

**Welcome to B&R Automation Software**



# Contents (8.30.2005)

<b>1 Welcome to B&amp;R Automation Software™</b> .....	<b>1</b>
1.1 Jumping into Automation Software™.....	1
1.1.1 General information about B&R Automation Software™.....	1
1.1.1.1 What is B&R Automation Software™?.....	1
1.1.1.2 Three terms, one philosophy.....	1
1.1.1.3 The complete solution.....	2
1.1.2 Automation Software™ CD.....	2
1.1.2.1 Requirements.....	2
1.1.3 Ordering the B&R Automation Software™ CD.....	2
1.1.3.1 Evaluation version.....	3
1.1.3.2 Full version.....	3
1.1.3.3 Upgrade version.....	4
1.1.4 B&R Automation Studio™.....	4
1.1.4.1 Project creation and configuration.....	4
1.1.4.2 Programming languages.....	5
1.1.4.3 Library manager.....	9
1.1.4.4 Import / Export.....	11
1.1.4.5 Diagnostic tools.....	11
1.1.4.6 Service.....	12
1.1.4.7 Motion Components.....	13
1.1.4.8 Visual Components.....	14
1.1.4.9 TPU code linker.....	17
1.1.5 B&R Automation Net™.....	18
1.1.5.1 Transparent communication with B&R Automation Net™.....	18
1.1.5.2 Accessing B&R Automation Net™.....	18
1.1.5.3 Process Visualization Interface (PVI).....	19
1.1.5.4 PVI interface types.....	21
1.1.6 B&R Automation Runtime™.....	22
1.1.6.1 What is B&R Automation Runtime™?.....	22
1.1.6.2 One runtime system for all B&R automation targets.....	22
1.1.6.3 B&R automation targets in the industrial PC family.....	22
1.1.6.4 Installation kit.....	23
1.1.6.5 Ordering licenses.....	23
1.1.6.6 Runtime system ordering information.....	24
1.1.6.7 Automation Runtime™ AR010.....	24
1.1.6.8 Automation Runtime™ AR010/Desktop.....	25
1.1.6.9 Automation Runtime™ AR102.....	25
1.1.6.10 Automation Runtime™ AR105.....	26
1.1.7 Accessories.....	26
1.1.7.1 Overview.....	26
1.2 Getting started in B&R Automation Studio.....	26
1.2.1 Introduction.....	26
1.2.1.1 The most important tools.....	26
1.2.1.2 Conventions.....	29
1.2.1.3 Features.....	30
1.2.2 Software installation.....	31
1.2.2.1 Automation Net system requirements.....	31
1.2.2.2 Automation Studio system requirements.....	32
1.2.2.3 Installing B&R Automation Net.....	33
1.2.2.4 Installing Microsoft Internet Explorer.....	34
1.2.2.5 Installing B&R Automation Software.....	35
1.2.2.6 Entering user data.....	36
1.2.2.7 Setting the program folder.....	36
1.2.2.8 Selecting software components.....	37
1.2.2.9 Adding or installing software components.....	39
1.2.2.10 Licensing.....	40
1.2.2.11 Uninstalling B&R Automation Software.....	42
1.2.2.12 Installing using a response file ("silent installation").....	44
1.2.2.13 AR installation disks.....	46
1.2.3 Getting started.....	46

# Contents (8.30.2005)

## 1 Welcome to B&R Automation Software™

1.2.3.1	Starting B&R Automation Studio.....	46
1.2.3.2	Example program.....	48
1.2.3.3	How do I create a project?.....	49
1.2.3.4	Creating a cyclic object.....	57
1.2.3.5	Building the object and transferring it to the controller.....	68
1.2.3.6	Testing the object.....	70
1.2.4	Frequently asked questions.....	80
1.2.4.1	How do I install the operating system?.....	80
1.2.4.2	How can I work without hardware?.....	83
1.2.4.3	Kann ich mehrere B&R Automation Studio Versionen installieren?.....	85
1.3	Shortcut keys.....	87
1.3.1	General.....	87
1.3.2	Desktop.....	89
1.3.3	Text editors.....	89
1.3.4	Build & transfer.....	89
1.3.5	Debugger.....	89
1.3.6	LAD editor.....	90

## 2 How do I use the help system?.....91

2.1	Online help window – General.....	91
2.2	You can find information in the online help system in one of the following ways:.....	91

# 1 Welcome to B&R Automation Software™

Welcome to the Bernecker & Rainer Online Help System

.....We gladly offer our support as you work with our products.

.....And we will assist in answering difficult questions.

We wish you lots of fun and success working with our products.

Your B&R Team

## 1.1 Jumping into Automation Software™

To provide you with a better overview of B&R Automation Software, this chapter will present the fundamental components of Automation Software. A brief description of each individual component should serve to give you an overview of its use and advantages. You will also obtain information about how to order and license an Automation Software CD.

### 1.1.1 General information about B&R Automation Software™

This section should illustrate the philosophy represented by B&R Automation Software™.

#### 1.1.1.1 What is B&R Automation Software™?

B&R Automation Software™ combines all the software packages necessary for [configuring](#), [programming](#), [diagnosing](#), and operating all B&R Control Systems™, B&R Motion Systems™ und B&R Panel Systems™ components.

Complex automation tasks require the right tools. An increasing number of automation projects are being created in a shorter amount of time. In these projects, all components must be seamlessly compatible.

Automation projects often begin with a very simple control task and later develop into complex positioning and visualization applications.

It is a distinct advantage for the user if i.e. changing to a B&R control family with higher performance requires little or no changes to program source codes. It should also not be necessary to leave the B&R development environment when switching between application areas, e.g. from positioning to visualization.

For all of these reasons, B&R offers the perfect programming and diagnostics tools for every phase of an automation project's development. The user himself determines which components he wants to have when installing the development system. New software does not have to be ordered when the requirements change. The user only has to select the required components from the installation CD.

#### 1.1.1.2 Three terms, one philosophy

The essential core areas when creating automation applications are described at B&R with the following terms:

B&R term	Core area
<a href="#">B&amp;R Automation Studio™</a>	Program creation
<a href="#">B&amp;R Automation Net™</a>	Uploading/Downloading programs and data to/from the controller
<a href="#">B&amp;R Automation Runtime™</a>	Executing the controller program

At B&R, these terms are unified as a single philosophy, "B&R Automation Software™". It includes all components required to create, transfer, and execute automation programs and data.

### 1.1.1.3 The complete solution

The B&R concept is based on complete solutions. These solutions expand the automation world well beyond the traditional hardware module strategy.

The core directive when developing all B&R products is always compatibility this refers to hardware and, with B&R Automation Software™, now also refers to communication, programming, and application software.

### 1.1.2 Automation Software™ CD

The B&R Automation Software™ CD combines all the tools required for configuring and programming B&R Control Systems™, B&R Motion Systems™, and B&R Panel Systems™ components in a single development environment.

The following development tools can be installed from the B&R Automation Software™ CD:

Development tool	Description
B&R Automation Studio™	For configuration, programming, visualization, and diagnostics of controllers and drives.
B&R Automation Net™ / PVI	For connecting Windows™ programs and graphics to our controller families.
B&R Automation Runtime™ Installation Kit	Used to create installation diskettes for B&R Automation Runtime™ systems.
B&R Motion Components™	For creating and configuring positioning, gear, and cam profile applications.
B&R Visual Components™	For configuring PANELWARE™ visualization devices.
Automation Studio Version Changer	For switching between different software base versions when more than one are installed.
Windows CE Installation Sets	Tool for creating Windows CE Installation diskettes.

In addition, the CD contains all the libraries necessary for programming and all drivers necessary for operating network devices. Utilities such as Internet Explorer, database drivers, as well as programming and training documentation are also included.

#### 1.1.2.1 Requirements

The following hardware is required before installing Automation Software components:

- MS Windows™ XP
- 400 MHz Pentium II processor
- RAM: At least 256 MB
- Hard drive: At least 500 MB
- XGA 1024 x 768

### 1.1.3 Ordering the B&R Automation Software™ CD

Uniting all of the software packages required to configure and program all B&R Control Systems™, B&R Motion Systems™, and B&R Panel Systems™ components into a single development environment is also consistent with how everything is ordered. Therefore, depending on the application, only a few model numbers are needed to order the B&R Automation Software™ CD.

B&R considers service and maintenance of their software components to be very important and is constantly striving to expand software capabilities and solve possible problems to fulfill the requirements of automation technology. To make these changes accessible not only to new customers, but also to our existing customers, it is necessary for customers to register with B&R by licensing their software.

Licensing also is the basis of our upgrade service, which automatically supplies the customer with new versions of the B&R Automation Software™ CD.

**Note:**

This chapter only applies to licensing the development tools on the B&R Automation Software™ CD. The software used on serial systems during runtime is subject to other licensing regulations. An explanation is provided in the "B&R Automation Runtime™" chapter.

The development tools on the B&R Automation Software™ CD are designed for the following uses:

- Evaluation version: For customer evaluation or use at exhibitions and customer presentations
- Full version: For use in software development
- Upgrade version: For B&R customers

**1.1.3.1 Evaluation version**

**1.1.3.1.1 Order data**

Order data	Short description
1A4000.01	B&R Automation Software™ CD Evaluation Version, development system for MS Windows 95/98/ME/2000/NT4.0/XP, German and English version, can be used for 30 days

**1.1.3.1.2 General information**


The evaluation version of the B&R Automation Software™ CD can be used for programming evaluation, presentations, exhibitions, and similar applications.

The evaluation version does not require [licensing](#), but it can only be used for 30 days. This means that the evaluation version works like the full version after first being installed on a computer, and all development system components can be used without limitations for 30 days. After 30 days, the components no longer function and a corresponding message is displayed.

If an evaluation CD with a higher version number is installed, the runtime limit is reset and the program can be evaluated for another 30 days.

**1.1.3.2 Full version**

**1.1.3.2.1 Order data**

Order data	Short description	Short description
1A4000:L1	B&R Automation Software™ Single Set, consists of: 1 B&R Automation Software™ CD for MS Windows XP, German and English version, 1 workstation license.	
1A4000:L5	B&R Automation Software™ Multiple Set, consists of: 5 B&R Automation Software™ CDs for MS Windows XP, German and English version, 5 workstation licenses.	

1A4000:LU	B&R Automation Software™ Unlimited Set, consists of: 20 B&R Automation Software™ CDs for MS Windows XP, German and English version, company license
-----------	---

### 1.1.3.3 Upgrade version

#### 1.1.3.3.1 Order data

Order data	Short description
1A4000:U1	B&R Automation Software™ Single Upgrade Set consists of: 1x Upgrade Service for 12 months
1A4000:U5	B&R Automation Software™ Multiple Upgrade Set consists of: 5x Upgrade Service for 12 months
1A4000:UU	B&R Automation Software™ Unlimited Upgrade Set consists of: 20x Upgrade Service for 12 months

#### 1.1.3.3.2 General information

With the purchase of an upgrade version, an upgrade contract is automatically made between B&R and the customer, and the B&R Upgrade Service is activated. From that point on, new versions of the B&R Automation Software CD™ (model no. 1A4000.01) are sent to the customer's delivery address for the next 12 months.

The customer is reminded to extend the contract about 3 to 4 weeks before it expires. If another Upgrade Version is not ordered after the 12 months have passed, the upgrade contract is cancelled and B&R no longer automatically sends upgrades.

### 1.1.4 B&R Automation Studio™

The core package for programming automation applications is B&R Automation Studio™. It allows B&R automation devices to be easily and efficiently configured and programmed using a graphic user interface that runs in MS Windows™.

Clearly arranged structures allow simple operation and a brief "settling-in" period. Comprehensive online help is available if you have any problems. The entire package is available in English and German. During installation, the user decides which language should be used.

#### 1.1.4.1 Project creation and configuration

##### 1.1.4.1.1 Project creation

An important characteristic of a development system is the ability to establish a reference between the project and the physical system at any time. B&R Automation Studio™ offers this function in the project view. Here, the relationship between the hardware and software configuration is shown graphically.

##### 1.1.4.1.2 Hardware configuration

The hardware configuration on the left side of the main window shows the target system graphically using a tree structure. This "hardware tree" can be created by the user without a connection to the PLC (offline) or automatically with an active connection to the PLC (online). The modules are lined up according to their slots and shown according to their type (CPU, digital input, digital output, etc.).



Detailed information about the target system, from installation dimensions to terminal blocks, can be viewed in the integrated hardware documentation.

#### 1.1.4.1.3 Software configuration

The software configuration on the right side of the main window graphically displays the connection to the hardware.

The most important relationship between software and hardware is the direct assignment of symbolic program variables to physical terminals. The terminal is no longer accessed in the program with the terminal ID. Instead, it is accessed using this symbolic process variable name. Therefore, changing the "wiring" does not require program modifications.

The software project consists of one or more program objects (tasks) which are placed in various task classes. Task classes are groups of programs which are processed with the same cycle time (e.g. 10 ms). In addition to program objects, data, system, and advanced objects for special functions are also included. All these objects are created using dialog boxes (Wizards).

Additional properties for these objects can be efficiently configured using corresponding property dialog boxes. Downloading the operating system also takes place using menu items in the software configuration.

