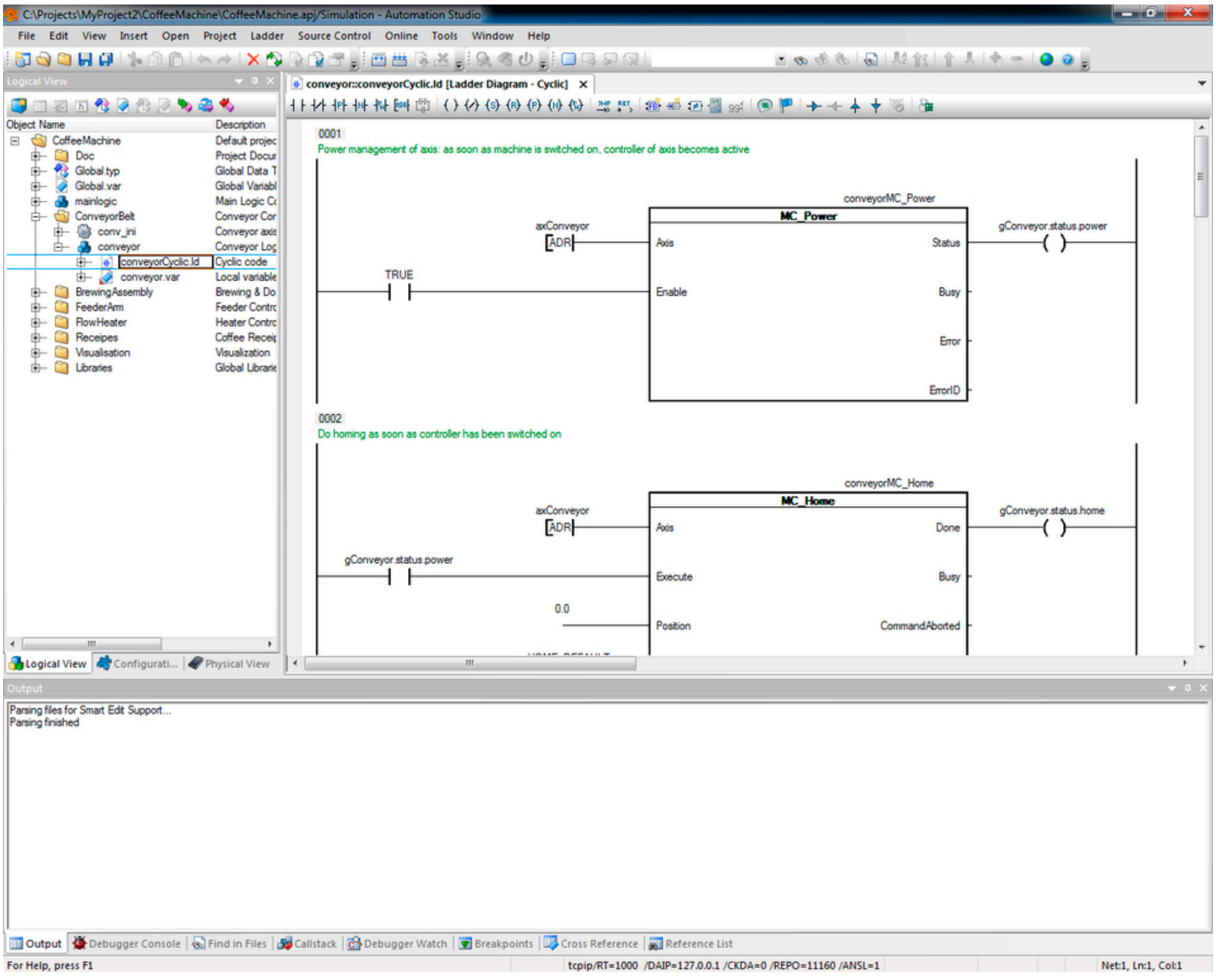
Hotspots in the overview allow easy access to the desired section.

Disclaimer
All information contained in this Online Help is current as of its creation. We reserve the right to change the contents of this Online Help without warning. The information contained herein is believed to be accurate as of the date of publication, however, Bernecker + Rainer Industrie-Elektronik Ges.m.b.H. makes no warranty, expressed or implied, with regard to the products or the documentation contained within this Online Help. In addition, Bernecker + Rainer Industrie-Elektronik Ges.m.b.H. shall not be liable in the event of incidental or consequential damages in connection with or resulting from the furnishing, performance, or use of these products. The software names, hardware names, and trademarks used in this document are registered by the respective companies.

Automation Studio Help

5.3 The workspace

Automation Studio is divided into various sections, each with a specific function or task.



Object Name	Description
CoffeeMachine	Default projec
Doc	Project Docur
Global.typ	Global Data T
Global.var	Global Variabl
mainlogic	Main Logic Cc
ConveyorBelt	Conveyor Cor
conv_ini	Conveyor axis
conveyor	Conveyor Log
conveyorCyclic.ld	Cyclic code
conveyor.var	Local variable
BrewingAssembly	Brewing & Do
FeederArm	Feeder Contrc
FlowHeater	Heater Contrc
Receipes	Coffee Receip
Visualisation	Visualization
Libraries	Global Librarie

0001
Power management of axis: as soon as machine is switched on, controller of axis becomes active

0002
Do homing as soon as controller has been switched on

```
Parsing files for Smart Edit Support...
Parsing finished
```

For Help, press F1 tcpip/RT=1000 /DAIP=127.0.0.1 /CKDA=0 /REPO=11160 /ANSL=1 Net:1, Ln:1, Col:1

The Automation Studio workspace

- The menus and the task bar provide access to the extensive Automation Studio functions.
- On the left side of the screen, you see the project explorer, which contains tabs that allow you to create, manage and edit the components of your project.
- On the right side, the editors are displayed in one or more workbooks. This is where you can program and configure your project data.
- At the bottom of the workspace you'll find the output window, which displays information such as messages while compiling your project.

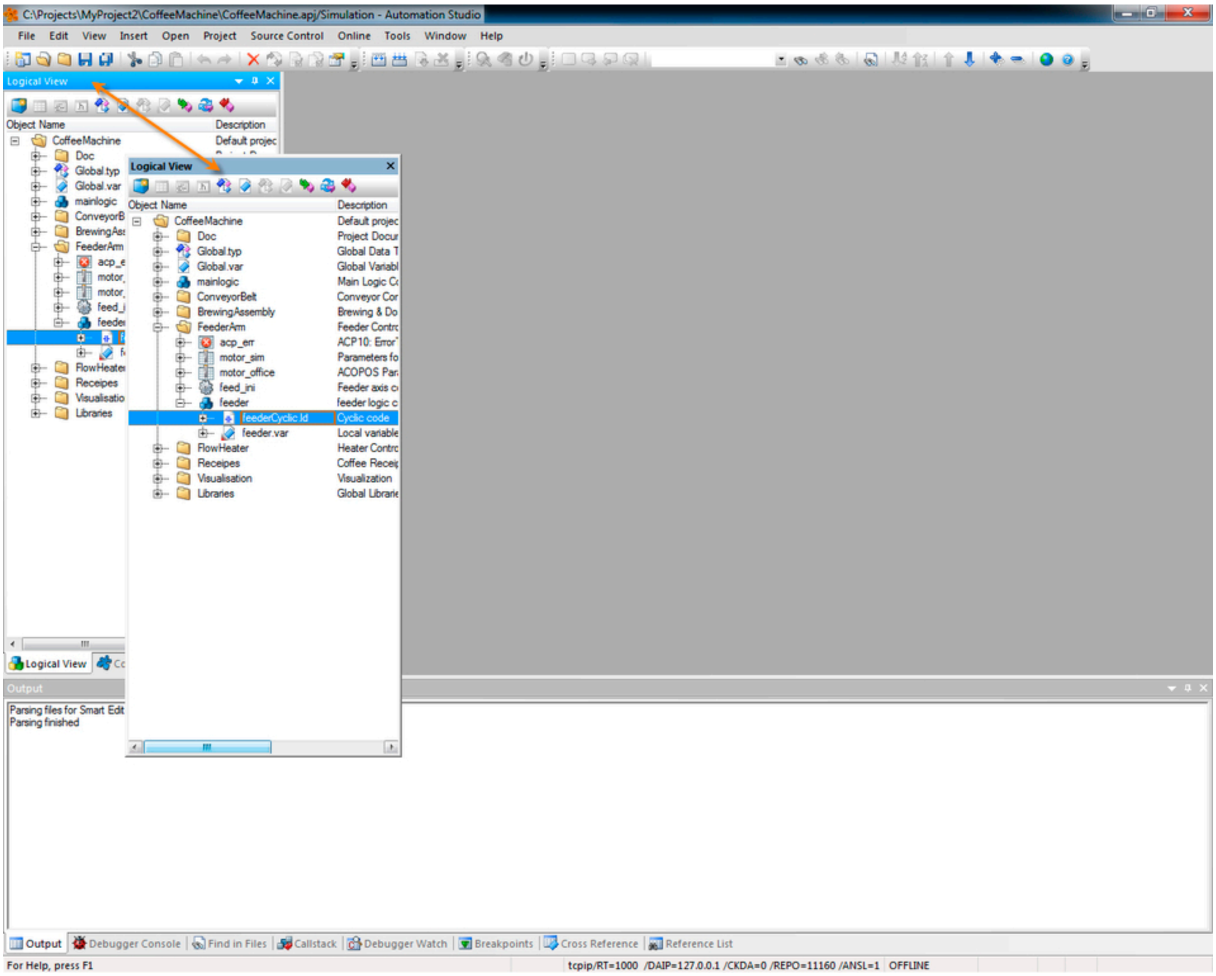
? Project Management / The workspace

Working with Automation Studio

The goal of this exercise is to become familiar with the Automation Studio environment.