#### 1.1.4.2 Programming languages

##### 1.1.4.2.1 General information

The following programming languages can be used in B&R Automation Studio™:

B&R Automation Basic™

ANSI C

IEC1131 programming languages:

IEC61131 Ladder Diagram (LD)

IEC61131 Instruction List (IL)

IEC61131 Structured Text (ST)

IEC61131 Sequential Function Chart (SFC)

Syntax coloring (program keywords are colored) eases the creation and analysis of source code for all programming languages. Declaring variables, constants, and data types (structures) takes place in context during programming or in dialog boxes.

Error messages are shown in the output window if programming errors occur. Double-clicking on these error messages opens the corresponding program editor and the faulty program lines are highlighted.

Program comments can be added between the lines, to the right of program commands, or in other suitable locations in the language itself to provide a detailed explanation of the program section. Documentation possibilities (printouts) are also available in all programming languages.

Different zoom settings or fonts can be used to compensate for low screen resolution or if more data should be displayed.

##### 1.1.4.2.2 B&R Automation Basic™

The screenshot shows the B&R Automation Studio interface. The main window displays a ladder logic program with the following code:

```

0011
0012 IF (do135mdef = 1) THEN
0013   IO_dataD14w.enable := 1
0014   IO_dataD14w.ioident := IO_mD14w.ioident
0015
0016 ("Ausführen des Schaufelauftrages")
0017
0018 (" FUNCTION_BLOCK IO_data
0019   VAR_INPUT
0020     enable:BOOL;
0021     ioident:UDINT;
0022   VAR_OUTPUT
0023     status:UINT;
0024   ")
0025
0026 IO_data_Temp FUB IO_data) ("FUB Aufruf mit Alias Name")
0027
0028
0029 IO_data_Temp.enable = 1 ("Eingangsparameter")
0030 IO_data_Temp.ioident = ioident
0031
0032 stat_data = IO_data_Temp.status ("Statuswert, muß wenn alles OK = 0 sein")
0033
0034
0035 IF (IO_dataD14w.status = 0) THEN ("Parameter für IO_data werden in Abhängigkeit vom...")
0036   do135init := 1 ("... Status beschrieben")
0037   do135mdef := 0
0038 ENDIF
0039
0040 ENDIF

```

An inset window titled "a\_tempo [Declaration]" shows a table of variables:

Name	Type	Scope	Attribute	Value
IO_d14w	BOOL	local	memory	*remanent
IO_dataD14w	BOOL	local	memory	*remanent
IO_data_Temp	BOOL	local	memory	*remanent
IO_mD14w	BOOL	global	memory	*remanent
Temp1	BOOL	local	memory	*remanent
Temp2	BOOL	local	memory	*remanent
Temp3	BOOL	local	memory	*remanent
Temp4	BOOL	local	memory	*remanent
do135mdef	BOOL	local	memory	*remanent
ioident	BOOL	local	memory	*remanent
stat_data	BOOL	local	memory	*remanent

The status bar at the bottom indicates "Line 3 of 12".

## Automation Basic (AB)

B&R Automation Basic™ is a text and instruction–based high–level language developed by B&R. In addition to standard programming language elements, it contains language elements which have been greatly improved and adapted for B&R controllers.

B&R Automation Basic™ was developed from the PL2000 programming language and is therefore suitable for updating existing projects that were created with PG2000 (predecessor to B&R Automation Studio™), as well as for creating new projects in all areas of automation.

### 1.1.4.2.3 ANSI C

The screenshot shows the B&R Automation Studio interface with the following ANSI C code:

```

UINT Auflösung = HUNDERTSTEL;
/*
|
|-----Auflösung: HUNDERTSTEL / ZEHNTTEL ----- */

do135init=0; /* Statusvariable,
wenn Initialisierung fertig -> do135init = 1 */

/* Kopieren der eingestellten Auflösung auf die zu schreibenden Daten */
do135cfg=Auflösung;
iop.master_adr=1; /* immer 1 */
iop.slave_adr=0; /* immer 0 */
iop.module_adr=0; /* Moduladresse des Analoginterfaces (bei CP Interface = 0) */
iop.io_type=5; /* 5...2003-Steuerung */
iop.mode = 0; /* Daten auf Modul schreiben */

/*
WICHTIG WICHTIG WICHTIG WICHTIG WICHTIG WICHTIG WICHTIG WICHTIG
Hier kann der Steckplatz des AT Modules verändert werden !
Zur Zeit ist Steckplatz 1 eingestellt !!!!!
(20..Steckplatz 1) (60..Steckplatz 2) (92..Steckplatz 3) (124..Steckplatz 4) */

iop.intern_off = 20; /*(Slot Nummer des ANP - 1)*32 + Register Nummer * 2
/* z.Bsp.:ANP auf Slot 2, Register 14 -> (2 - 1)*32+14*2=60*/

iop.data_len = 2;

iop.data_adr = adr(do135cfg);

IO_mD14w.enable=1;
IO_mD14w.iop_adr=4iop;
IO_mD14w.ac_adr=((UINT*)4accr);
IO_mD14w.io_index=0;

/* Anlegen eines Schaufelauftrages */
IO_mphydef (4IO_mD14w);

IF (IO_mD14w.status == 0) /* wenn alles funktioniert hat */

```

The status bar at the bottom indicates "Ln 2, Col 1".

## ANSI C

ANSI C was developed in the mid–70s from the language B (Basic Combined Programming Language) and is now a standardized, machine–oriented high–level language which has become very well known throughout the world. Even though it is not standard for all PLC manufacturers, B&R has fully integrated this language into B&R

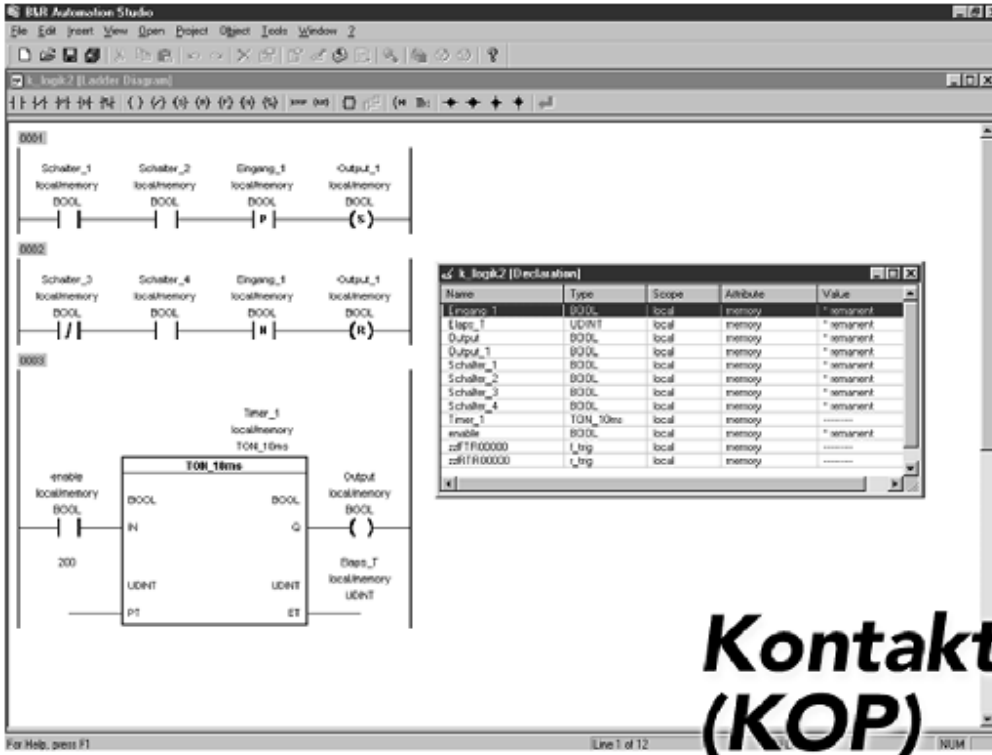
Automation Studio™.

The ANSI C high-level language is mostly used to create problem-oriented program sections where calculations are made, algorithms are created, or existing C code is used.

1.1.4.2.4 IEC1131 programming languages

The IEC1131 standard summarizes the requirements for PLC hardware and their programming systems. The goal is reducing costs for training as well as creating and implementing programs using uniform methods. The IEC 1131-3 standard is a guideline for PLC programming and was developed by an international standardization group (International Electrotechnical Commission).

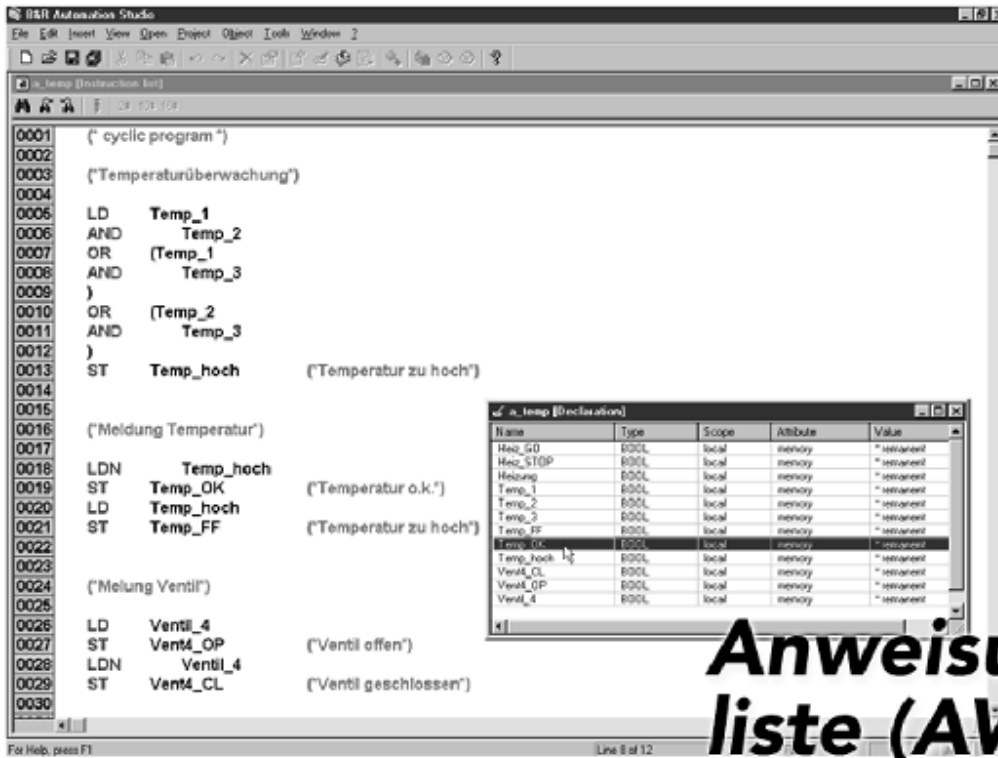
1.1.4.2.4.1 Ladder Diagram (LD)



**Kontaktplan  
(KOP)**

LD is a programming method that uses diagrams. It is derived from electromechanical relay systems and defines the current flow through individual networks. For this reason, ladder diagrams look similar to electrical circuit diagrams and are mostly limited to processing Boolean signals (1=TRUE, 0=FALSE).

1.1.4.2.4.2 Instruction List (IL)



The screenshot shows the B&R Automation Studio interface with an AWL program. The main window displays the following instructions:

```

0001 (" cyclic program ")
0002
0003 ("Temperaturüberwachung")
0004
0005 LD Temp_1
0006 AND Temp_2
0007 OR (Temp_1
0008 AND Temp_3
0009 )
0010 OR (Temp_2
0011 AND Temp_3
0012 )
0013 ST Temp_hoch ("Temperatur zu hoch")
0014
0015
0016 ("Meldung Temperatur")
0017
0018 LDN Temp_hoch
0019 ST Temp_OK ("Temperatur o.k.")
0020 LD Temp_hoch
0021 ST Temp_FF ("Temperatur zu hoch")
0022
0023
0024 ("Melung Ventil")
0025
0026 LD Ventil_4
0027 ST Ventil_OP ("Ventil offen")
0028 LDN Ventil_4
0029 ST Ventil_CL ("Ventil geschlossen")
0030

```

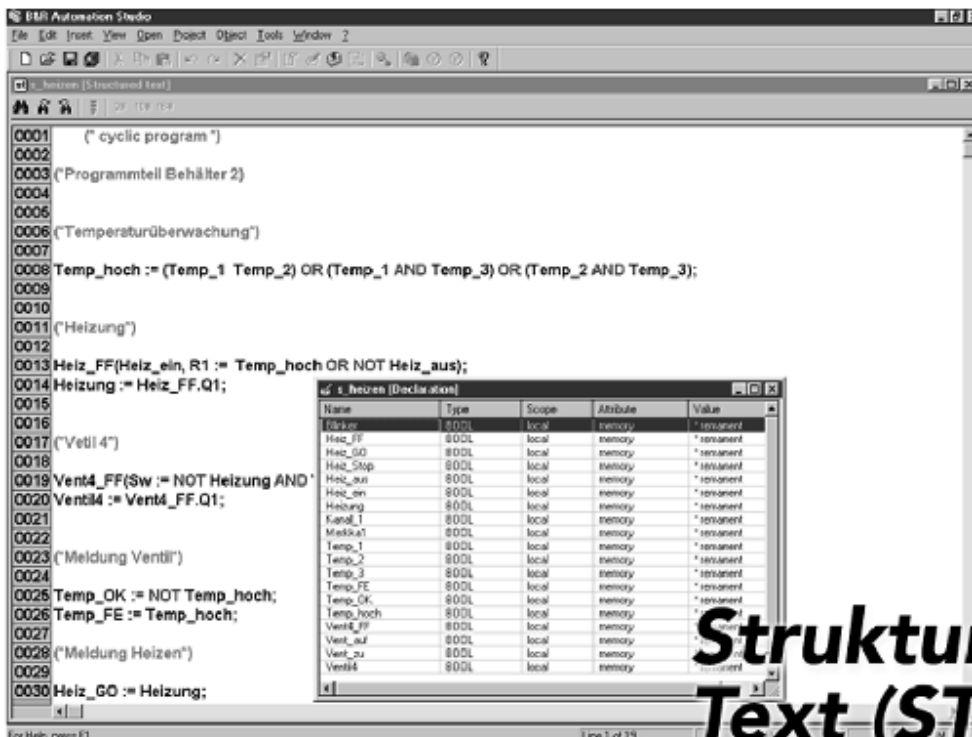
The declaration window shows the following table:

Name	Type	Scope	Attribute	Value
Heiz_GO	BOOL	local	memory	*retained
Heiz_STOP	BOOL	local	memory	*retained
Heizung	BOOL	local	memory	*retained
Temp_1	BOOL	local	memory	*retained
Temp_2	BOOL	local	memory	*retained
Temp_3	BOOL	local	memory	*retained
Temp_FF	BOOL	local	memory	*retained
Temp_OK	BOOL	local	memory	*retained
Temp_hoch	BOOL	local	memory	*retained
Ventil_CL	BOOL	local	memory	*retained
Ventil_OP	BOOL	local	memory	*retained
Ventil_4	BOOL	local	memory	*retained

**Anweisungsliste (AWL)**

IL is a second generation instruction-oriented machine code programming language, similar to an easier assembler language. IL is mostly used for formulating short program sections with logical functions (AND, OR, etc.) or as a common language used with other text and graphics-based languages.

#### 1.1.4.2.4.3 Structured Text (ST)



The screenshot shows the B&R Automation Studio interface with a Structured Text program. The main window displays the following instructions:

```

0001 (" cyclic program ")
0002
0003 ("Programmteil Behälter 2")
0004
0005
0006 ("Temperaturüberwachung")
0007
0008 Temp_hoch := (Temp_1 Temp_2) OR (Temp_1 AND Temp_3) OR (Temp_2 AND Temp_3);
0009
0010
0011 ("Heizung")
0012
0013 Heiz_FF(Heiz_ein, R1 := Temp_hoch OR NOT Heiz_aus);
0014 Heizung := Heiz_FF.Q1;
0015
0016
0017 ("Ventil 4")
0018
0019 Ventil_FF(Sw := NOT Heizung AND
0020 Ventil_4 := Ventil_FF.Q1;
0021
0022
0023 ("Meldung Ventil")
0024
0025 Temp_OK := NOT Temp_hoch;
0026 Temp_FE := Temp_hoch;
0027
0028 ("Meldung Heizen")
0029
0030 Heiz_GO := Heizung;

```

The declaration window shows the following table:

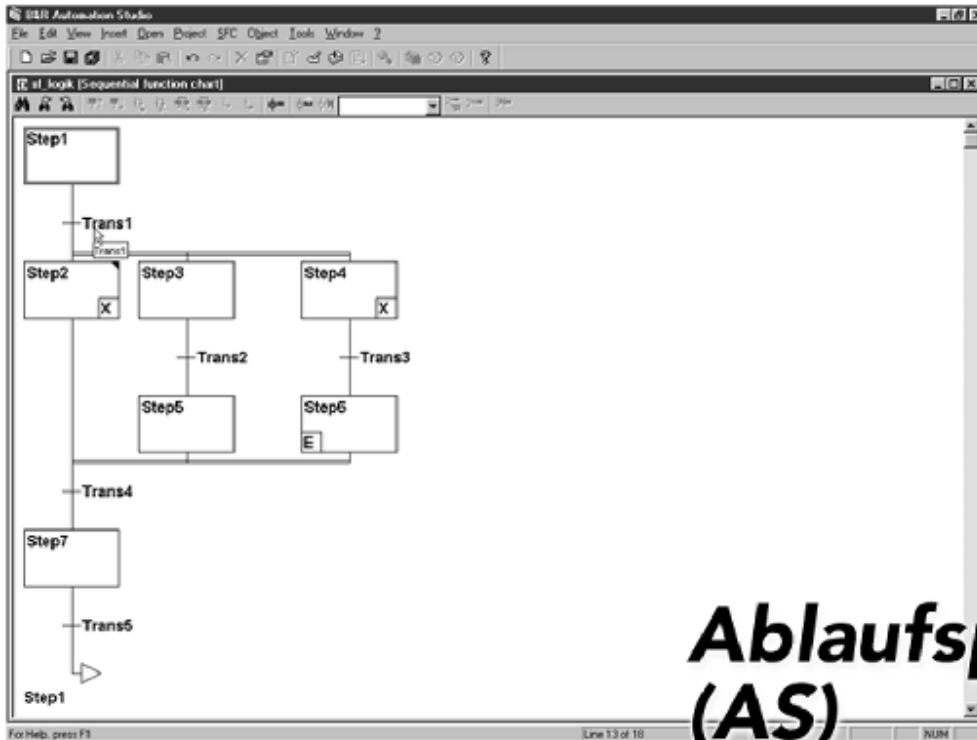
Name	Type	Scope	Attribute	Value
Heiz_GO	BOOL	local	memory	*retained
Heiz_FF	BOOL	local	memory	*retained
Heiz_STOP	BOOL	local	memory	*retained
Heiz_aus	BOOL	local	memory	*retained
Heiz_ein	BOOL	local	memory	*retained
Heizung	BOOL	local	memory	*retained
Fanal_1	BOOL	local	memory	*retained
Melkakt	BOOL	local	memory	*retained
Temp_1	BOOL	local	memory	*retained
Temp_2	BOOL	local	memory	*retained
Temp_3	BOOL	local	memory	*retained
Temp_FE	BOOL	local	memory	*retained
Temp_OK	BOOL	local	memory	*retained
Temp_hoch	BOOL	local	memory	*retained
Ventil_FF	BOOL	local	memory	*retained
Ventil_4	BOOL	local	memory	*retained
Ventil_auf	BOOL	local	memory	*retained
Ventil_zu	BOOL	local	memory	*retained
Ventil4	BOOL	local	memory	*retained

**Strukturierter Text (ST)**

The Structured Text programming language is a high-level language that does not use machine code instructions like IL. Instead it uses abstract commands to create powerful command structures.

Structured Text, like B&R Automation Basic™, is used in all branches of automation technology as a problem-oriented high-level language.

1.1.4.2.4.4 Sequential Function Chart (SFC)

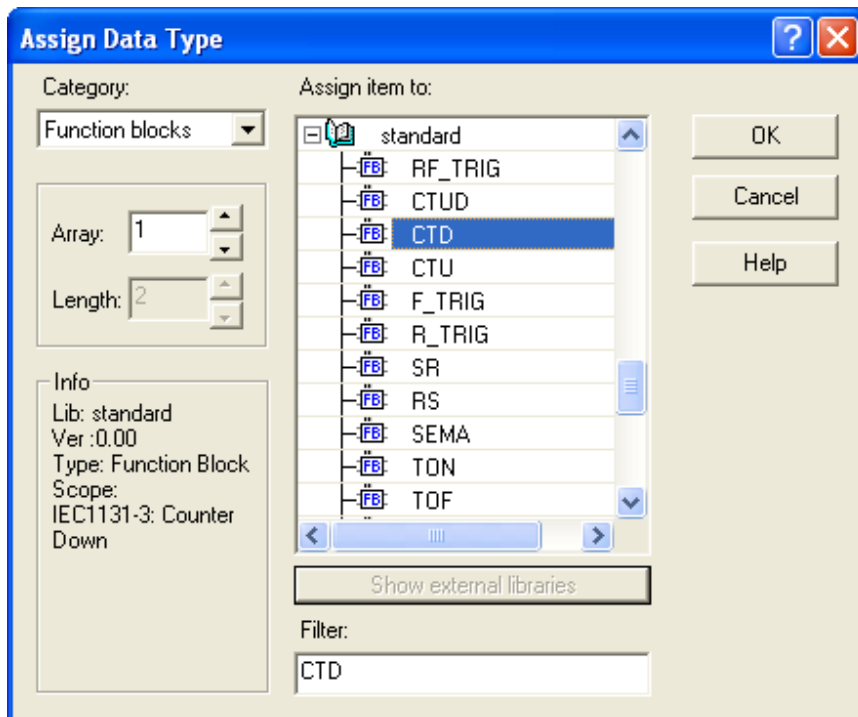


Sequential Function Chart (SFC) was created to separate complex tasks into clear steps and to define control flow between these steps. Sequential processes can be formulated which are executed simultaneously. These steps can be written in another programming language or in SFC.

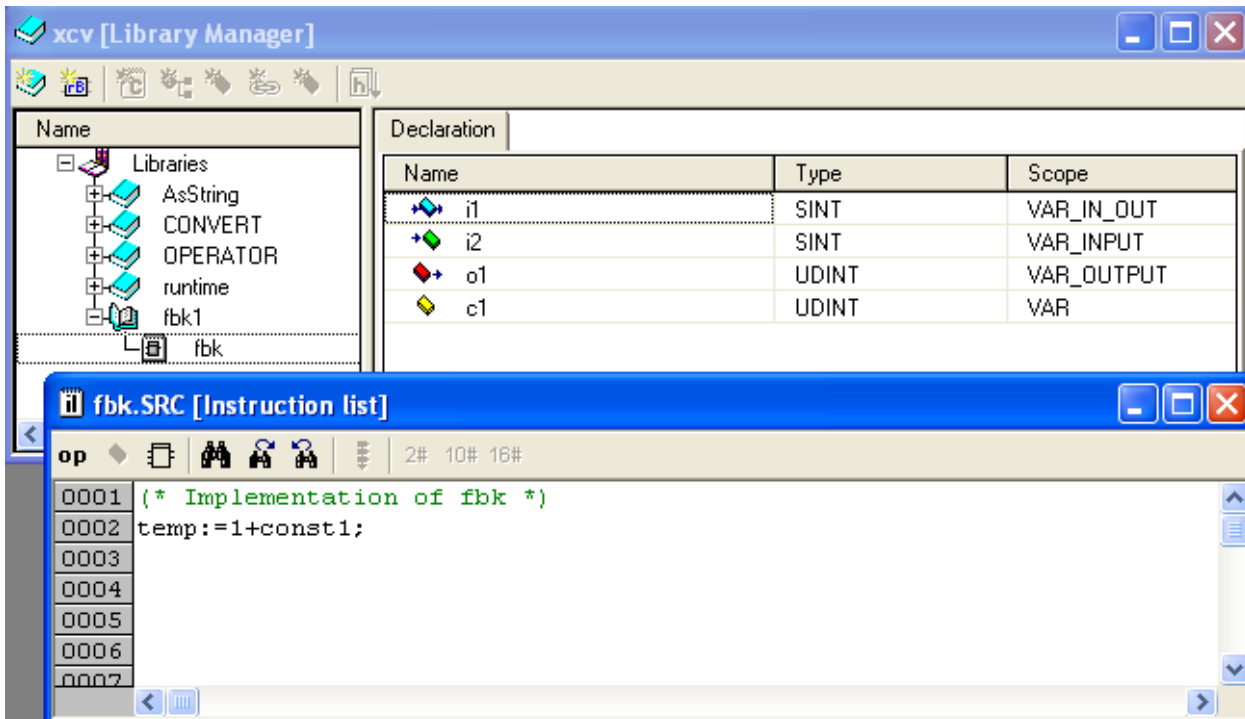
SFC was developed from step sequencer programming and is especially suited for processes with status changes in steps, e.g. a washing machine (prewash, wash, etc.) or a chemical mixing process.

1.1.4.3 Library manager

Program sections that are required in several locations in the program can be stored as subprograms in function blocks. This not only saves development time, it reduces the program size as well. Function blocks are then simply supplied with parameters and called whenever they are needed.