5.3.1 Changing the positions of windows

When you open a project, the windows are docked on the main Automation Studio window. The windows can be undocked and moved by clicking and dragging the title bar.



```
Parsing files for Smart Edit
Parsing finished
```

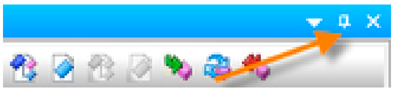
```
For Help, press F1        tcpip/RT=1000 /DAIP=127.0.0.1 /CKDA=0 /REPO=11160 /ANSL=1  OFFLINE
```

Docking and undocking windows

Project Management / The workspace / Project explorer

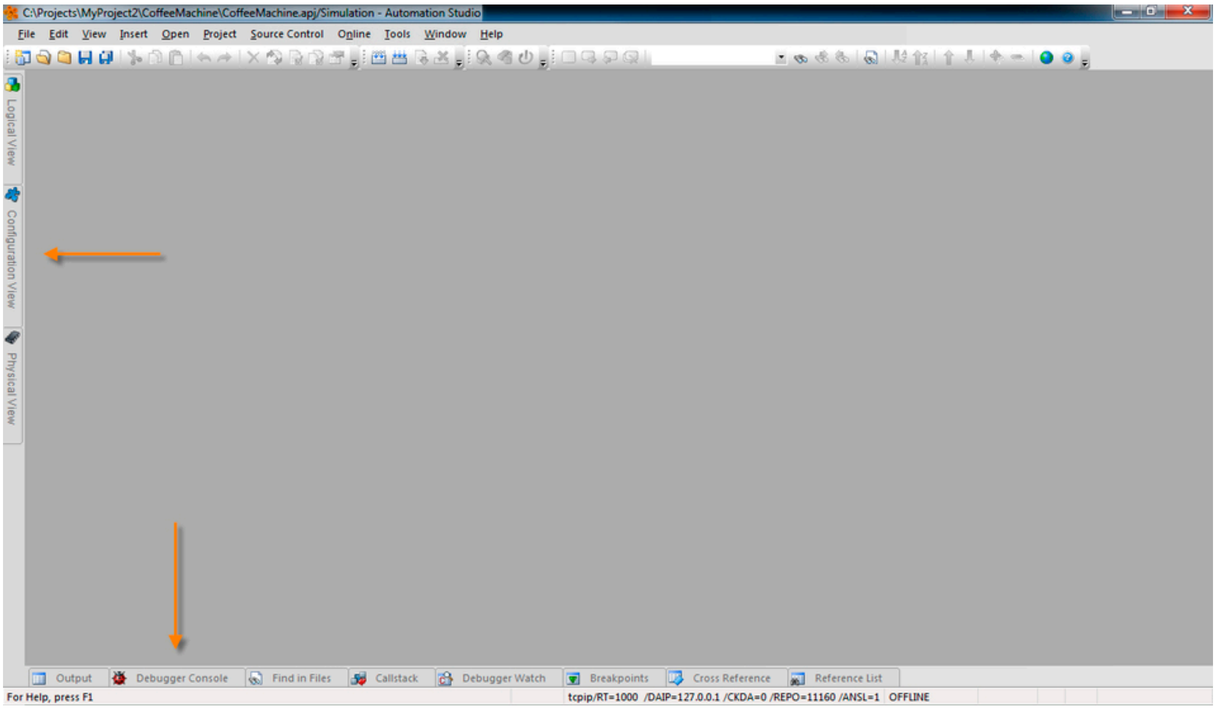
5.3.2 Automatic hiding of windows

To create more space on the monitor when working with other editors (e.g. graphic programming editors), the project explorer and output window can be hidden automatically.



Setting with the pin symbol

Simply click on the pin icon in the title bar of the project explorer or the output window.

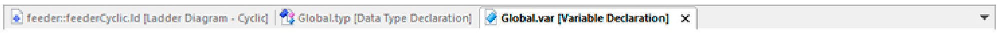

C:\Projects\MyProject2\CoffeeMachine\CoffeeMachine.apj/Simulation - Automation Studio
File Edit View Insert Open Project Source Control Online Tools Window Help
Logical View
Configuration View
Physical View
Output
Debugger Console
Find in Files
Callstack
Debugger Watch
Breakpoints
Cross Reference
Reference List
For Help, press F1
tcpip/RT=1000 /DAIP=127.0.0.1 /CKDA=0 /REPO=11160 /ANSL=1 OFFLINE

Hiding the project explorer and output window

? Project Management / The workspace / Project explorer

5.3.3 Workbooks

Workbook mode presents a clear way of displaying open windows while making it much easier to switch between them. Depending on the configured mode, the windows can also overlap or be arranged next to or over one another.

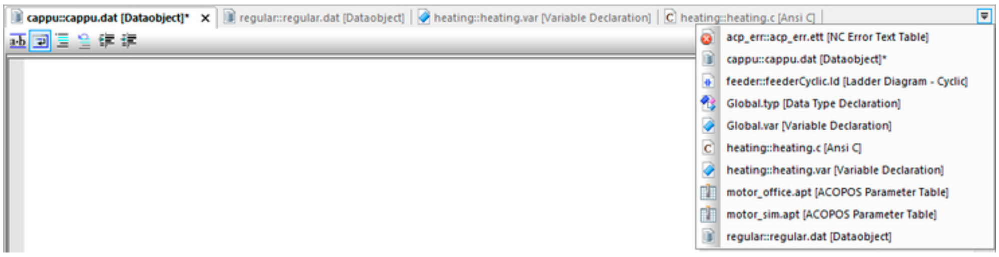


feeder::feederCyclic.ld [Ladder Diagram - Cyclic] | Global.typ [Data Type Declaration] | Global.var [Variable Declaration] ×

Workbook

If multiple editors are opened, each is opened in a separate workbook. If multiple workbooks are open, the resolution of the display may not allow them to be displayed side-by-side.

To maintain an overview of the opened workbooks, a list of them opens if you press the drop-down symbol on the right side of the title bar.



cappu::cappu.dat [Dataobject]* × | regular::regular.dat [Dataobject] | heating::heating.var [Variable Declaration] | heating::heating.c [Ansi C]

- acp_err::acp_err.ett [NC Error Text Table]
- cappu::cappu.dat [Dataobject]*
- feeder::feederCyclic.ld [Ladder Diagram - Cyclic]
- Global.typ [Data Type Declaration]
- Global.var [Variable Declaration]
- heating::heating.c [Ansi C]
- heating::heating.var [Variable Declaration]
- motor_office.apt [ACOPOS Parameter Table]
- motor_sim.apt [ACOPOS Parameter Table]
- regular::regular.dat [Dataobject]

Overview of open workbooks

Project Management / The workspace / Workbook mode

5.3.4 Menu and shortcut menu

The main menu provides access to all of the functions in Automation Studio.

Depending on the context (active editor or window), individual menu items may be shown/hidden or enabled/disabled. This allows only the functions available in the current context to be offered.

Many of the functions that can be reached from main menu can also be carried out using toolbars, shortcut keys or the shortcut menu (right click).

Project management / Workspace / Menus

Project Management / Workspace /

Project management toolbars / Workspace / Key combinations

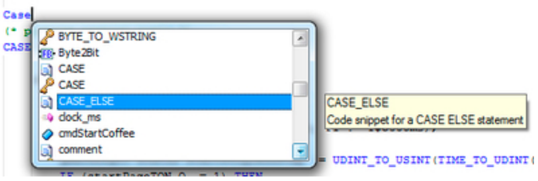
5.3.5 Convenient operation during programming - SmartEdit

The SmartEdit feature combines a series of functions that support the intelligent input of information in Automation Studio editors.

The shortcut **<TAB>** automatically completes partially entered terms:

This is supportet for:

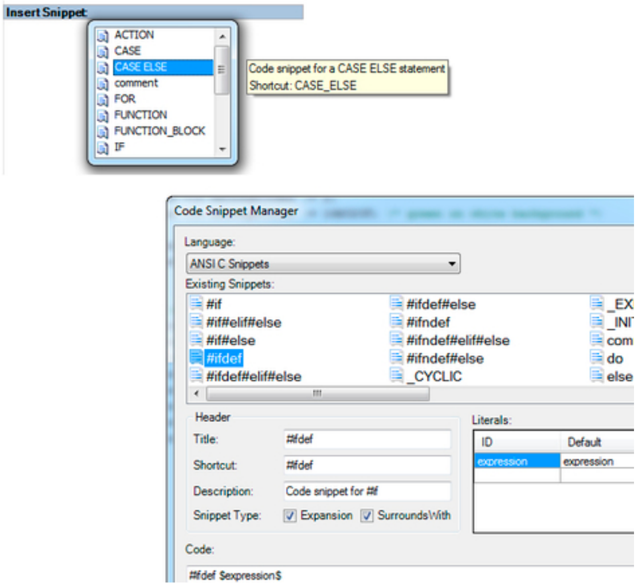
- Variable names and structure members
- Function name
- Language constructs (IF THEN, CASE, etc.)



CASE_ELSE
Code snippet for a CASE ELSE statement

Smart Edit

With the shortcut **<CTRL> + <q>, <k>** the Code Snippets can be inserted. It is this ready-made source code parts, which can manage the users in the Code Snippet Manager itself.



Code snippet for a CASE ELSE statement
Shortcut: CASE_ELSE

Code Snippet Manager

Language: ANSI C Snippets

Header

Title: #ifdef

Shortcut: #ifdef

Description: Code snippet for #if

Snippet Type: ☑ Expansion ☑ SurroundsWith

Literals:

ID	Default
expression	expression

Code:

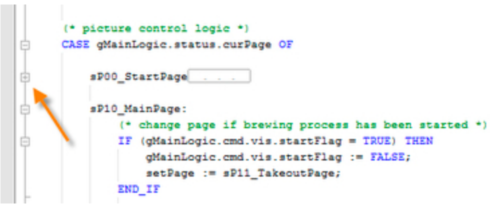
```
#ifdef $expression$
```

Inserting a Code Snippet and the Code Snippet Manager

? Programming \ Editors \ General operations \ SmartEdit

Other editing functions facilitate the overview in the program code:

- Fast Navigation
- Tooltips
- Syntax coloring
- Coloring changed lines of code
- Collapse and expand code segments



```
(* picture control logic *)
CASE gMainLogic.status.curPage OF

    sP00_StartPage ...

    sP10_MainPage:
        (* change page if brewing process has been started *)
        IF (gMainLogic.cmd.vis.startFlag = TRUE) THEN
            gMainLogic.cmd.vis.startFlag := FALSE;
            setPage := sP11_TakeoutPage;
        END_IF
```

Collapse and expand code segments

5.4 Basic concept

The development of the software in an Automation Studio project can be arranged according to the structure of the machine.

This makes the organization of the software easy to follow, because there is a direct relationship between program and machine.

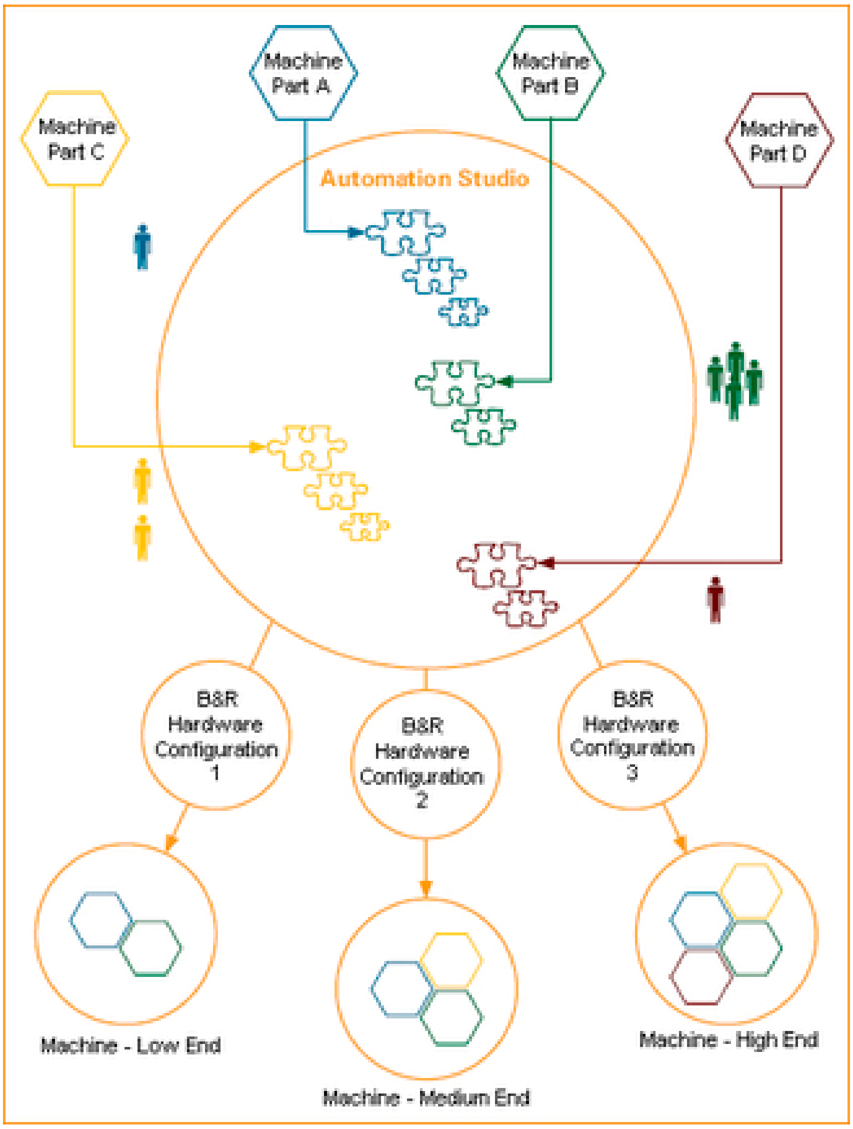


Fig. Structure and principle of Automation Studio

Once they are programmed and configured, software components can be assigned to different hardware configurations.