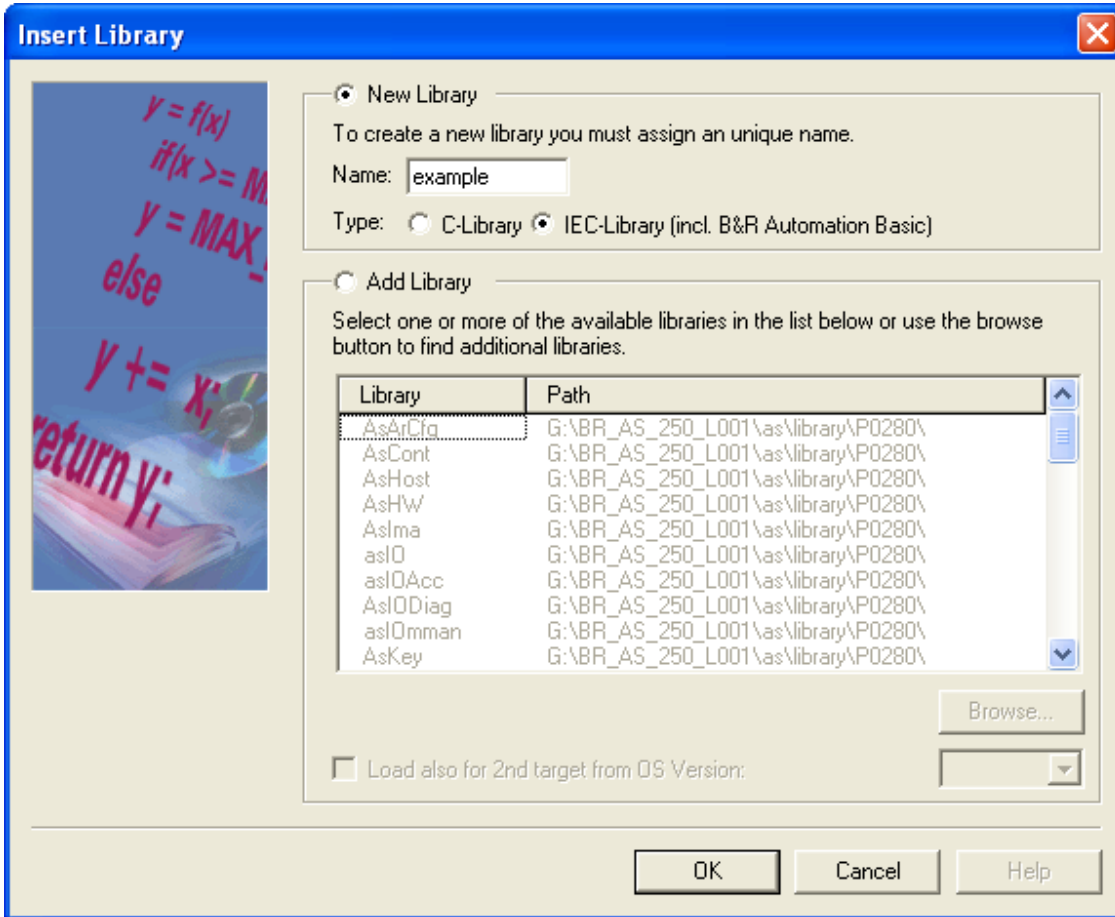


The user can select from a wide variety of standard function blocks that are already integrated into B&R Automation Studio™. These functions range from simple logic and mathematic operations to communication protocols and complex control algorithms. Using function blocks greatly decreases project engineering and development efforts.



Function blocks that have the same or similar subjects are grouped together in libraries and managed using the library manager.

The library manager also allows custom function blocks to be created. Function block interfaces, as well as data types and constants used in a library are entered in a dialog box. Function blocks can be created for all programming languages. The help system can also be expanded to include documentation for custom function blocks.



#### 1.1.4.4 Import / Export

The integrated import/export function allows you to import source code for other systems into the B&R Automation Studio™ or to make your own source code available to other development environments. Transfer takes place using conversion to/from ASCII text. Using existing PL2000 programs and function blocks is also possible.

#### 1.1.4.5 Diagnostic tools

##### 1.1.4.5.1 General information

Just because the program can be compiled without problems after creating the code in B&R program editors does not necessarily mean that the program will run error-free on the target system. The commissioning utilities in B&R Automation Studio™ provide optimal support for analysis and diagnostics of logical and process-oriented problems.

##### 1.1.4.5.2 Online variable monitor

Using the online variable monitor, the current value of all types of process variables (including elements of structures and arrays) can be displayed numerically or logically and can be changed (written to) during operation. Forcing, which means manually setting the input and output states on the system, can also be done in the variable monitor.

##### 1.1.4.5.3 Variable trace

Variable trace is a frequently used commissioning tool. It can be used to graphically display how a variable value changes over time. Fast-changing processes, e.g. processes that change too quickly to be seen with the naked eye in the variable monitor, can be represented in a diagram and even measured (e.g. to determine the time between two status changes) using measuring cursors.

Variable trace is configurable in B&R Automation Studio™. For example, the digital input that should be used to start recording the measurements can be defined. Measurements are recorded in-sync on the B&R PLC. This guarantees that no values are "overlooked". Measurements are recorded even when B&R Automation Studio™

is not connected to the PLC! These measurements are recorded in a data buffer on the PLC and can be uploaded and displayed / analyzed using B&R Automation Studio™ at a later time.

The screenshot displays the B&R Automation Studio interface. On the left, a 'Name' list shows variables like 'Prozent\_ein' and 'Analog\_in'. The main area shows several variable traces (e.g., 'Prozent\_ein', 'Analog\_in') with their values over time. On the right, a 'Variablen-Trace' window shows the current values of these variables: fehler = FALSE, high = FALSE, low = FALSE, ok = FALSE, Prozent\_ein = 74, and Analog\_in = 24480. Below the traces, a 'Source-Level-Debugger' window shows the source code with a breakpoint set at line 0015, 'action 21.99'. The code includes a case statement for 'Prozent\_ein' with actions for 'low', 'high', and 'fehler'.

#### 1.1.4.5.4 Debuggers

Debuggers (source code level commissioning utilities) can be used to insert breakpoints into the program. The program can be executed in single steps or procedure steps starting from these breakpoints. If the "Line Coverage" function is enabled, the program lines currently being executed are highlighted in the source code.

#### 1.1.4.5.5 System logbook

If a fatal error or critical change of state occurs, the system enters time stamped messages in the system logbook for the controller. They can be displayed using B&R Automation Studio™. It is also possible for the user to enter additional system messages, such as operation malfunctions, in the system logbook using function blocks. These messages can also be displayed using B&R Automation Studio™.

#### 1.1.4.6 Service

Programming in B&R Automation Studio™ can be done over a serial connection or network/modem connection. The connection can even switch between different media (e.g. from RS232 to CAN). This routing function makes remote diagnosis and programming very easy.

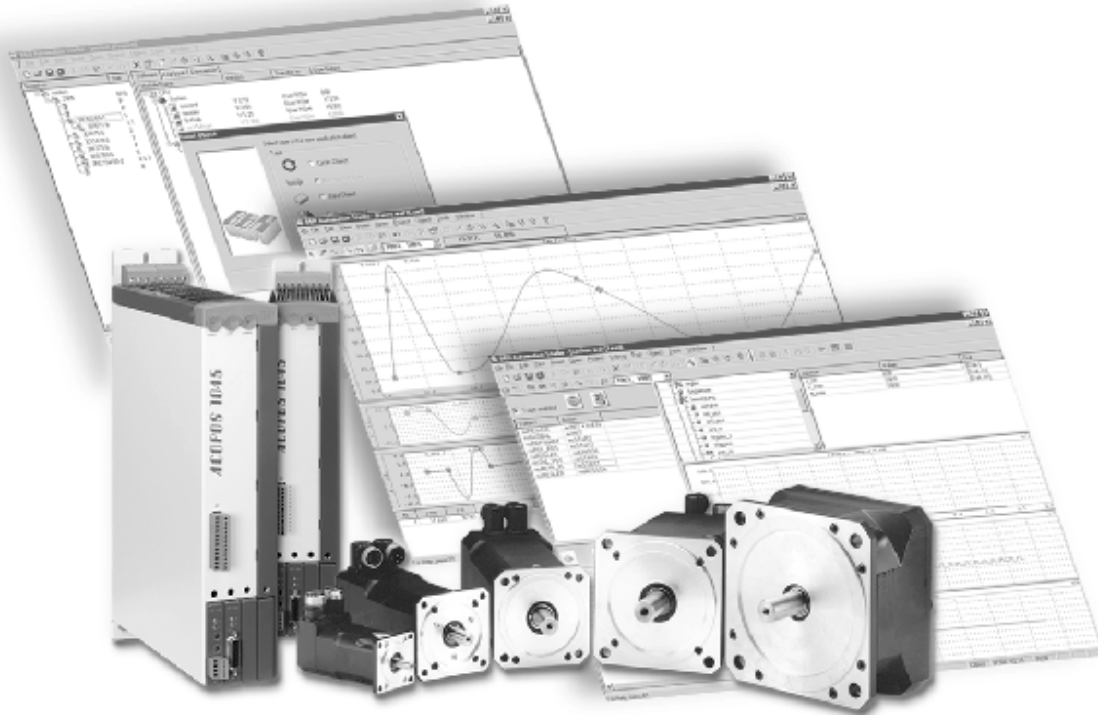
Service features also include erasing the application memory and updating the operating system on the controller. A cold restart (starting the controller and reinitializing all variables) and a warm restart (starting the controller with the variable values used before the controller was turned off) are also included in the service features.



## 1.1.4.7 Motion Components

### 1.1.4.7.1 General information

Motion Components provides an integrated solution for the development, configuration, and diagnosis of gear, positioning, and cam profile applications. The axes are integral components of the project (objects). Axis parameters are edited and managed in the project.



### 1.1.4.7.2

#### 1.1.4.7.3 Parameter editor

The parameter editor allows you to edit, copy, and display initialization parameters for an axis (e.g. encoder interface, limit values, reference modes, etc.). These parameters are displayed as plain text and physical units.

#### 1.1.4.7.4 Diagnostics using the variable monitor

The monitor displays all relevant data while the axis is being operated (e.g. movement active/inactive, controller on/off, simulation mode on/off, errors yes/no, etc.). Any parameters and axis states can be chosen. Simultaneous diagnosis of multiple axes is also possible.

#### 1.1.4.7.5 Diagnostics using the oscilloscope function

Up to 10 channels can be displayed simultaneously by defining trigger events such as start movement, end movement, or manual trigger. Powerful zoom and measurement cursor functions are also integrated for evaluation.

#### 1.1.4.7.6 Commissioning using an efficient testing environment

A special testing environment allows all the necessary initialization and movement parameters for the axis to be set or changed. The integrated oscilloscope function (for displaying movement data like position, speed, and lag errors) makes it very easy to optimize movements.

### 1.1.4.7.7 Cam profiles

This is a tool that makes it easy to create custom movements (complicated calculations are no longer necessary). Movements can be created and tested directly in the editor for fast and optimal results. The movement can be easily analyzed by displaying higher derivatives (jolt) and dynamic torque (power).

### 1.1.4.7.8 Cam editor

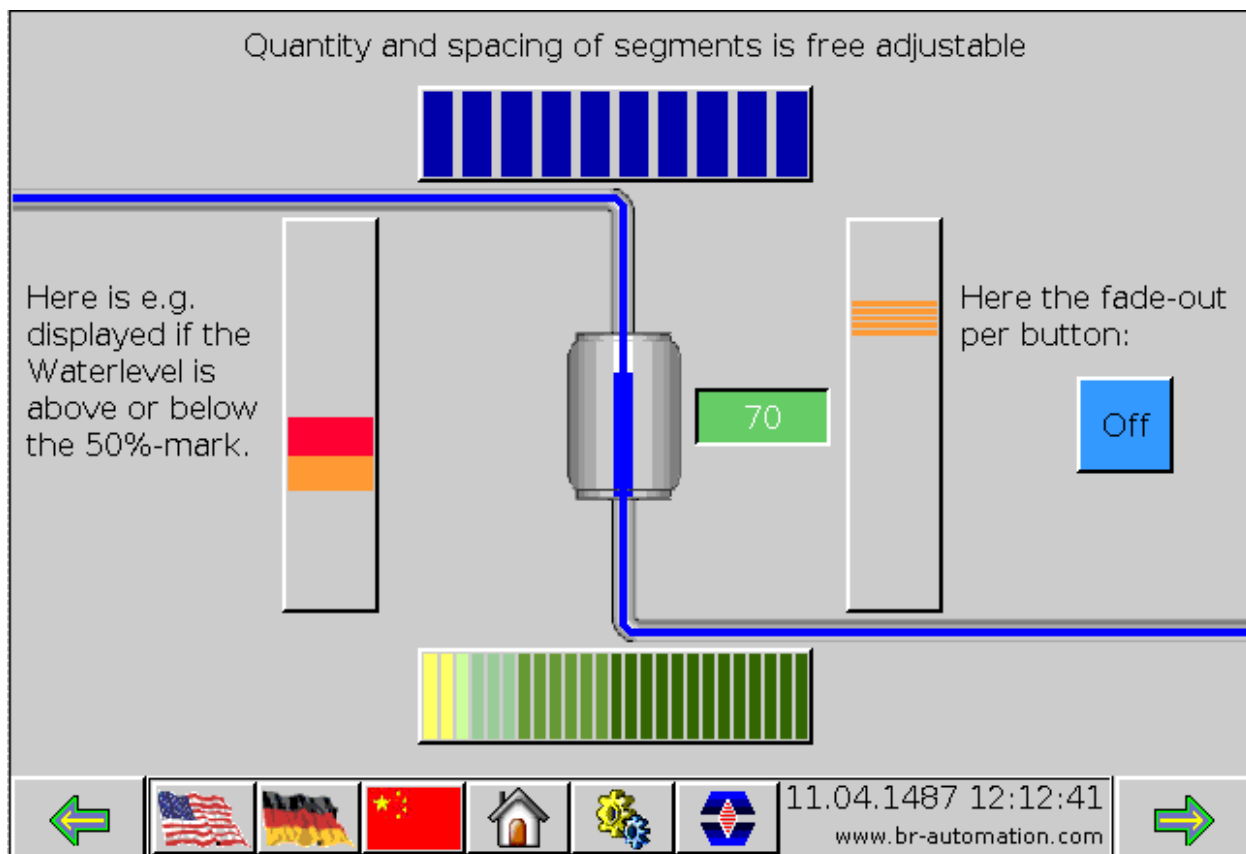
The cam editor is used to create cam profiles according to VDI2143 guidelines. Add-ons for fixed points, synchronous sections, and splines with three segments round off the editor functions. Cam profiles are optimized according to position, speed, acceleration, jolt, and dynamic torque (power).

Powerful editor functions allow curves to be easily modified using the cursor and provide a clear workflow even with complex associations. The curves can be displayed mathematically or physically. The physical display also allows monitoring of static and dynamic (accelerated master) limits.

## 1.1.4.8 Visual Components

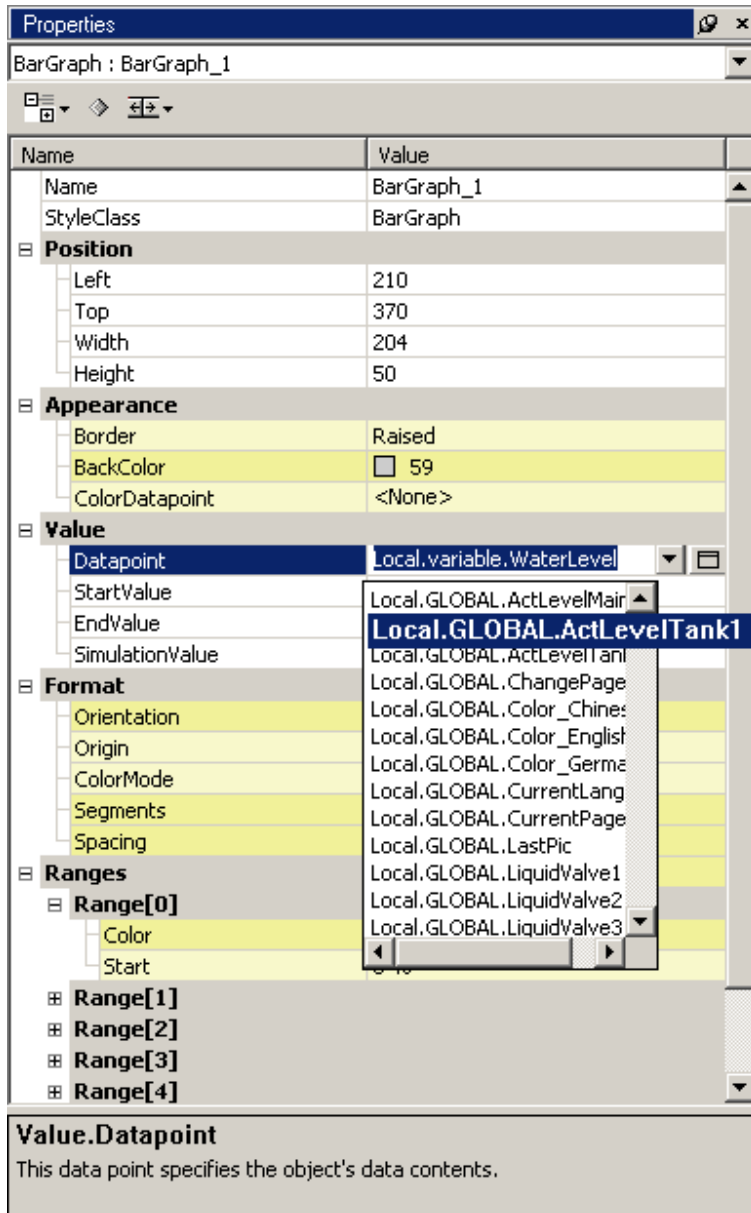
### 1.1.4.8.1 Solving visualization and control tasks together

Using Visual Components in B&R Automation Studio™, process pictures and other visualization elements can be created and tested together with the control task. This results in optimal interaction between visual and logic elements. This considerably shortens the time required for project engineering.



### 1.1.4.8.2 Picture editor

Process pictures are created with the Visual Components picture editor, which is shown in B&R Automation Studio™ when creating or editing a panel object.



The picture editor adapts to the hardware. This means that the picture editor only allows functions that apply to the selected device. For example, if a project is created for a Power Panel PP21, the area in the picture editor which can be edited is limited to 4x20 characters, and the 34 keys are displayed in the correct arrangement. On the other hand, if a B&R IPC with a touch screen is used, the area which can be edited is expanded to 640x480 pixels, and softkeys are displayed.

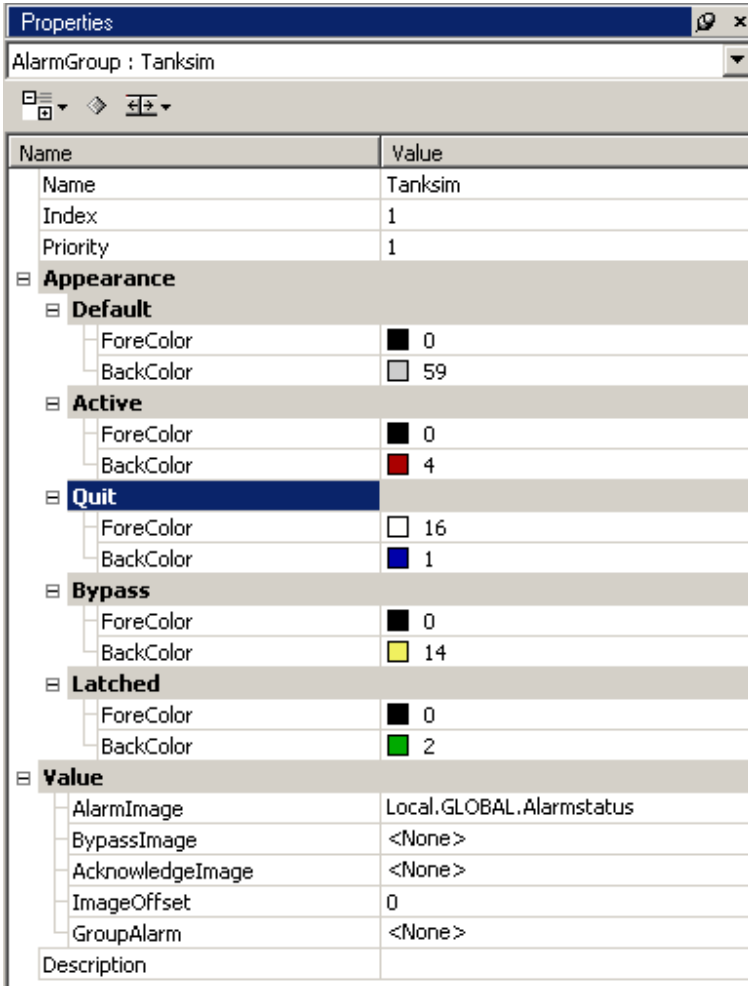
In text mode, the picture editor allows the placement of text, bars graphs and input/output fields. In graphics mode, additional graphic features such as lines, rectangles or bitmaps are available. A bitmap collection simplifies the creation of frequently used picture elements.

Dynamic controller values are selected from existing process variables available in the controller database and "connected" when the process picture is created.

Text can be entered and managed in multiple languages. During runtime, texts can be switched online to the language desired (including Asian fonts).

#### 1.1.4.8.3 Alarm system

Alarms can be added, deleted, or individually configured in the alarm editor. Individual alarms from one group can be combined to form a group alarm.



Alarms are managed as bit arrays on the controller. One bit array is assigned to each alarm group. If the bit is set to 0 (FALSE), the alarm is inactive. If it is 1 (TRUE), the alarm is active and is displayed in the alarm list. Alarms are displayed using alarm fields. The alarm fields display the current status of the alarms during runtime. The following general alarm fields are currently available:

- Alarm line
- Alarm list
- Alarm summary
- Alarm history

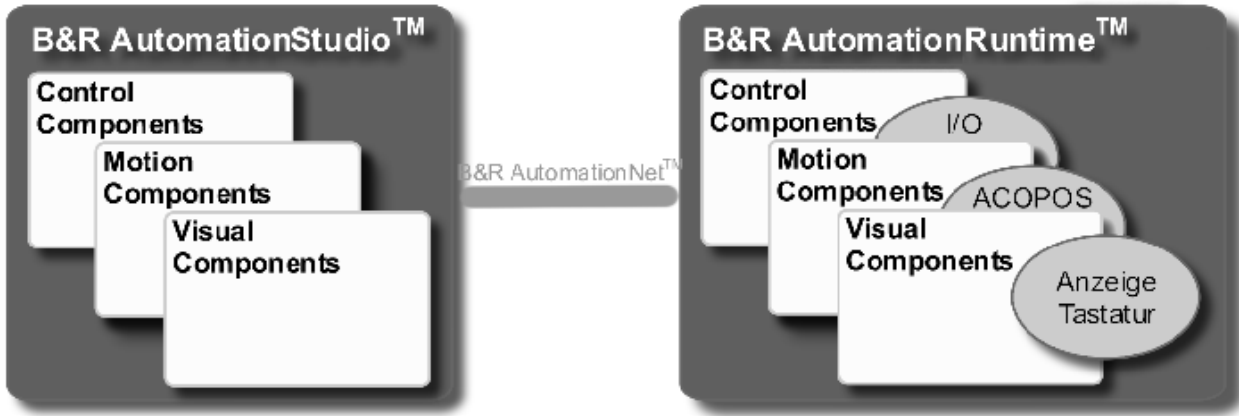
Properties of an alarm can be set individually. You can specify whether the alarm has to be acknowledged, whether it can be bypassed, and whether it can trigger a group alarm. Printout parameters can also be set individually. A line can be sent to the printer depending on how the alarm is configured.

#### 1.1.4.8.4 Transferring the project to the target system

Compiling and transferring the project to B&R Automation Runtime™ is also done together here. In other words, interdependencies between the visualization and control program are checked, recompiled if necessary, and transferred to B&R Automation Runtime™ via B&R Automation Net™. B&R Automation Net™ allows data to be transferred via a serial, CAN, or ETHERNET connection.

#### 1.1.4.8.5 Runtime

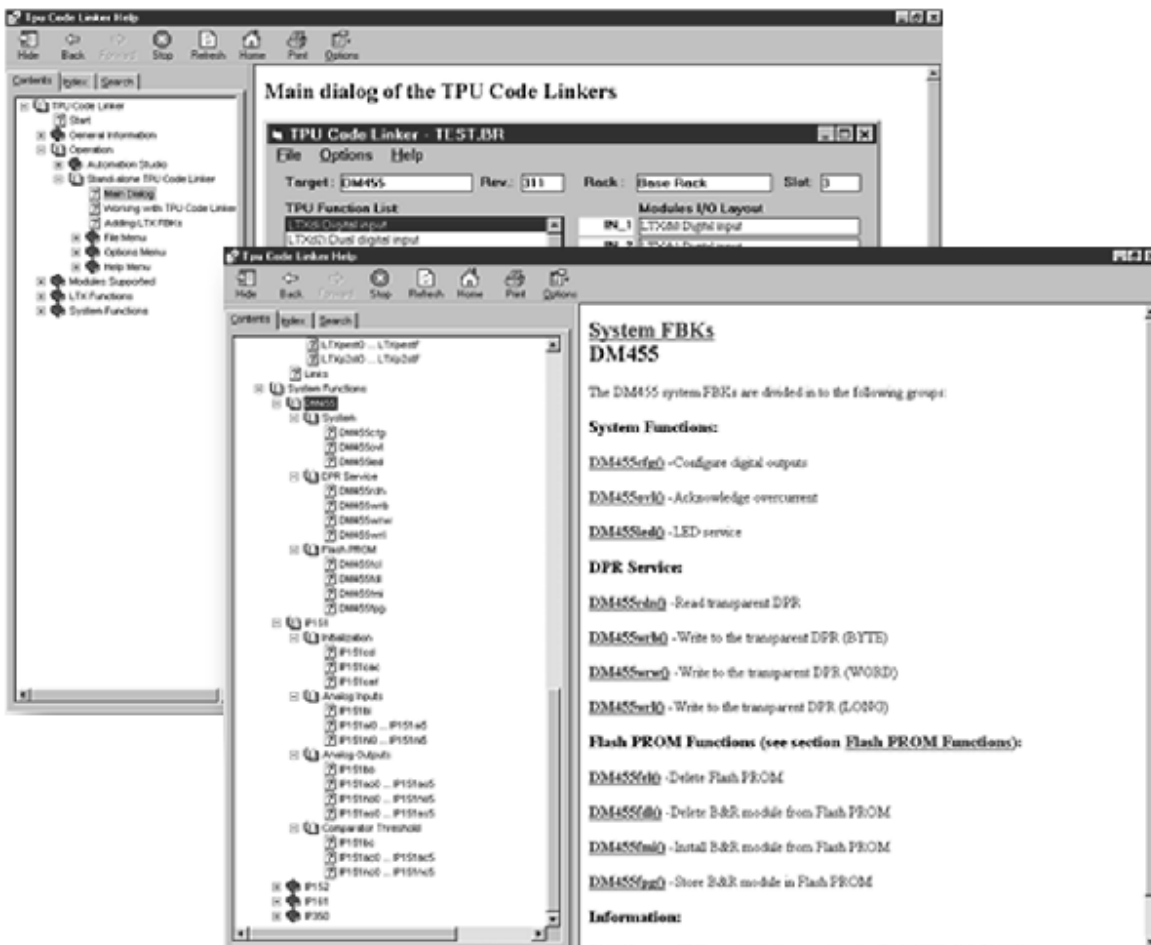
B&R Automation Studio™ was not the only software expanded to include Visual Components. B&R Automation Runtime™, the operating system for B&R target systems, was also expanded to include software components that allow the display and operation of process pictures.