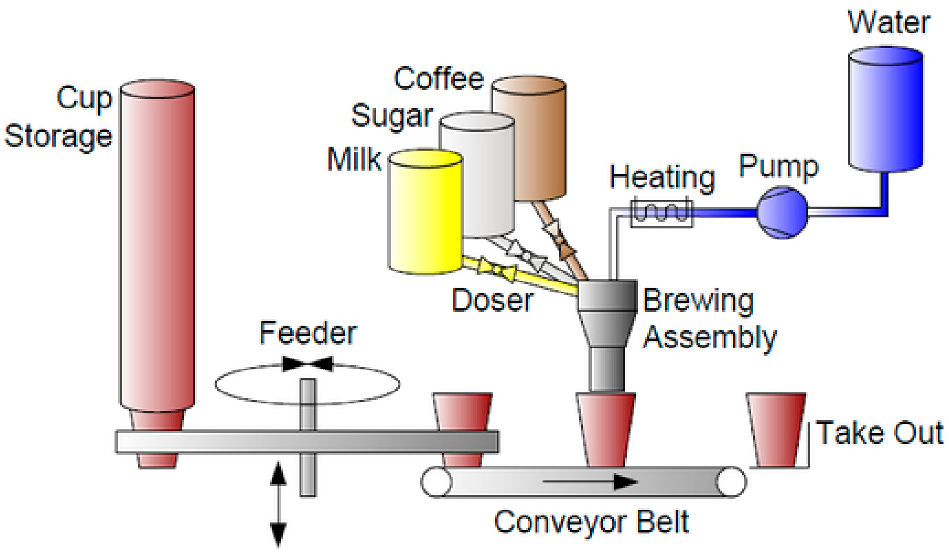
This allows multiple versions of a machine type to be managed under a single project, each with a different configuration of software and hardware.

5.5 Creating a concept

The "CoffeeMachine" sample can be used to illustrate the path from an initial concept to a full representation of the software and hardware components in Automation Studio.

5.5.1 System overview

The system (machine) its functional units are the foundation for representing the software structure in Automation Studio.



System overview

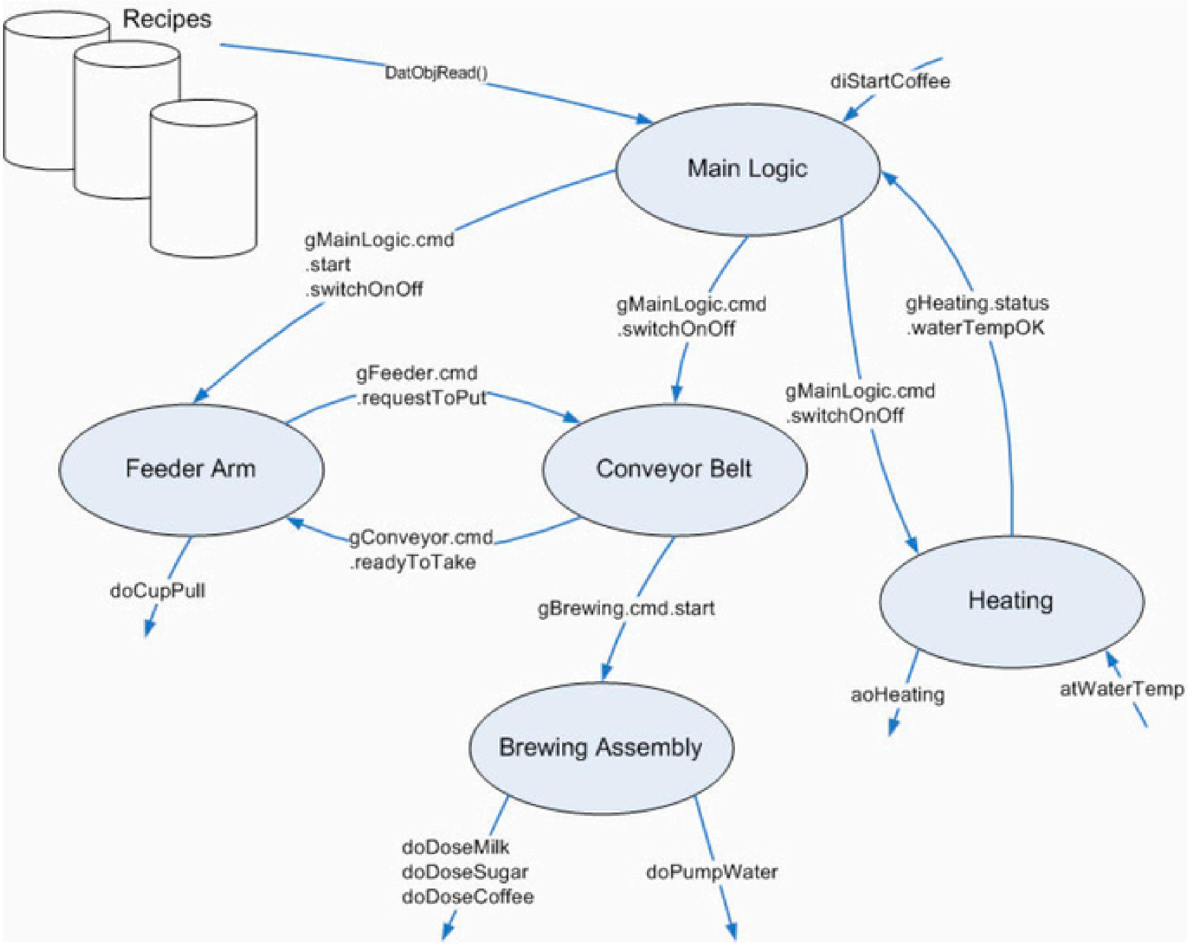
The basic process can be outlined as follows:

- The cup extractor takes a cup and places it on the conveyor.
- The cup is moved to a defined filling position.
- A dosing unit defines the amounts of sugar, milk and coffee for the specified type of coffee.
- The water is heated and pumped into the cup.
- The conveyor moves the cup to the dispense position.

Programming / Structured software development

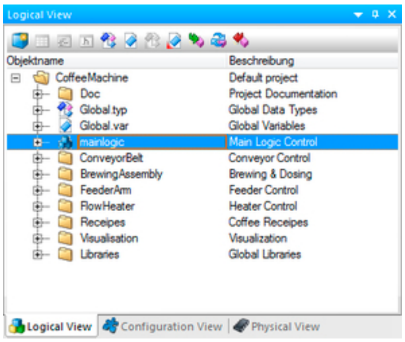
5.5.2 Function description

The software is structured according to the preparation process.



Function description of a program

The individual functions can be represented in Automation Studio using clearly organized configuration and programming.



Objektname	Beschreibung
CoffeeMachine	Default project
Doc	Project Documentation
Global.typ	Global Data Types
Global.var	Global Variables
mainlogic	Main Logic Control
ConveyorBelt	Conveyor Control
BrewingAssembly	Brewing & Dosing
FeederArm	Feeder Control
FlowHeater	Heater Control
Receipes	Coffee Receipes
Visualisation	Visualization
Libraries	Global Libraries

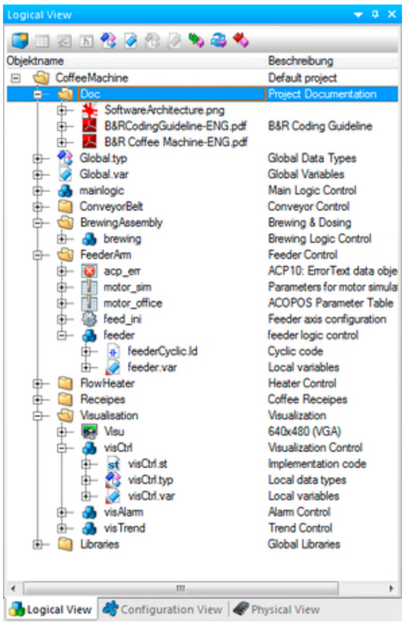
Organization of software in Automation Studio

5.6 Organization of software

Program organization elements (POEs) are displayed in the **Logical View** arranged in a tree structure.

The elements are organized into packages (directories). A package can contain, for example, all the software elements needed for a particular part of the system, as well as all the corresponding documentation.

In this view there is not a direct correlation to the actual hardware. It only serves to organize and manage the program organization elements.