Because of the direct connection between visualization and controller in B&R Automation Runtime™, there is no delay when accessing variables during runtime, and the display is generated at optimum speed. Keyboard operation also directly influences the controller (important for key operation).

**1.1.4.9 TPU code linker**

A TPU is an integrated module from Motorola's 32-bit M68300 microcontroller family. It can be used to carry out timing tasks without placing an extra load on the CPU. Each incoming action triggers an event. The TPU handles the processing of these events so that the CPU does not have to run interrupt routines. Since some B&R modules (e.g. CP474, IP161, etc.) use processors with a TPU, the TPU can be put to use with the help of the TPU code linker.



The following timing functions are supported:

- Input recognition / input edge counter
- Output comparator
- Pulse width modulation
- Synchronized pulse width modulation

- Period measurement
- Period measurement with edge recognition
- Position synchronized pulse generator
- Stepper motors
- Gate measurement

The TPU code linker allows the functions described above to be assigned to the 16 channels on the TPU which enables the corresponding functions on the connected digital inputs or outputs. The preset values or results are transferred to/from the application program using function blocks.

## 1.1.5 B&R Automation Net™

### 1.1.5.1 Transparent communication with B&R Automation Net™

With the philosophy "One Tool, Many Targets", B&R Automation Software™ provides the possibility to program and configure all types of B&R systems with one uniform development tool. However, this means that communication that takes place between these components also has to be uniform to guarantee that communication is transparent for all stations.



Each station must be able to exchange and process program objects and/or process variable objects

- Regardless of the operating system being used (PCCSW, B&R Automation Runtime™, Windows™ 98/ME/NT/2000/XP)
- Regardless of the media being used (RS232, CAN, ETHERNET, POWERLINK, PROFIBUS, modem, memory, etc.),
- Regardless of the transfer protocol being used (INA2000, Net2000, Mininet, etc.).

One of the primary tasks of B&R Automation Net™ is to always be able to establish and maintain an optimal connection between intelligent modules from B&R.

### 1.1.5.2 Accessing B&R Automation Net™

Communication within B&R Automation Net™ can be transparent, but different interfaces must be used at the end points where the information is actually processed on the respective operating systems.

Each operating system thus requires a corresponding interface to B&R Automation Net™. Therefore, the obvious choice was to use a function block interface (INA Client FBKs) to access PLC systems and, for example, a DLL interface (PVI) to access B&R Automation Net™ on Windows™ systems.

### **1.1.5.3 Process Visualization Interface (PVI)**

#### **1.1.5.3.1 General information**

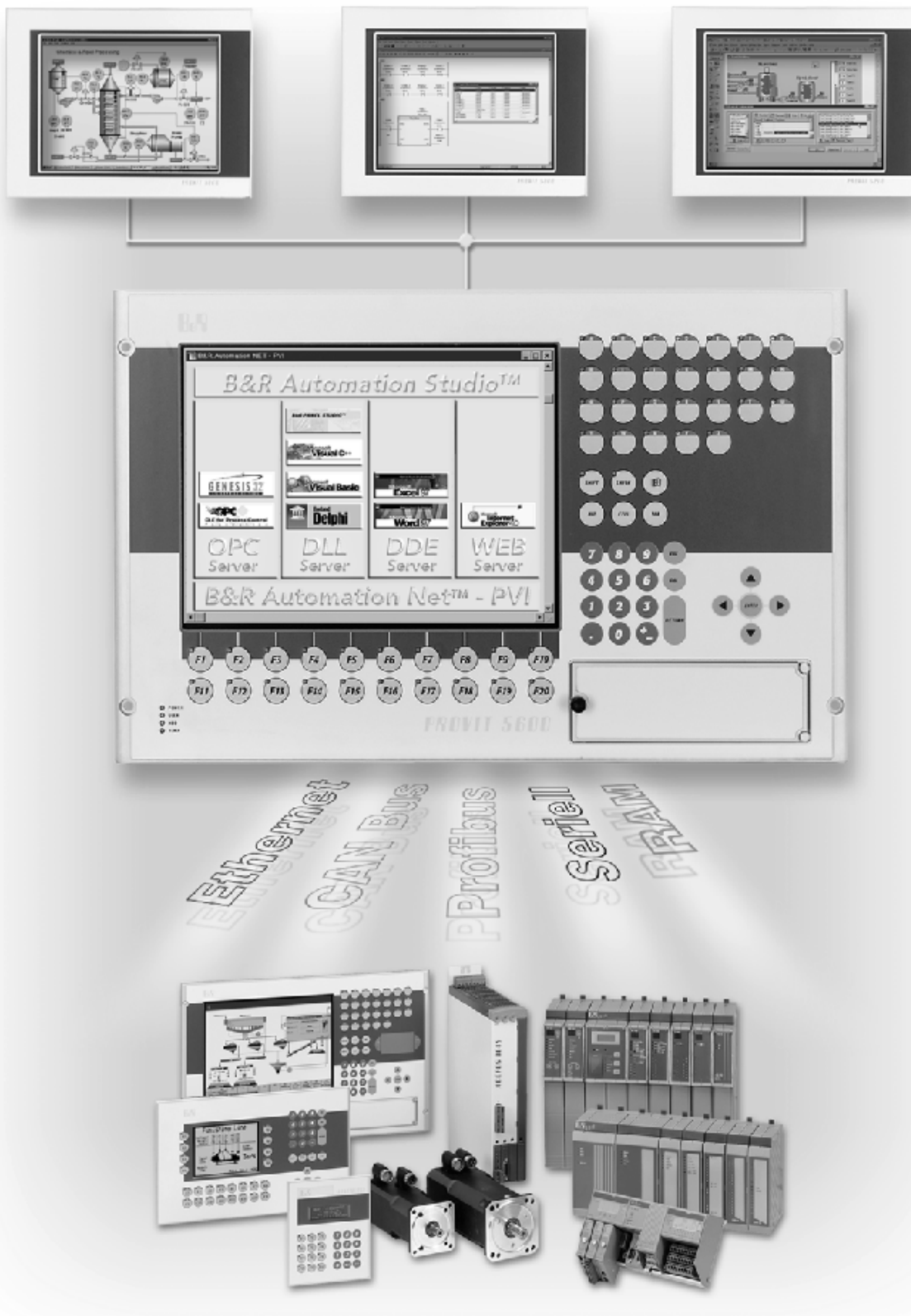
Using the Process Visualization Interface (PVI) to access B&R Automation Net™ creates a common interface to the B&R industrial PC environment for all Windows™-based software packages.

#### **1.1.5.3.2 Centralized communication on the IPC**

With PVI, you don't just get uniform, coordinated access to various media and protocols. It also assists the user when working with all types of process data (variables, modules, status, etc.) and when controlling connected devices (Start, Stop).

During the development of PVI, the main focus was on incorporating all services and transfer possibilities required for B&R Automation Studio™ and all other B&R standard packages. This is also the advantage for the user. Incorporating these programming device functions provides the user with additional options. For instance, programs can be downloaded and uploaded without even having to start B&R Automation Studio™.

These functions are provided to the user and all B&R standard packages via a PVI client interface. This interface was developed as a Windows DLL (Dynamic Link Library). This represents the "tightest" and, with regard to performance, the most optimal interface.



### 1.1.5.3.3 Client/Server technology

The PVI client/server principle allows PVI applications to be operated remotely. That means PVI applications (in particular B&R Automation Studio™) which function as PVI clients can transmit/receive process variables and programs to/from the PVI server via networks, modem connections, or other communication media. Changing the PVI application is therefore not necessary, regardless of whether its being run locally or remotely.



**1.1.5.3.4 Open for other architectures**

Using additional PVI expansions, it is possible to link architectures such as DDE, OPC, or HTML to PVI and then send them on to B&R Automation Net™. These PVI expansions were developed as expansion servers and are included in the PVI standard package.

**1.1.5.4 PVI interface types**

**1.1.5.4.1 PVI DDE server**

Many programming environments and older applications use the Dynamic Data Exchange interface (DDE) to exchange data. This interface uses the server name, topic, and item to specify addresses.

**1.1.5.4.2 PVI Web server**

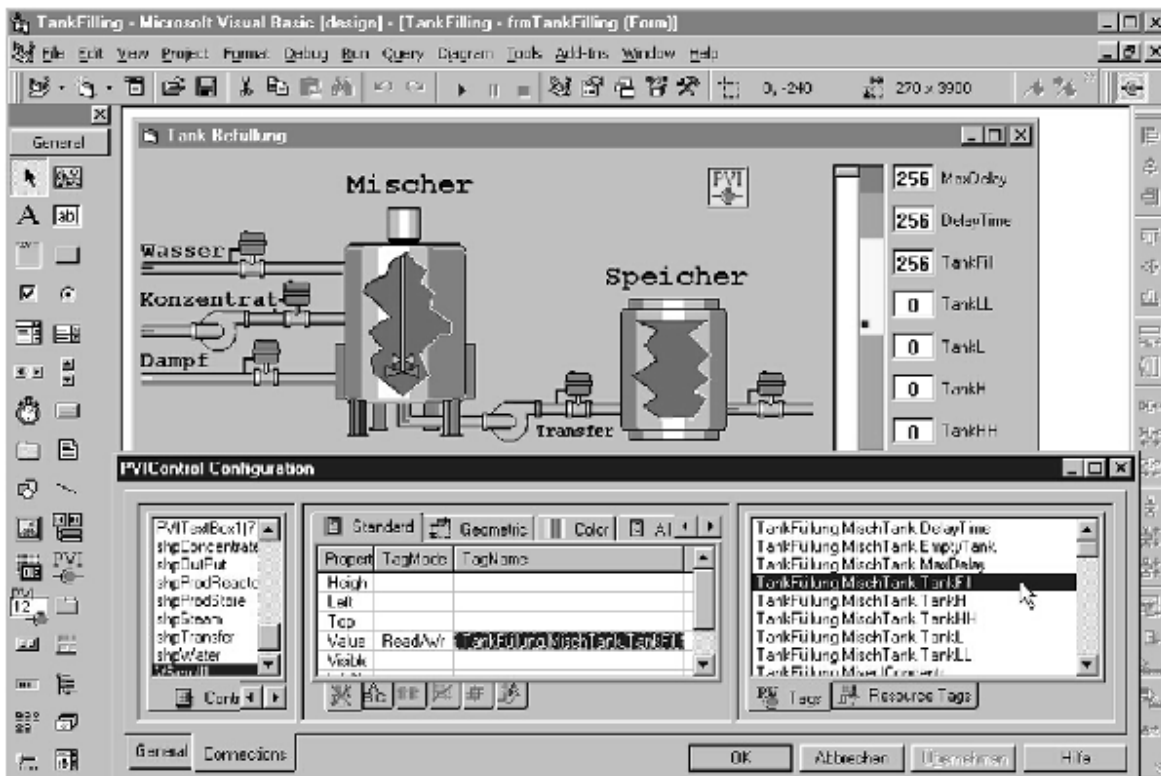
The PVI Web server was created for applications where a standard Web browser by itself is sufficient for displaying process data. For this, an IPC connected to the controller and running the Web server is required. Another PC with a standard Internet browser can establish a connection to the Web server via a network (e.g. intranet or dial-up). After the connection is made, the process data is displayed on the screen.

**1.1.5.4.3 PVI OPC server**

"OLE for Process Control" is a standardized interface mostly used for visualization or SCADA packages (Supervisory Control and Data Acquisition). It is essentially used to carry out commonly formulated read/write instructions on controllers. The PVI OPC server is also used by our SCADA package Genesis32™ to access B&R Automation Net™.

**1.1.5.4.4 PVI Control**

Another interface is provided with PVI Controls. It is a interface to B&R Automation Net™ specially suited for Visual Basic and VBA programmers, e.g. for communication with B&R PLC networks. The main focus was placed on simple operation. In a short amount of time, PLC variables can be displayed by simply "connecting" them to Visual Basic standard controls without additional programming. Setting parameters instead of programming is the strategy employed.



This simple configuration is coupled with powerful support for online language changes. The respective formats, scaling functions, fonts (including Asian) and hysteresis monitoring are also available in addition to displaying messages in other languages.

PVI Controls therefore create a convenient and efficient interface for small and mid-sized applications for first-time programmers as well as visualization experts.

#### 1.1.5.4.5 INA2000 Client FBKs

PLC programs access B&R Automation Net™ using [INA2000 Client FBKs](#). These function blocks not only allow process variables to be read from a PLC connected to the network; it is also possible to actively send process variables from one PLC to another. Cyclically polling variables is no longer necessary. This drastically decreases the load on transfer lines. Transferring buffered data (e.g. messages, status changes with time stamp) is also possible.

### 1.1.6 B&R Automation Runtime™

#### 1.1.6.1 What is B&R Automation Runtime™?

Runtime systems are software environments where programs can be run and tested.

At B&R, programs are created using B&R Automation Studio™. These programs are then transferred to B&R Automation Runtime™ system via B&R Automation Net™. They can be run and tested there.

B&R Automation Runtime™ is fully embedded in the corresponding B&R automation target (this is the hardware where B&R Automation Runtime™ is installed). It allows application programs to access I/O systems (e.g. via fieldbus) and other devices (interfaces, networks, etc.).

#### 1.1.6.2 One runtime system for all B&R automation targets

B&R Automation Runtime™ completely takes over the B&R automation target (target hardware) and creates a uniform interface between the application program and the "device environment".

Using a uniform interface provides the user with a decisive advantage: switching from one automation target to another requires little or no changes to be made to the source code. Program source code can basically be used by all B&R Automation Runtime variations and exchanged between them.

For the many users who are already familiar with our 2003/2005/2010 PLC systems and our PCCSW operating system software, there is no change of philosophy. PCCSW can actually be seen as the predecessor of B&R Automation Runtime™. B&R Automation Runtime™, as a further development of PCCSW, should be viewed as a collective term for all B&R controller operating systems. The functions previously available in PCCSW are consequently also provided in B&R Automation Runtime™.

#### 1.1.6.3 B&R automation targets in the industrial PC family

B&R Automation Runtime™ is used on three B&R automation targets from the B&R IPC family.

##### 1.1.6.3.1 AR010 for B&R IPC5000 with Windows™ 2000 or Windows™ XP

A real-time operating system is imposed over the operating system (Windows™ XP or Windows™ XP Embedded) which takes total control of PC resources and handles the operating system as the task with the lowest priority within this real-time operating system.

B&R Automation Runtime™ AR010 is based on this real-time operating system. Therefore, the control tasks running on B&R Automation Runtime™ AR010 have priority over operating system tasks. They even continue running if Windows™ XP or Windows™ XP Embedded no longer appears to be functioning.

2005 or 2010 controller expansions or CAN I/O is used for the I/O connection.

### 1.1.6.3.2 AR102 for B&R IPC2001 Embedded Version

B&R Automation Runtime™ AR102 is based on an embedded real-time operating system that was adapted specifically for our controller tasks. No Windows™ based components are necessary.

CAN, ISA I/O, and other systems are used for the I/O connection.

### 1.1.6.3.3 AR105 for B&R IPC5000 Embedded Version

B&R Automation Runtime™ AR105 is based on the same embedded real-time operating system as AR102. However, it has been adapted for the IPC5000. CAN, ISA I/O, PCI bus, and other systems are used for the I/O connections.

## 1.1.6.4 Installation kit

### 1.1.6.4.1 General information

If B&R Automation Runtime™ is not already on B&R automation devices from the B&R IPC family, it must first be installed. This is done using installation diskettes which are created using the "B&R Automation Runtime™ Installation Kit". This installation kit is included on the B&R Automation Software™ CD.

The B&R Automation Runtime™ Installation Kit is a MS Windows™-based program. It is run on the development machine from the menu program (started when the CD is inserted). After entering certain options concerning the target device, the installation kit creates a set of diskettes for the target system.

Diskettes are used to copy data because these are the only media that can be used on all IPCs. CD drives are not always available, and serial or parallel communication is slow and hard to configure.

The installation diskettes created on the development computer with the B&R Automation Runtime™ Installation Kit include a bootable diskette and several subsequent diskettes with compressed files.

After booting the system with the boot disk, the user is prompted to insert the follow-up disks until B&R Automation Runtime™ is completely installed.

After completing the installation and installing the hardware security key, B&R Automation Runtime™ can be run on the target system after it is rebooted. The automation project can then be loaded and tested using B&R Automation Studio™.

### 1.1.6.4.2 Typical use

#### 1.1.6.4.2.1 Creating the installation diskettes

- Insert the B&R Automation Software™ CD
- Start the B&R Automation Runtime™ Installation Kit for AR102 or AR105
- Select device options and create installation diskettes

#### 1.1.6.4.2.2 Installing B&R Automation Runtime™ on target systems

- Attach the license sticker.
- Install the hardware security key.
- Insert the first installation diskette.
- Call up the partitioning options (partitions the hard disk or CompactFlash).
- Boot the target system.
- Call up the formatting options (formats the hard disk or CompactFlash).
- Boot the target system.
- Start the installation (installs B&R Automation Runtime™).
- Insert the follow-up diskettes.
- After rebooting, the target system is ready for operation

## 1.1.6.5 Ordering licenses

B&R Automation Runtime™ is the runtime system for automation projects developed with B&R Automation Studio™. It is either pre-installed on the B&R automation device or it is installed on the B&R automation device by the user using a diskette set.

**Note:**

A B&R Automation Runtime™ installation disk set is created using the B&R Automation Runtime™ Installation Kit. This installation kit is included on the B&R Automation Software™ CD.

For B&R automation devices with B&R Automation Runtime™ already installed (e.g. newer generation B&R SYSTEM 2005), it is not necessary to order a B&R Automation Runtime™ license separately since it is included with the B&R automation device.

For B&R automation devices without B&R Automation Runtime™ installed (e.g. IPC2001), a runtime license must be ordered separately for each B&R automation device. It may be the case that the type of operating system to be used on the device could not be determined before the IPC was ordered.

The B&R Automation Runtime™ license consists of a license sticker, which is attached to the B&R automation device, and a hardware security key, which is installed in or connected to the device.

There are different types of B&R Automation Runtime™ licenses available depending on the type of the B&R automation device.

**1.1.6.6 Runtime system ordering information**

Runtime systems for executing and diagnosing automation projects that are developed with B&R Automation Studio™.

Model number	Description
1A4600.10	B&R Automation Runtime™ AR010 Operating license for AR010 on the IPC5000/Windows™ NT, consists of hardware copy protection and license sticker
1A4600.20	B&R Automation Runtime™ AR010/Desktop Operating license for AR010 on desktop PCs, consists of hardware copy protection, key holder, and license sticker
1A4601.02	B&R Automation Runtime™ AR102 Operating license for AR102 on the IPC2001, consists of hardware copy protection and license sticker
1A4601.05	B&R Automation Runtime™ AR105 Operating license for AR105 on the IPC5000, consists of hardware copy protection and license sticker

**1.1.6.7 Automation Runtime™ AR010****1.1.6.7.1 Features**

AR010 for B&R IPC5000 with Windows™ XP (mod. no. 1A4600.10).

Features	Description
Automation target	The automation target for AR010 is the IPC5000.
B&R Automation Net™ / PVI	The RS232 interface (COM2) can be used as the online interface. If B&R Automation Studio™ and AR010 are installed on the same computer, a local TCP/IP connection can be used.
I/O connections	LS187, LS189, and LS172 drivers are provided for I/O connections.
Floating point processor	With Intel processors, using the FPU on the processor has been included to perform calculations.

AR010 can only be installed on a computer with an already functioning copy of Windows™ XP. Once it has been successfully installed, AR010 can communicate with B&R Automation Studio™. Any AR010 upgrades are installed from the AR010 setup files.

### 1.1.6.7.2 Ordering information

Each copy of B&R Automation Runtime™ AR010 (mod. no. 1A4600.10) is delivered with a runtime license in the form of a security key (hardware dongle), as well as a license sticker.

AR010 operation is only possible on IPC5000 devices if the security key is installed in the target computer or attached to one of the computer's LPT interfaces using a key ring (mod. no. 9A0003.01, not included in delivery).

The B&R Automation Software™ CD (see section "Installation Kit") is required to install B&R Automation Runtime™ AR010 on the automation target (target machine).

### 1.1.6.8 Automation Runtime™ AR010/Desktop

#### 1.1.6.8.1 Features

AR010 for Desktop PCs with Windows™ XP (Mod. No. 1A4600.20)

The features of B&R Automation Runtime™ AR010/Desktop are identical to those of B&R Automation Runtime™ AR010 for the IPC5000.

The only difference is the possibility to operate the runtime system on a conventional PC. In this case, an additional attachment for the security key is included with delivery.

#### 1.1.6.8.2 Ordering information

Each copy of B&R Automation Runtime™ AR010/Desktop (mod. no. 1A4600.20) is delivered with a runtime license in the form of a security key (hardware dongle), a key attachment for the security key, and a license sticker.

AR010/Desktop can be operated on desktop PCs as well as on IPC devices if the copy protection is attached to an LPT interface on the target machine using the key attachment included with the delivery or installed in the target machine.

The B&R Automation Software™ CD (see the "Installation Kit" section) is required to install B&R Automation Runtime™ AR010/Desktop on the automation target (target machine).

### 1.1.6.9 Automation Runtime™ AR102

#### 1.1.6.9.1 Features

AR102 for B&R IPC200x with an embedded real-time operating system (mod. no. 1A4601.02).

Features	Description
Automation target	Automation targets for the AR102 include all IPCs from the B&R IPC2000 family.
B&R Automation Net™ / PVI	RS232 (COM1 – COM4), CAN, or Ethernet can be used as an online interface.
I/O connections	CAN or ISA I/O can be used for the I/O connections.
Floating point processor	With Intel processors, using the FPU on the processor has been included to perform calculations.

AR102 can be installed on an IPC2000 with the B&R Automation Runtime™ Installation Kit. Once it has been successfully installed, AR102 can communicate with B&R Automation Studio™. Possible AR102 upgrades are made using B&R Automation Studio™ in the form of operating system upgrades or installed using the B&R Automation Runtime™ Installation Kit.

### 1.1.6.9.2 Ordering information

Each copy of B&R Automation Runtime™ AR102 (mod. no. 1A4601.02) is delivered with a runtime license in the form of a security key (hardware dongle), as well as a license sticker.

AR102 operation is only possible on IPC2000 devices if the security key is installed in the target machine or attached to one of the computer's LPT interfaces using a key ring (Model Nr. 9A0003.01, not included in delivery).

The B&R Automation Software™ CD (see the "Installation kit" section) is required to install B&R Automation Runtime™ AR102 on the automation target (target machine).

### 1.1.6.10 Automation Runtime™ AR105

#### 1.1.6.10.1 Features

AR105 for B&R IPC5000 with an embedded real-time operating system (mod. no. 1A4601.05).

Features	Description
Automation target	Automation targets for the AR105 include all IPCs from the B&R IPC5000 family.
B&R Automation Net™ / PVI	RS232 (COM1 – COM4), CAN, or Ethernet can be used as an online interface.
I/O connections	LS192, CAN or ISA I/O can be used for the I/O connections.
Floating point processor	With Intel processors, using the FPU on the processor has been included to perform calculations.

AR105 can be installed on an IPC5000 with the B&R Automation Runtime™ Installation Kit. Once it has been successfully installed, AR105 can communicate with B&R Automation Studio™. Possible AR105 upgrades are made using B&R Automation Studio™ in the form of operating system upgrades or installed using the B&R Automation Runtime™ Installation Kit.

#### 1.1.6.10.2 Ordering information

Each copy of B&R Automation Runtime™ AR105 (mod. no. 1A4601.05) is delivered with a runtime license in the form of a security key (hardware dongle), as well as a license sticker.

AR105 operation is only possible on IPC5000 devices if the security key is installed in the target machine or attached to one of the computer's LPT interfaces using a key ring (mod. no. 9A0003.01, not included in delivery).

The B&R Automation Software™ CD (see the "Installation kit" section) is required to install B&R Automation Runtime™ AR105 on the automation target (target computer).

## 1.1.7 Accessories

### 1.1.7.1 Overview

Model number	Description
0G0001.00–090	Cable PC <--> PLC/PW, RS232, online cable
9A0003.01	Dallas Key Ring adapter to use Dallas dongle on a parallel interface

## 1.2 Getting started in B&R Automation Studio

### 1.2.1 Introduction

#### 1.2.1.1 The most important tools

The following table provides general overview of the most important tools in B&R Automation Studio™ and their capabilities.