Objektname	Beschreibung
CoffeeMachine	Default project
Doc	Project Documentation
SoftwareArchitecture.png	
B&RCodingGuideline-ENG.pdf	B&R Coding Guideline
B&R Coffee Machine-ENG.pdf	
Global.typ	Global Data Types
Global.var	Global Variables
mainlogic	Main Logic Control
ConveyorBelt	Conveyor Control
BrewingAssembly	Brewing & Dosing
brewing	Brewing Logic Control
FeederArm	Feeder Control
acp_err	ACP10: ErrorText data obje
motor_sim	Parameters for motor simula
motor_office	ACOPOS Parameter Table
feed_ini	Feeder axis configuration
feeder	feeder logic control
feederCyclic.ld	Cyclic code
feeder.var	Local variables
FlowHeater	Heater Control
Receipes	Coffee Receipes
Visualisation	Visualization
Visu	640x480 (VGA)
visCtrl	Visualization Control
visCtrl.st	Implementation code
visCtrl.typ	Local data types
visCtrl.var	Local variables
visAlarm	Alarm Control
visTrend	Trend Control
Libraries	Global Libraries

Logical view

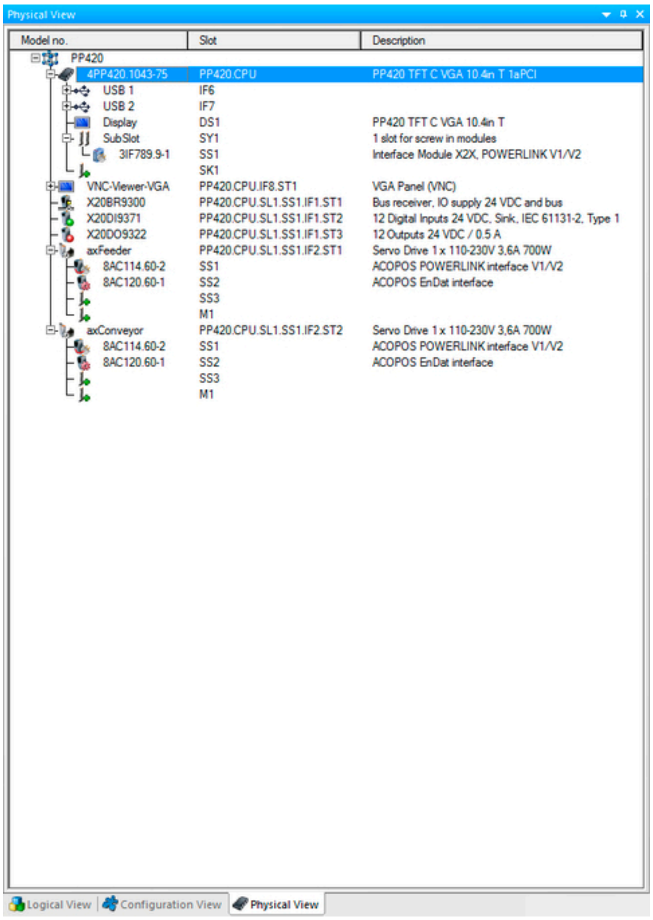
?

Project Management / Logical View

5.7 Organization of hardware

The hardware components of a system are configured and managed in the **Physical View**.

When a new project or system configuration is created, (see 5.8 "Managing the system configuration") the node for the CPU is determined by selecting the hardware (e.g. 4PP420.1043-B5).



Model no.	Slot	Description
PP420		
4PP420.1043-75	PP420.CPU	PP420 TFT C VGA 10.4in T 1aPCI
USB 1	IF6	
USB 2	IF7	
Display	DS1	PP420 TFT C VGA 10.4in T
SubSlot	SY1	1 slot for screw in modules
3IF789.9-1	SS1	Interface Module X2X, POWERLINK V1/V2
	SK1	
VNC-Viewer-VGA	PP420.CPU.IF8.ST1	VGA Panel (VNC)
X20BR9300	PP420.CPU.SL1.SS1.IF1.ST1	Bus receiver, IO supply 24 VDC and bus
X20DI9371	PP420.CPU.SL1.SS1.IF1.ST2	12 Digital Inputs 24 VDC, Sink, IEC 61131-2, Type 1
X20DO9322	PP420.CPU.SL1.SS1.IF1.ST3	12 Outputs 24 VDC / 0.5 A
axFeeder	PP420.CPU.SL1.SS1.IF2.ST1	Servo Drive 1x 110-230V 3,6A 700W
8AC114.60-2	SS1	ACOPOS POWERLINK interface V1/V2
8AC120.60-1	SS2	ACOPOS EnDat interface
	SS3	
	M1	
axConveyor	PP420.CPU.SL1.SS1.IF2.ST2	Servo Drive 1x 110-230V 3,6A 700W
8AC114.60-2	SS1	ACOPOS POWERLINK interface V1/V2
8AC120.60-1	SS2	ACOPOS EnDat interface
	SS3	
	M1	

Physical View of the PP420 configuration

The following tasks are performed in the Physical View:

- Inserting and configuring I/O modules
- Assigning variables to I/O data points
- Configuring fieldbus modules and interfaces in a network
- Inserting and configuring hardware modules (e.g. terminals, fieldbus devices)
- Opening a software configuration

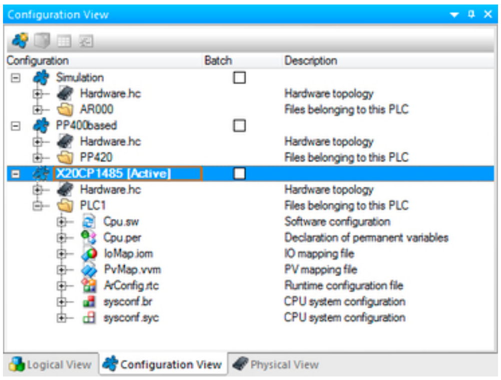
? Project Management / Physical View

5.8 Managing the system configuration

The various configurations of a machine type are managed in the **Configuration View**.

The configurations can differ in what software and hardware is used.

As the "CoffeeMachine" example shows, this project has two different configurations. The first is purely a simulation environment, while the other corresponds to the actual hardware structure.



Configuration	Batch	Description
Simulation	□	
Hardware.hc		Hardware topology
AR000		Files belonging to this PLC
PP400based	□	
Hardware.hc		Hardware topology
PP420		Files belonging to this PLC
X20CP1485 [Active]	□	
Hardware.hc		Hardware topology
PLC1		Files belonging to this PLC
Cpu.sw		Software configuration
Cpu.per		Declaration of permanent variables
IoMap.iom		IO mapping file
PvMap.vvm		PV mapping file
ArConfig.rtc		Runtime configuration file
sysconf.br		CPU system configuration
sysconf.syc		CPU system configuration

Configuration View with active PP420 configuration

A configuration consists of hardware and software. Only one configuration may be **[Active]** at a time.

When you activate a configuration by double-clicking on it, the hardware assigned to the configuration is displayed in the Physical View.

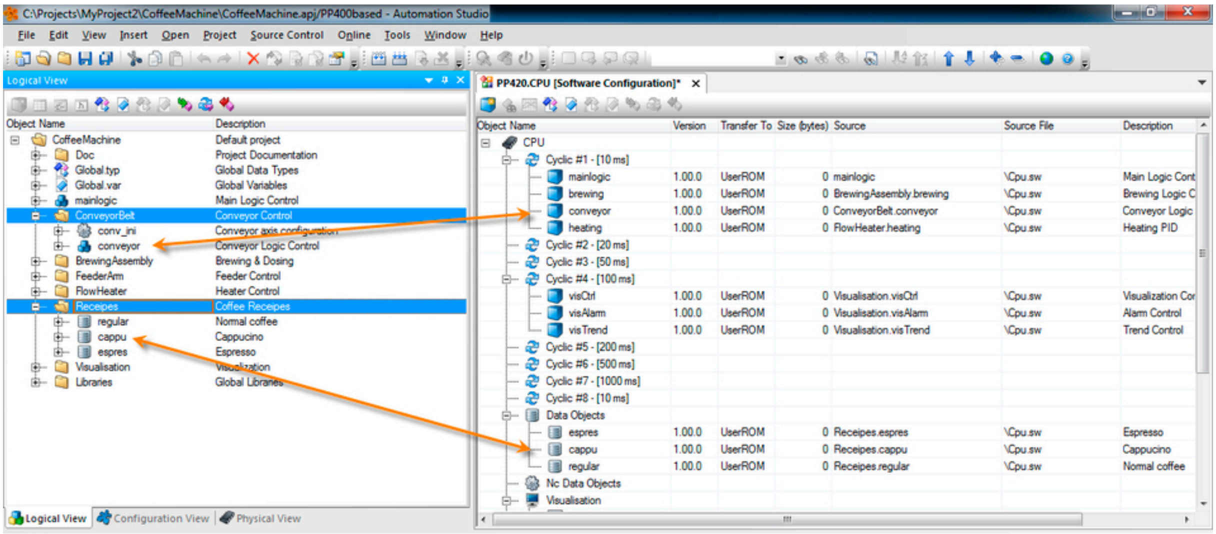
Automation Studio Help: Project Management / Configuration View

5.9 Relationship between functionality and task

The software elements of the currently active configuration (CPU) that can be transferred to the target system are displayed in the Logical View.

There are two ways to assign a software element to a configuration:

- Create a software element with automatic assignment
- Assign an existing software element to a new configuration


C:\Projects\MyProject2\CoffeeMachine\CoffeeMachine.apj/PP400based - Automation Studio
File Edit View Insert Open Project Source Control Online Tools Window Help
Logical View
PP420.CPU [Software Configuration]*
Logical View
Configuration View
Physical View

Object Name	Description
CoffeeMachine	Default project
Doc	Project Documentation
Global.typ	Global Data Types
Global.var	Global Variables
mainlogic	Main Logic Control
ConveyorBelt	Conveyor Control
conv_ini	Conveyor axis configuration
conveyor	Conveyor Logic Control
BrewingAssembly	Brewing & Dosing
FeederArm	Feeder Control
FlowHeater	Heater Control
Receipes	Coffee Receipes
regular	Normal coffee
cappu	Cappucino
espres	Espresso
Visualisation	Visualization
Libraries	Global Libraries

Object Name	Version	Transfer To	Size (bytes)	Source	Source File	Description
CPU						
Cyclic #1 - [10 ms]						
mainlogic	1.00.0	UserROM	0	mainlogic	\Cpu.sw	Main Logic Cont
brewing	1.00.0	UserROM	0	BrewingAssembly.brewing	\Cpu.sw	Brewing Logic C
conveyor	1.00.0	UserROM	0	ConveyorBelt.conveyor	\Cpu.sw	Conveyor Logic
heating	1.00.0	UserROM	0	FlowHeater.heating	\Cpu.sw	Heating PID
Cyclic #2 - [20 ms]						
Cyclic #3 - [50 ms]						
Cyclic #4 - [100 ms]						
visCtrl	1.00.0	UserROM	0	Visualisation.visCtrl	\Cpu.sw	Visualization Co
visAlarm	1.00.0	UserROM	0	Visualisation.visAlarm	\Cpu.sw	Alarm Control
visTrend	1.00.0	UserROM	0	Visualisation.visTrend	\Cpu.sw	Trend Control
Cyclic #5 - [200 ms]						
Cyclic #6 - [500 ms]						
Cyclic #7 - [1000 ms]						
Cyclic #8 - [10 ms]						
Data Objects						
espres	1.00.0	UserROM	0	Receipes.espres	\Cpu.sw	Espresso
cappu	1.00.0	UserROM	0	Receipes.cappu	\Cpu.sw	Cappucino
regular	1.00.0	UserROM	0	Receipes.regular	\Cpu.sw	Normal coffee
Nc Data Objects						
Visualisation						

Software assignments

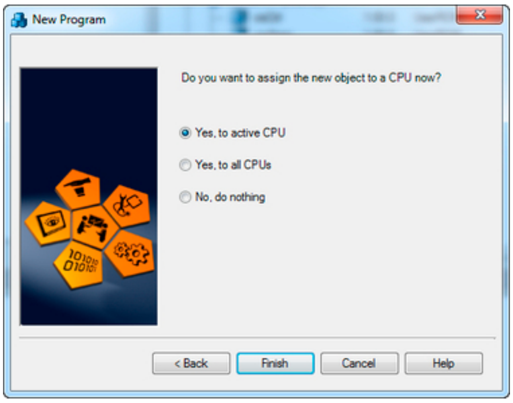
? Project Management / Logical View / Automatic software assignments

5.9.1 Open the software configuration

The software configuration can be opened by double-clicking on the CPU entry in the Physical View, or by selecting **<Open> / <Software configuration>** in the menu.

5.9.2 Assignment when creating

If you create a software element using the Automation Studio Wizard, the last dialog box gives you the chance to decide how it is added to the software configuration.


New Program

Do you want to assign the new object to a CPU now?

● Yes, to active CPU

○ Yes, to all CPUs

○ No, do nothing

Assigning the software to one or more CPUs

You can assign them to the active configuration (Yes, to active CPU) or to all CPUs (Yes, to all CPUs).

In these cases, Automation Studio automatically inserts the software elements in the default position in the software configuration.

The order and priority of programs in the software configuration can be changed by moving them to the different task classes.

Real-time Operating System / Method of operation / Runtime performance / Task classes

5.9.3 Assignment to a new configuration

When a new hardware configuration is added to a project, the software elements must eb added to the software configuration manually.

Create a new configuration

Create a new configuration in the "CoffeeMachine" project with an X20 CPU (X20CP1485)

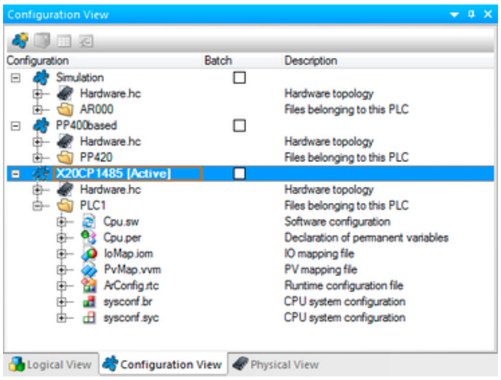
1) Create a new configuration called "X20CP1485"

2) Adding the required hardware

3) Open the software configuration

4) Assigning the software

The goal of this exercise not to create a fully functional configuration. It's purpose is only to show how software elements can be assigned to the new configuration.



Configuration	Batch	Description
Simulation	□	
Hardware.hc		Hardware topology
AR000		Files belonging to this PLC
PP400based	□	
Hardware.hc		Hardware topology
PP420		Files belonging to this PLC
X20CP1485 [Active]	□	
Hardware.hc		Hardware topology
PLC1		Files belonging to this PLC
Cpu.sw		Software configuration
Cpu.per		Declaration of permanent variables
IoMap.iom		IO mapping file
PvMap.vvm		PV mapping file
Arconfig.rtc		Runtime configuration file
sysconf.br		CPU system configuration
sysconf.syc		CPU system configuration

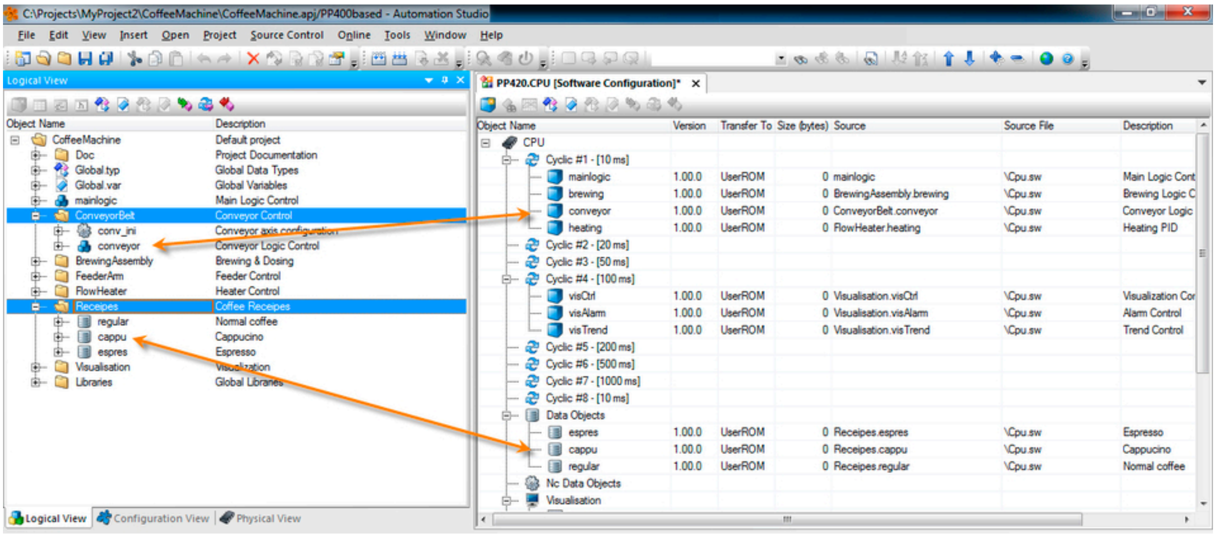
Inserting a new configuration

The steps for inserting a new configuration are the same as for creating a new project.

5.9.4 Manual assignment

With the software configuration open select the Logical View in the Project Explorer.

Drag and drop and object from the Logical View to the correct position in the software configuration.



C:\Projects\MyProject2\CoffeeMachine\CoffeeMachine.apj/PP400based - Automation Studio
File
Edit
View
Insert
Open
Project
Source Control
Online
Tools
Window
Help
Logical View
PP420.CPU [Software Configuration]*
Logical View
Configuration View
Physical View

Object Name	Description
CoffeeMachine	Default project
Doc	Project Documentation
Global.typ	Global Data Types
Global.var	Global Variables
mainlogic	Main Logic Control
ConveyorBelt	Conveyor Control
conv_ini	Conveyor axis configuration
conveyor	Conveyor Logic Control
BrewingAssembly	Brewing & Dosing
FeederArm	Feeder Control
FlowHeater	Heater Control
Receipes	Coffee Receipes
regular	Normal coffee
cappu	Cappucino
espres	Espresso
Visualisation	Visualization
Libraries	Global Libraries

Object Name	Version	Transfer To	Size (bytes)	Source	Source File	Description
CPU						
Cyclic #1 - [10 ms]						
mainlogic	1.00.0	UserROM	0	mainlogic	\Cpu.sw	Main Logic Cont
brewing	1.00.0	UserROM	0	BrewingAssembly.brewing	\Cpu.sw	Brewing Logic C
conveyor	1.00.0	UserROM	0	ConveyorBelt.conveyor	\Cpu.sw	Conveyor Logic
heating	1.00.0	UserROM	0	FlowHeater.heating	\Cpu.sw	Heating PID
Cyclic #2 - [20 ms]						
Cyclic #3 - [50 ms]						
Cyclic #4 - [100 ms]						
visCtrl	1.00.0	UserROM	0	Visualisation.visCtrl	\Cpu.sw	Visualization Cor
visAlarm	1.00.0	UserROM	0	Visualisation.visAlarm	\Cpu.sw	Alarm Control
visTrend	1.00.0	UserROM	0	Visualisation.visTrend	\Cpu.sw	Trend Control
Cyclic #5 - [200 ms]						
Cyclic #6 - [500 ms]						
Cyclic #7 - [1000 ms]						
Cyclic #8 - [10 ms]						
Data Objects						
espres	1.00.0	UserROM	0	Receipes.espres	\Cpu.sw	Espresso
cappu	1.00.0	UserROM	0	Receipes.cappu	\Cpu.sw	Cappucino
regular	1.00.0	UserROM	0	Receipes.regular	\Cpu.sw	Normal coffee
Nc Data Objects						
Visualisation						

Assigning software elements to the software configuration

This allows the software elements needed for a specific system configuration to be assigned individually.

6 CONFIGURING THE HARDWARE

This chapter is based on the project created in Chapter 4.

This project already contains a configuration based on the „PC based Simulation Runtime“ and a Ladder Diagram program with two variables.

Open the project and insert an X20 configuration

Close the active project by selecting **<File> / <Close Project>** from the menu. The start page appears again, where the previously created project can be opened from a list of "Recent Projects".

Assign the **LampTest“**„ program to the task class Cyclic #1.

1) Create a new configuration

2) Assign the program to the software configuration

3) Insert I/O modules on the X2X Link interface

4) Assign variables to I/Os

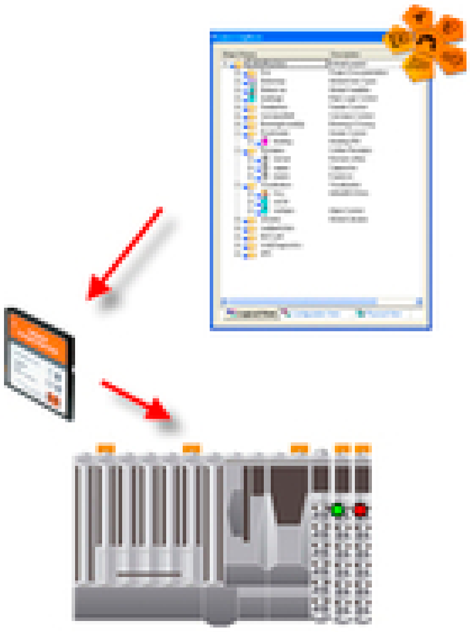
5) Configuring the Ethernet Interface

6) Compile the project

7) Create a CompactFlash card

Requirements:

- CompactFlash adapter
- Compact Flash with 512 MB
- SG4 CPU (e.g. X20CP1485)
- Digital input and output module (X20DI9371 and X20DO9322)
- Point-point connection between the PC and the X20 CPU with a crossover Ethernet cable

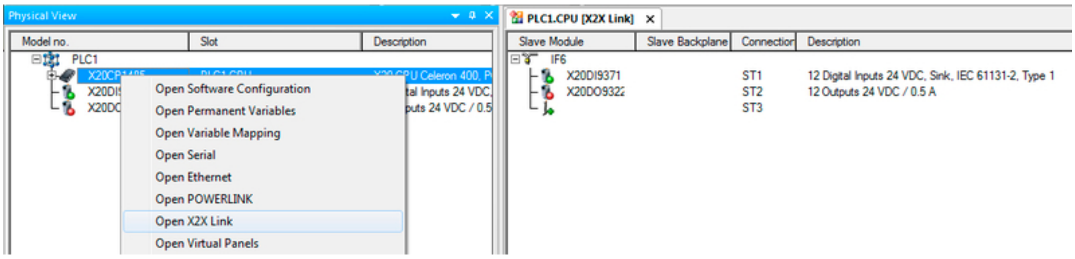


Setting up a Compact Flash card

6.1 Inserting I/O modules

In an X20 system, an I/O module is inserted on the X2X Link interface.

In Automation Studio, the interface editors can always be opened by selecting **<Open IF-Name>** from the menu or the shortcut menu



Slave Module	Slave Backplane	Connection	Description
X20DI9371		ST1	12 Digital Inputs 24 VDC, Sink, IEC 61131-2, Type 1
X20DO9322		ST2	12 Outputs 24 VDC / 0.5 A
		ST3	

Opening an interface in Automation Studio

Insert digital input and output module (X20DI9371 and X20DO9322)

Using the Getting Started chapter of the Automation Studio Help system as a guide, insert the two X20 modules in the Physical View.

1) Help chapter: **Automation Software / Getting Started / Creating programs with Automation Studio / Assigning variables to I/Os**

2) Open the X2X Link editor

3) Insert the modules in the X2X Link editor

6.1.1 X2X Link

I/O modules (e.g. X20) are connected with a uniform X2X Link backplane.

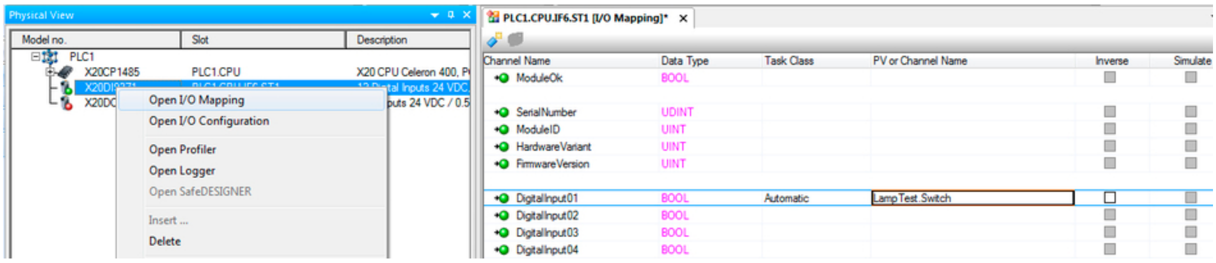
X2X Link guarantees the highest protection against disturbances as well as worldwide application without special tools thanks to its twisted copper cables.

? Communication / X2X

6.2 Assign variables to I/Os

I/O mapping refers to the assignment of controller program variables to I/O channels on the module.

Variables are assigned to an I/O channel in the I/O Mapping editor, which can be opened by selecting - **<Open I/O Mapping>>** in the shortcut menu of the corresponding I/O module, or by selecting **<Open> / <I/O Mapping>** from the main menu.



Model no.	Slot	Description
PLC1		
X20CP1485	PLC1.CPU	X20 CPU Celeron 400, P
X20DI9371		12 Digital Inputs 24 VDC,
X20DC		puts 24 VDC / 0.5

Channel Name	Data Type	Task Class	PV or Channel Name	Inverse	Simulate
ModuleOk	BOOL			□	□
SerialNumber	UDINT			□	□
ModuleID	UINT			□	□
HardwareVariant	UINT			□	□
FirmwareVersion	UINT			□	□
DigitalInput01	BOOL	Automatic	LampTest.Switch	□	□
DigitalInput02	BOOL			□	□
DigitalInput03	BOOL			□	□
DigitalInput04	BOOL			□	□

Opening the I/O mapping of a module

Assign variables to I/Os

Using the Getting Started chapter of the Automation Studio Help system as a guide, assign the two variables to a digital input channel and a digital output channel.

1) **Automation Software / Getting Started / Creating programswith Automation Studio / Assigning variables to I/Os**

2) Open I/O mapping

3) Assign the variable "Switch" to a digital input channel

4) Assign the variable "Lamp" to a digital output channel

?

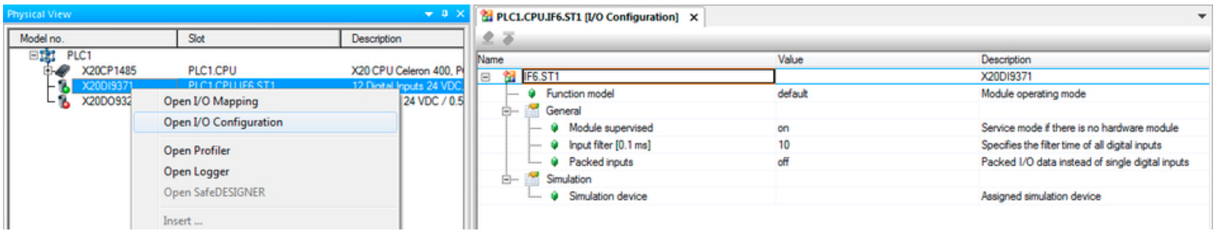
Programming / I/O configuration / I/O mapping

6.2.1 IO configuration

The ever-increasing functionality of remote B&R I/O modules results in more and more possibilities and operating modes in which these modules can be used.

The I/O configuration allows you to configure I/O modules without having to do any programming.

The I/O channels are configured in the I/O configuration editor, which can be opened by selecting - **<Open I/O Configuration>** in the shortcut menu of the corresponding I/O module, or by selecting **<Open>> / <I/O Configuration>** from the main menu.



Model no.	Slot	Description
PLC1		
X20CP1485	PLC1.CPU	X20 CPU Celeron 400, P
X20DI9371	PLC1.CPU.IF6.ST1	12 Digital Inputs 24 VDC
X20DO932		24 VDC / 0.5

Name	Value	Description
IF6.ST1		X20DI9371
Function model	default	Module operating mode
General		
Module supervised	on	Service mode if there is no hardware module
Input filter [0.1 ms]	10	Specifies the filter time of all digital inputs
Packed inputs	off	Packed I/O data instead of single digital inputs
Simulation		
Simulation device		Assigned simulation device

Opening the I/O configuration of a module

Automation Studio Help: Programming / I/O configuration

6.3 Setting up a network connection

Automation Studio requires a network connection in order to communicate with the controller.

The following requirements are necessary to make this possible:

- PC and controller are in the same network
- The permitted and assigned IP addresses are known

If the PC is already operated in a network, make a note of the settings before changing them so you can change them back later.

Please contact your system administrator for more detailed information about changing the network settings.

Using the Automation Studio Help system as a guide, set up a network connection between Automation Studio and the X20 CPU.

The chapter "Getting Started / Creating programs in Automation Studio / First project with X20 CPU" shows you how to establish the connection on the following pages:

- Ethernet settings on the target system
- Ethernet settings on the PC
- Configure online connection

Programming / Build & Transfer / Establish connection with target system Communication / Ethernet / AR configuration / Interface configuration

6.4 Build and transfer the project

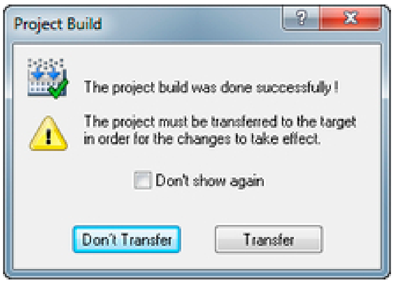
Once the program is complete, it can be built and transferred to the target system.

The target system is started up the first time when creating the CompactFlash card in Automation Studio, since this process involves partitioning the card, transferring the correct version of the operating system and transferring the project with the necessary system settings.

6.4.1 Build and Rebuild

A build regenerates all the changes that have taken place since the previous build, and is started using the build icon on the toolbar or by pressing the F7 key. A successful build is confirmed in the output window as follows Build: 0 error(s), 0 warning(s).

A successful build is confirmed in the output window as follows **Build**: **0 error(s), 0 warning(s)**.



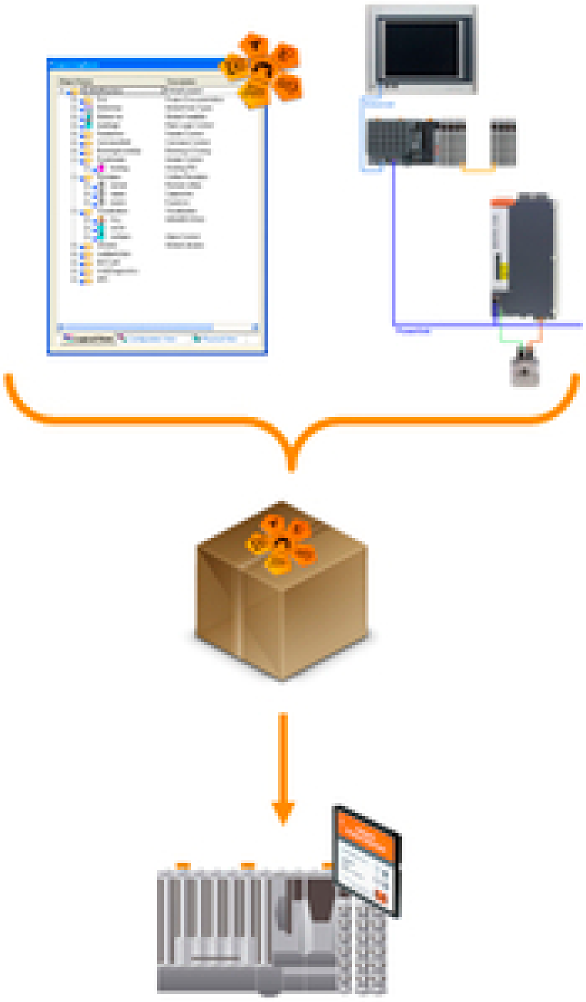
Project Build

The project build was done successfully !

The project must be transferred to the target in order for the changes to take effect.

□ Don't show again

Building the project

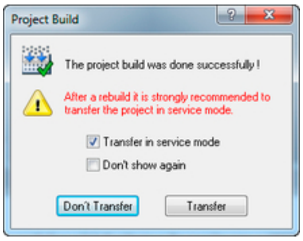


Build and transfer

The project can only be transferred to the target system if the target system has a workable configuration (Ethernet settings on target system) on the CompactFlash card.

During a rebuild, all software objects in the active configuration are regenerated, regardless of whether their source data has changed since the last build.

After a rebuild, the project should not be transferred with the CPU in RUN mode if a global variable is used in more than one task! It is recommended to change the CPU to SERVICE mode before transferring the project.



Project Build

The project build was done successfully !

After a rebuild it is strongly recommended to transfer the project in service mode.

☑ Transfer in service mode

□ Don't show again

Rebuilding the project

Programming / Build & Transfer / Build

6.4.2 Create a CompactFlash card

A CompactFlash card is created using the Automation Studio „Runtime Utility Center“. This can be opened by selecting **<Tools> / <Generate Compact Flash>** from the menu.

You will need a CompactFlash card and a CompactFlash adapter. During creation of the Compact Flash the Automation Studio Project will be copied to the Compact Flash. This can be controlled by a setting in the Configuration View

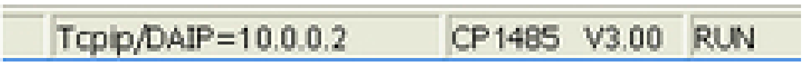
Create a CompactFlash card

Using the Automation Studio Help system as a guide, create a CompactFlash card.

The chapter "Getting Started / Creating programs in Automation Studio / First project with X20 CPU" shows you how to establish the connection on the following pages:

- Create a CompactFlash card

After the CompactFlash card is inserted in the X20 CPU, the online connection between Automation Studio and the target system is established automatically as long as the connection is configured correctly (see 6.3 "Setting up a network connection"). Once there is an Ethernet connection between the PC and the X20 CPU, the status of the online connection is displayed on the status bar.



Tcpip/DAIP=10.0.0.2 CP1485 V3.00 RUN

Run mode

The project can now be edited or diagnosed and then transferred via the online connection.

7 PROGRAMMING

7.1 Programs

A program is a program organization unit (POU) as defined in IEC 61131. It possesses the ability to directly access all global variables, functions, and function blocks.

7.2 Programming languages

Programs can be created in several different programming languages in Automation Studio. It is possible to combine several programming languages within one project.

The following programming languages are available

Programming language	IEC61131	Comment
Ladder Diagram (LD)	Yes	Graphical
Function Block Diagram (FBD)	Yes	Graphical
Continuous Function Chart (CFC)	No	Graphical
Sequential Function Chart (SFC)	Yes	Graphical & textual
Instruction List (IL)	Yes	Textual
Structured Text (ST)	Yes	Textual
ANSI C (C)	No	Textual

Table: Programming language overview

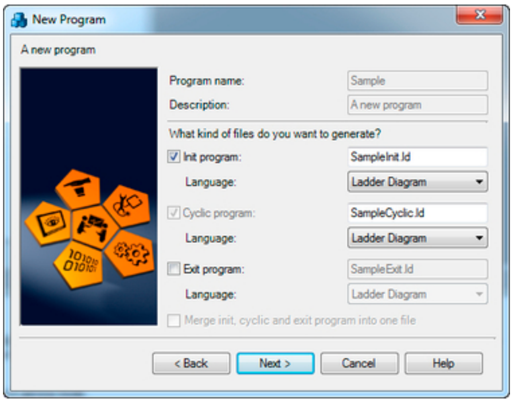
In Automation Studio, all textual programming languages use the same editor. Diagnostic tools are therefore always the same and are operated in the same way. This uniformity makes simplifies workflow and increases productivity.

Function blocks from B&R standard libraries can be called and used in all programming languages.

Programming / Programs

7.3 Initialization and cyclic sections of a program

When creating a program, the user can decide what sections should make up the program.



A new program

Program name: Sample

Description: A new program

What kind of files do you want to generate?

☑ Init program: SampleInit.ld

Language: Ladder Diagram

☑ Cyclic program: SampleCyclic.ld

Language: Ladder Diagram

□ Exit program: SampleExit.ld

Language: Ladder Diagram

□ Merge init, cyclic and exit program into one file

Selecting the program sections when creating a program

Task initialization

When the cyclic system is started, every task runs through its initialization program. This initialization program can contain program code that defines variable values.

Cyclic section of a task

The cyclic section of the program starts after task initialization is complete. Variables that are assigned values there retain them until they receive new ones or the system is restarted.

Exit section of the program

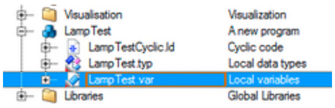
A task's exit program is called when uninstalling (deleting) the task. If resources (memory, interfaces) were used in the init or cyclic program, then these resources must be correctly released.

Additional information can be found in training module TM213.

7.4 Variables and data types

Variables are locations in memory for application-specific data ranges. In Automation Studio, variables are declared in a .var file.

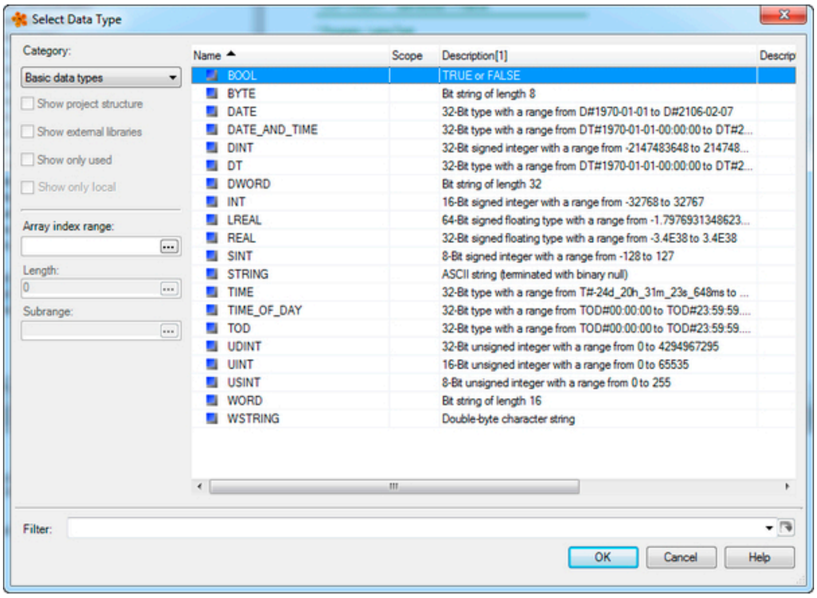
Programming is simplified by the use of symbolic variable names.



Visualisation	Visualization
LampTest	A new program
LampTestCyclic.ld	Cyclic code
LampTest.typ	Local data types
LampTest.var	Local variables
Libraries	Global Libraries

Variable declaration

Data types describe the properties of a variable, such as the range or the precision of the number contained in the variable, or which operations can be performed on it.



Select Data Type

Category: Basic data types

□ Show project structure
□ Show external libraries
□ Show only used
□ Show only local

Array index range:

Length: 0

Subrange:

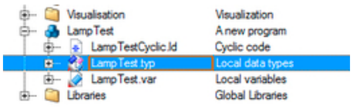
Name	Scope	Description[1]	Descrip
BOOL		TRUE or FALSE	
BYTE		Bit string of length 8	
DATE		32-Bit type with a range from D#1970-01-01 to D#2106-02-07	
DATE_AND_TIME		32-Bit type with a range from DT#1970-01-01-00:00:00 to DT#2...	
DINT		32-Bit signed integer with a range from -2147483648 to 214748...	
DT		32-Bit type with a range from DT#1970-01-01-00:00:00 to DT#2...	
DWORD		Bit string of length 32	
INT		16-Bit signed integer with a range from -32768 to 32767	
LREAL		64-Bit signed floating type with a range from -1.7976931348623...	
REAL		32-Bit signed floating type with a range from -3.4E38 to 3.4E38	
SINT		8-Bit signed integer with a range from -128 to 127	
STRING		ASCII string (terminated with binary null)	
TIME		32-Bit type with a range from T#-24d_20h_31m_23s_648ms to ...	
TIME_OF_DAY		32-Bit type with a range from TOD#00:00:00 to TOD#23:59:59....	
TOD		32-Bit type with a range from TOD#00:00:00 to TOD#23:59:59....	
UDINT		32-Bit unsigned integer with a range from 0 to 4294967295	
UINT		16-Bit unsigned integer with a range from 0 to 65535	
USINT		8-Bit unsigned integer with a range from 0 to 255	
WORD		Bit string of length 16	
WSTRING		Double-byte character string	

Filter:

Data type of a variable

Programming / Variables and datatypes

User-defined data types can be created by the user based on the simple data types. In Automation Studio, user-defined data types, also known as derived data types, are declared in a .typ file.



Visualisation	Visualization
LampTest	A new program
LampTestCyclic.ld	Cyclic code
LampTest.typ	Local data types
LampTest.var	Local variables
Libraries	Global Libraries

Declaration of data types

Programming / Variables and datatypes / Data types / Derived data types

7.4.1 Variable scope

A project's packages can be arranged in a nested structure in the Logical View. This allows the data and functionality to be organized modularly.

This structure determines the scope and visibility of the declared variables and data types. A variable can therefore be defined "logically" at the appropriate location in the project.

In Automation Studio, the visibility of the variables is determined by the position of the .var file:

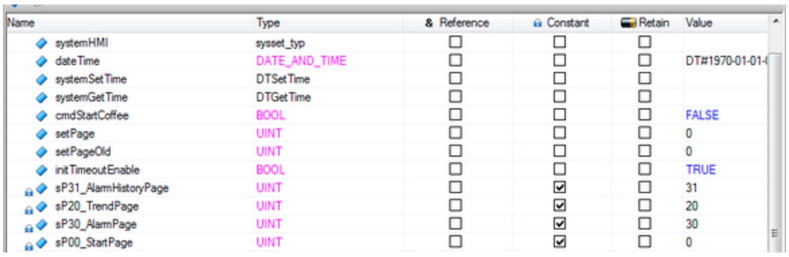
Local variables are defined with a local scope for a program, and can't be used in the project's other POUs.

Global variables at the highest level are visible in the entire project. They can therefore be used in any program at any level of a package.

Package-local variables are declared within a package and are valid in the respective package and all subordinate packages and programs. However, the validity of these variables is global from the standpoint of the controller.

7.4.2 Initializing and buffering variable values

A data type and other properties can be assigned to a variable in the variable declaration.



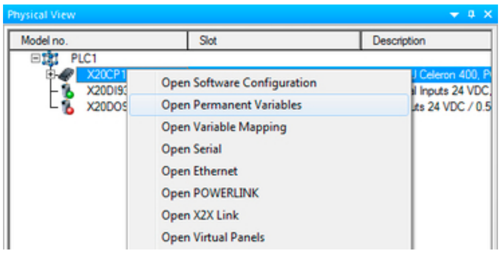
Name	Type	Reference	Constant	Retain	Value
systemHMI	sysset_typ	□	□	□	
dateTime	DATE_AND_TIME	□	□	□	DT#1970-01-01-
systemSetTime	DTSetTime	□	□	□	
systemGetTime	DTGetTime	□	□	□	
cmdStartCoffee	BOOL	□	□	□	FALSE
setPage	UINT	□	□	□	0
setPageOld	UINT	□	□	□	0
initTimeoutEnable	BOOL	□	□	□	TRUE
sP31_AlarmHistoryPage	UINT	□	☑	□	31
sP20_TrendPage	UINT	□	☑	□	20
sP30_AlarmPage	UINT	□	☑	□	30
sP00_StartPage	UINT	□	☑	□	0

Example of a variable declaration

Constants are variables whose values must not change while a program is executed. A constant is assigned its initial value when the software is created.

Remanent and **permanent** variables (Retain) are **saved** in buffered memory (SRAM) so that they can be reloaded after the system is restarted. Unlike remanent variables, permanent variables are also protected against a cold restart.

In both cases, the buffering (battery, rechargeable battery) in the CPU or backplane is responsible for holding on to the data.



Inserting permanent variables

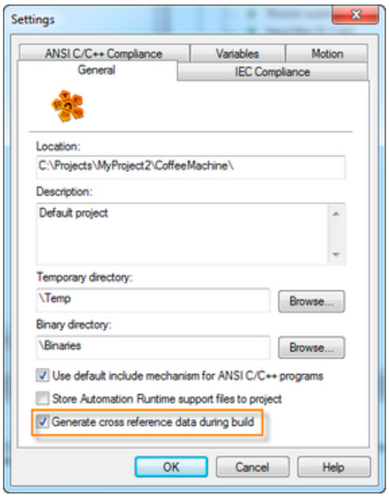
For variables to be stored in the permanent area, they have to be defined as **Retain** and **Global** in the variable declaration.

7.5 The cross reference list

Common search tasks can be handled easily with the help of the cross reference list.

All variables used in a program can be viewed in the list. The list also provides information about **where and how** each variable is used in the program (read or write access).

In the general project settings you can activate the option to generate this cross-reference list automatically during a project build. This setting is made on the "General" tab under **<Project> / <Settings>**.



Location:
C:\Projects\MyProject2\CoffeeMachine\

Description:
Default project

Temporary directory:
\Temp

Binary directory:
\Binaries

☑ Use default include mechanism for ANSI C/C++ programs

□ Store Automation Runtime support files to project

☑ Generate cross reference data during build

Activate cross-reference list

After the project is compiled, the cross-reference list functions are available.

Project Management / The Workspace / General project settings

7.6 Teamwork

Working in a team requires responsibilities to be divided up between the various team members.

Automation Studio has functions designed specifically to help teams work more efficiently:

- Sharing project files with minimum sizes.
- Working with source control

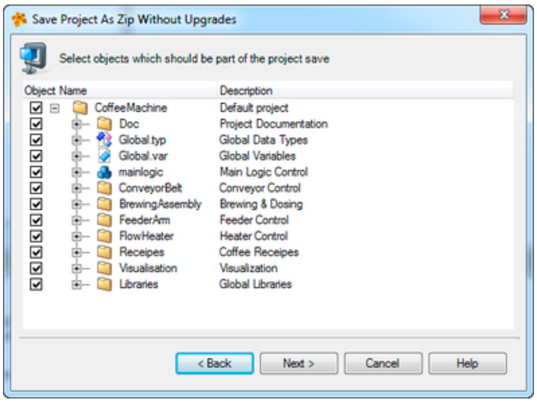
Project Management / Importing / Exporting projects

Project Management / Using source control systems

7.6.1 Exporting projects

An Automation Studio project can be shared with other programmers using the Export function.

Export a project by selecting **<File> / <Save Project As ZIP>>** oder das Menü **<File> / <Save Project As ZIP Without Upgrades>** while in the Logical View.



Select objects which should be part of the project save

Object Name	Description
☑ CoffeeMachine	Default project
☑ Doc	Project Documentation
☑ Global.typ	Global Data Types
☑ Global.var	Global Variables
☑ mainlogic	Main Logic Control
☑ ConveyorBelt	Conveyor Control
☑ BrewingAssembly	Brewing & Dosing
☑ FeederArm	Feeder Control
☑ FlowHeater	Heater Control
☑ Receipes	Coffee Receipes
☑ Visualisation	Visualization
☑ Libraries	Global Libraries

Exporting software from the Logical View

7.6.2 Exporting and importing software components

A completed software component can be exported and imported, allowing it to be shared or reused in other projects.

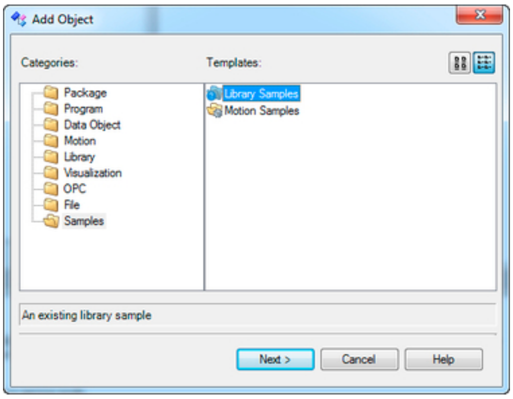
An export is performed on a package (directory) in the Logical View.

If this package contains references to standard libraries, these can be specified in the properties of the package. These properties are checked when importing, and any required libraries are added automatically.

Inserting a library sample

For some libraries, there are prepared samples that are available in the form of a package. These can be imported into an existing project if needed. This exercise is not based on any particular application. It is only intended to show how to import a completed software component.

From the Logical View, import the component by selecting **<Insert> / <Add Object>** from the menu.



An existing library sample

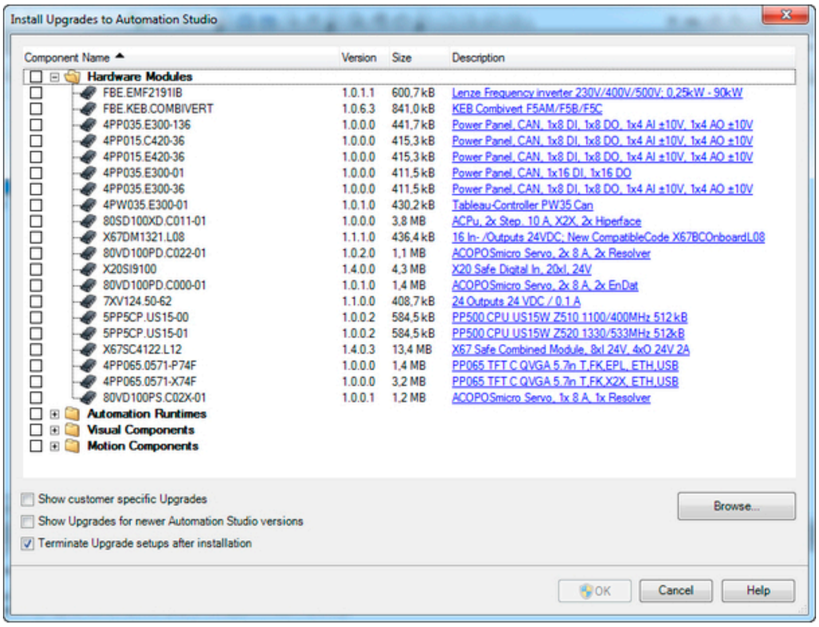
Importing a prepared library sample

7.7 Hardware and software upgrades

Upgradeable components make it possible to update hardware and motion libraries and to upgrade to new versions of Visual Components or Automation Runtime.

To upgrade components online, you must first start Automation Studio.

Then select **<Tools> / <Upgrades>** from the menu. The upgrades currently available on the B&R homepage are listed in a dialog box.



Component Name	Version	Size	Description
□ **Hardware Modules**			
□ FBE.EMF2191IB	1.0.1.1	600,7 kB	Lenze Frequency inverter 230V/400V/500V; 0,25kW - 90kW
□ FBE.KEB.COMBIVERT	1.0.6.3	841,0 kB	KEB Combivert F5AM/F5B/F5C
□ 4PP035.E300-136	1.0.0.0	441,7 kB	Power Panel, CAN, 1x8 DI, 1x8 DO, 1x4 AI ±10V, 1x4 AO ±10V
□ 4PP015.C420-36	1.0.0.0	415,3 kB	Power Panel, CAN, 1x8 DI, 1x8 DO, 1x4 AI ±10V, 1x4 AO ±10V
□ 4PP015.E420-36	1.0.0.0	415,3 kB	Power Panel, CAN, 1x8 DI, 1x8 DO, 1x4 AI ±10V, 1x4 AO ±10V
□ 4PP035.E300-01	1.0.0.0	411,5 kB	Power Panel, CAN, 1x16 DI, 1x16 DO
□ 4PP035.E300-36	1.0.0.0	411,5 kB	Power Panel, CAN, 1x8 DI, 1x8 DO, 1x4 AI ±10V, 1x4 AO ±10V
□ 4PW035.E300-01	1.0.1.0	430,2 kB	Tableau-Controller PW35 Can
□ 80SD100XD.C011-01	1.0.0.0	3,8 MB	ACPu, 2x Step, 10 A, X2X, 2x Hiperface
□ X67DM1321.L08	1.1.1.0	436,4 kB	16 In- /Outputs 24VDC; New CompatibleCode X67BCOnboardL08
□ 80VD100PD.C022-01	1.0.2.0	1,1 MB	ACOPOSmicro Servo, 2x 8 A, 2x Resolver
□ X20SI9100	1.4.0.0	4,3 MB	X20 Safe Digital In, 20xI, 24V
□ 80VD100PD.C000-01	1.0.1.0	1,4 MB	ACOPOSmicro Servo, 2x 8 A, 2x EnDat
□ 7XV124.50-62	1.1.0.0	408,7 kB	24 Outputs 24 VDC / 0.1 A
□ 5PP5CP.US15-00	1.0.0.2	584,5 kB	PP500 CPU US15W Z510 1100/400MHz 512 kB
□ 5PP5CP.US15-01	1.0.0.2	584,5 kB	PP500 CPU US15W Z520 1330/533MHz 512kB
□ X67SC4122.L12	1.4.0.3	13,4 MB	X67 Safe Combined Module, 8xI 24V, 4xO 24V 2A
□ 4PP065.0571-P74F	1.0.0.0	1,4 MB	PP065 TFT C QVGA 5.7in T,FK,EPL, ETH,USB
□ 4PP065.0571-X74F	1.0.0.0	3,2 MB	PP065 TFT C QVGA 5.7in T,FK,X2X, ETH,USB
□ 80VD100PS.C02X-01	1.0.0.1	1,2 MB	ACOPOSmicro Servo, 1x 8 A, 1x Resolver
□ **Automation Runtimes**			
□ **Visual Components**			
□ **Motion Components**			

□ Show customer specific Upgrades

□ Show Upgrades for newer Automation Studio versions

☑ Terminate Upgrade setups after installation

Upgradeable components

Project Management / Automation Studio upgrade

8 SUMMARY

Automation Studio is more than just a programming tool. It supports the user throughout the entire life cycle of a machine - from the initial concept to the finished project, from first commissioning to series production.

The ability to clearly structure the software based on machine parts and to work with different configurations makes it possible to manage multiple machine variations in one project and allows a whole team to work on the same project.

Automation Studio

Automation Studio is the reliable companion of the programmer, the service and maintenance technician, and offers just the right tool for any stage in a machine's life cycle.

TRAINING MODULES

TM210 – Working with Automation Studio
TM213 – Automation Runtime
TM220 – The Service Technician on the Job
TM223 – Automation Studio Diagnostics
TM230 – Structured Software Development
TM240 – Ladder Diagram (LD)
TM241 – Function Block Diagram (FBD)
TM242 – Sequential Function Chart (SFC)
TM246 – Structured Text (ST)
TM250 – Memory Management and Data Storage
TM261 – Closed Loop Control with LOOPCONR
TM400 – Introduction to Motion Control
TM410 – Working with Integrated Motion Control
TM440 – Motion Control: Basic Functions
TM441 – Motion Control: Multi-axis Functions
TM450 – ACOPOS Control Concept and Adjustment
TM460 – Initial Commissioning of Motors
TM480 – The Basics of Hydraulics
TM481 – Valve-based Hydraulic Drives
TM482 – Hydraulic Servo Pump Drives
TM500 – Introduction to Integrated Safety
TM510 – Working with SafeDESIGNER
TM530 – Developing Safety Applications
TM540 – Integrated Safe Motion Control
TM600 – Introduction to Visualization
TM610 – Working with Integrated Visualization
TM630 – Visualization Programming Guide
TM640 – Alarms, Trends and Diagnostics
TM670 – Advanced Visual Components
TM810 – APROL Setup, Configuration and Recovery
TM811 – APROL Runtime System
TM812 – APROL Operator Management
TM813 – APROL XML Queries and Audit Trail
TM830 – APROL Project Engineering
TM890 – The Basics of LINUX

TM210TRE.30-ENG / V1.2.0.15

www.br-automation.com