<b>Tool</b>	<b>Features</b>
Hardware configuration	<p>Hardware configuration features:</p> <ul style="list-style-type: none"> <li>• Automatic recognition of the target system</li> <li>• Graphic display of the target system</li> <li>• Detailed information about the target system</li> <li>• Simple definition of target system profiles</li> <li>• Direct assignment of symbolic variables to physical terminals</li> </ul>
Software configuration	<p>Software configuration features:</p> <ul style="list-style-type: none"> <li>• Efficient configuration of the target system</li> <li>• Clear display of the project hierarchy</li> <li>• Simple definition of software profiles</li> </ul>
Programming languages	<p>B&amp;R Automation Basic™ ..... formerly PL2000</p> <p>C Language ..... ANSI C</p> <p>Ladder Diagram ..... IEC 1131 Ladder Diagram (LD)</p> <p>Sequential Function Chart ..... IEC 1131 Sequential Function Chart (SFC)</p> <p>Instruction List ..... IEC 1131 Instruction List (IL)</p> <p>Structured ..... IEC 1131</p>

	Text                      Structured Text (ST)
Libraries and function blocks	<p>Many standard functions (function blocks) are included in the delivery of B&amp;R Automation Studio™ and are grouped together in various libraries. These library function blocks can save you a lot of time and effort when creating solutions to standard problems.</p>
Visualization	<p>Visual Components in B&amp;R Automation Studio™ makes it possible to create process diagrams and other visualization elements and test them together with the control task. The following tools are integrated:</p> <ul style="list-style-type: none"> <li>• Picture editor</li> <li>• Alarm system</li> </ul>
Technology functions	<p>Modular technology plug-ins allow the uniform integration of high-performance tools.</p> <ul style="list-style-type: none"> <li>• Configuring, analyzing, and diagnosing axis/drive parameters</li> <li>• CNC editor</li> <li>• Cam editor</li> </ul>
Diagnostics, service, etc.	<ul style="list-style-type: none"> <li>• Online variable forcing</li> <li>• Online variable monitoring</li> <li>• Real-time tracing</li> <li>• Complete source level debugger with breakpoints, single step, line coverage, and disassembler functions</li> <li>• Logbook entries for easy system status inquiries</li> </ul>




- Profiler function – measures system and task runtimes

**1.2.1.2 Conventions**

In order to be able to find and interpret information quickly and easily in this manual, the same visual aids, standard text formats, and terms are always used. These conventions are explained in the following.

**1.2.1.2.1 Visual aids**

<b>Convention</b>	<b>Description</b>
<b>CAPITALS</b>	Directory names, file names, etc. are shown in capital letters.
<b>Bold type</b> in small or capital letters	Menu items, command names, names of dialog fields, buttons, and options are shown using a bold typeface.
<b><i>Bold Italic</i></b>	Important terms and information are shown in bold italics.
"In Quotation Marks"	All texts that have to be entered in fields and references to other sections are shown in quotation marks.
<b>SMALL CAPITALS</b>	Names of keys on your keyboard, i.e. CTRL, ESC, etc.
	Important notices are displayed within a red border.
	Important tips are displayed within a blue border.
	Explanations of new terms are shown using this symbol.

### 1.2.1.2.2 Key Notations

Key combinations and sequences are differentiated as follows:

<b>Shortcut keys</b>	<b>Description</b>
KEY1+KEY2	A plus (+) between key names means that the first key is held down while the second is pressed.
KEY1, KEY2	A comma (,) between key names means that the keys are pressed and released one after another.

### 1.2.1.3 Features

B&R Automation Studio™ provides you with a system for developing all different types of automation tasks. Features of B&R Automation Studio™:

<b>Feature</b>	<b>Description</b>
Microsoft Windows Look & Feel	The B&R Automation Studio™ desktop complies with the Microsoft design guidelines for Windows programs. This makes it easier to become familiar with the program and reduces project engineering costs.
Supports many target systems	B&R Automation Studio™ is a development environment for many different target systems. This allows the automation platform to be scaled to meet your exact requirements.
More than IEC 1131	In addition to IEC 1131 languages, B&R Automation Studio™ offers high-level programming using ANSI C as well as a complete line of high-performance technology functions that considerably simplify project engineering for your machines and systems.
Commissioning	All diagnostic and service tools clearly show system behavior using graphic displays. Synchronous recording of data replaces an oscilloscope and allows exact settings to be made for machine parameters.
Contains all information	B&R has completely integrated all the product information which is relevant during the creation

and development of your B&R Automation Studio™ projects. Standard help functions provide easy access to all details related to the operation of this tool. Hardware–specific data is placed under user–friendly tabs.

## 1.2.2 Software installation

### 1.2.2.1 Automation Net system requirements

The computer on which B&R Automation Net™ is to be installed must meet the following requirements:

#### 1.2.2.1.1 PVI Runtime

	Recommended
Operating system	Windows 95/98/ME, Windows NT 4.0, Windows 2000, Windows XP
Software	In Windows 95, at least Windows socket (Winsock) 2.2 is needed
Processor	At least Pentium 166 MHz
RAM	At least 32 MB (depending on the number of process objects)
Hard disk space	At least 10 MB free

#### 1.2.2.1.2 PVI Runtime & Server

	Recommended
Operating system	Windows 95/98/ME, Windows NT 4.0 (SP3 for OPC Server), Windows 2000, Windows XP
Processor	At least Pentium 166 MHz
RAM	At least 64 MB (depending on the number of process objects)
Hard disk space	At least 100 MB free
Software	At least one Windows socket (Winsock) 2.2 is needed in Windows 95 Windows 95 with DCOM (OPC Server) VB 6.0 (PVI Control) MDAC (OPC Server)

Your computer also has to have an online interface for the connection between your programming device (computer) and the target system (controller). An open interface (COM1 – COM4) or a CAN interface can be used.

### 1.2.2.2 Automation Studio system requirements

The computer on which B&R Automation Studio™ is to be installed must meet the following requirements:

#### 1.2.2.2.1 Hardware

	Recommended
Operating system	Windows XP
Processor	At least Pentium II 400 MHz
RAM	At least 256 MB
Hard disk space	At least 500 MB free
Graphics	XGA (1024 x 768)

#### 1.2.2.2.2 Software

	Requirements
Internet Explorer	≥ 6.0 (on the "AS Tools" CD)
.NET Framework	V1.1 (on the "AS Tools" CD in the "DOTNET_FW" directory)

Your computer also has to have an open online interface for the connection between your programming device (computer) and the target system (controller). An open interface (COM1 – COM4) or a CAN interface can be used.

### 1.2.2.3 Installing B&R Automation Net

**Note:**

You need administrator rights to install software with the Windows NT/2000/XP operating system! Contact your network administrator if necessary.

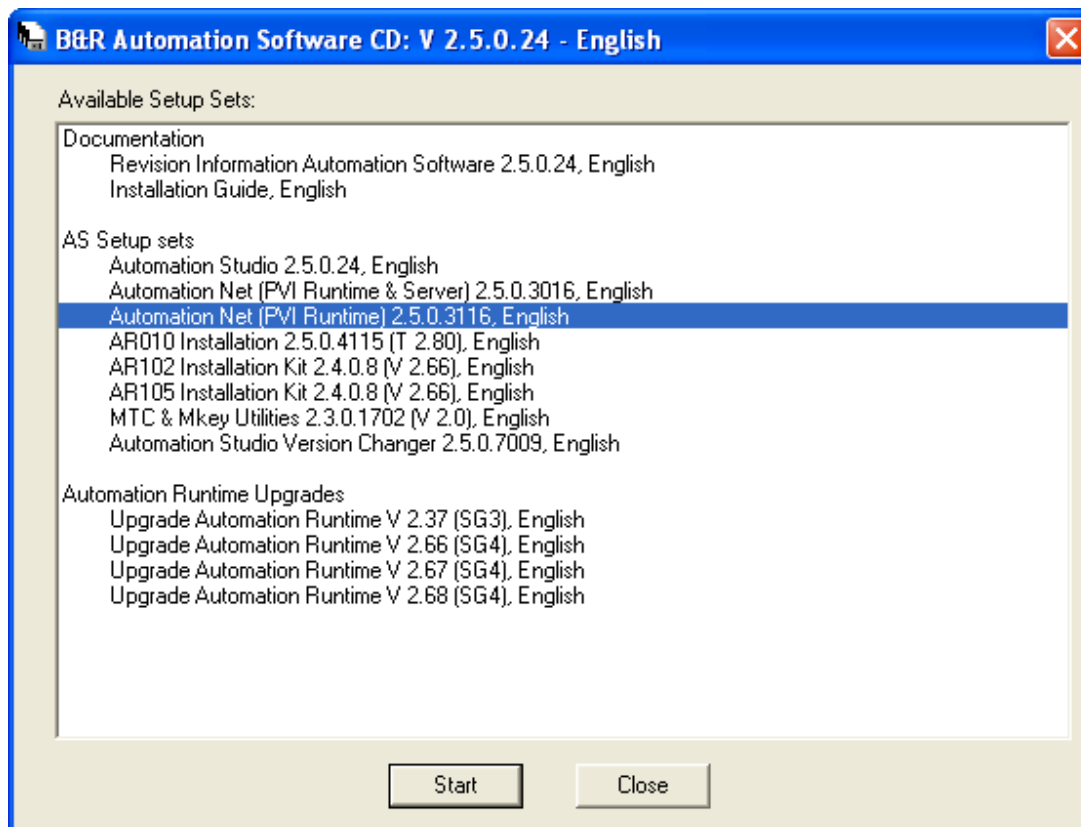
1. Start the operating system.
2. Insert the language-specific installation CD in the computer's CD-ROM drive.
3. If the "AutoPlay" option is activated on your computer, then the B&R Automation Installation window appears. If "AutoPlay" is not enabled, run the **BrMenu.exe** file from the root directory of your installation CD.

**Note:**

Automation Net™ represents the foundation for communication and must therefore be the first component installed on your computer.

If a current version of Automation Net has already been installed on your system, this procedure can be skipped.

To install the minimum version, select the "Automation Net (PVI Runtime)" entry in the "B&R Automation Installation" dialog box (BrMenu.exe). If you need PVI Controls, PVI Transfer, and various servers or examples, select "Automation Net (PVI Runtime & Server)".



Installation begins by pressing the **Start** button.

B&R Automation Studio can then be installed once this procedure has been carried out. This is installed to its own version-specific directory tree.

### 1.2.2.4 Installing Microsoft Internet Explorer

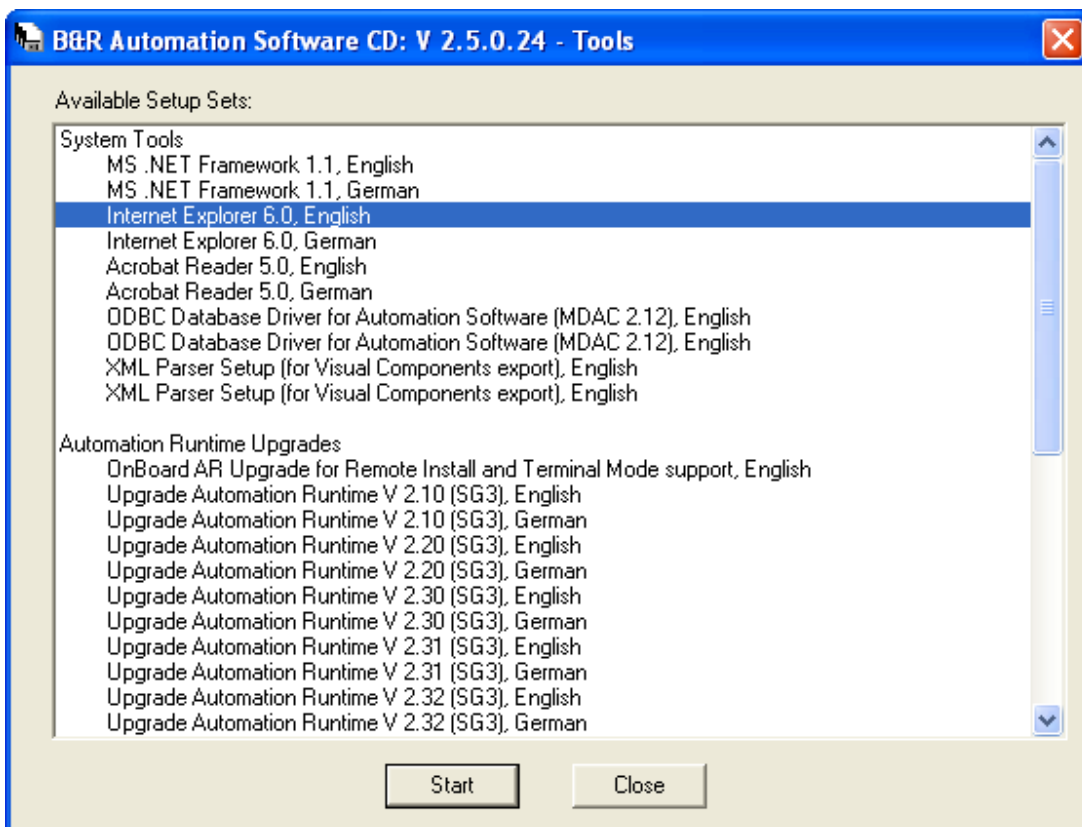


**Note:**

If Microsoft Internet Explorer version 6.0 (or higher) is already installed on your computer, then you can skip the following steps and proceed instead to the [Automation Software](#) installation.

You need Administrator rights to install software with the Windows NT/2000/XP operating system! Contact your network administrator if necessary.

1. Start the operating system.
2. Insert the language-specific installation CD in the computer's CD-ROM drive.
3. If the "AutoPlay" option is activated on your computer, then the B&R Automation Installation window appears. If "AutoPlay" is not enabled, run the BrMenu.exe file from the root directory of your installation CD.



Choose the desired language for Internet Explorer with the cursor keys and then begin the installation by pressing the **Start** button. The entire installation is menu driven. If the installation was successful, restart your computer so that all settings can be applied.

### 1.2.2.5 Installing B&R Automation Software

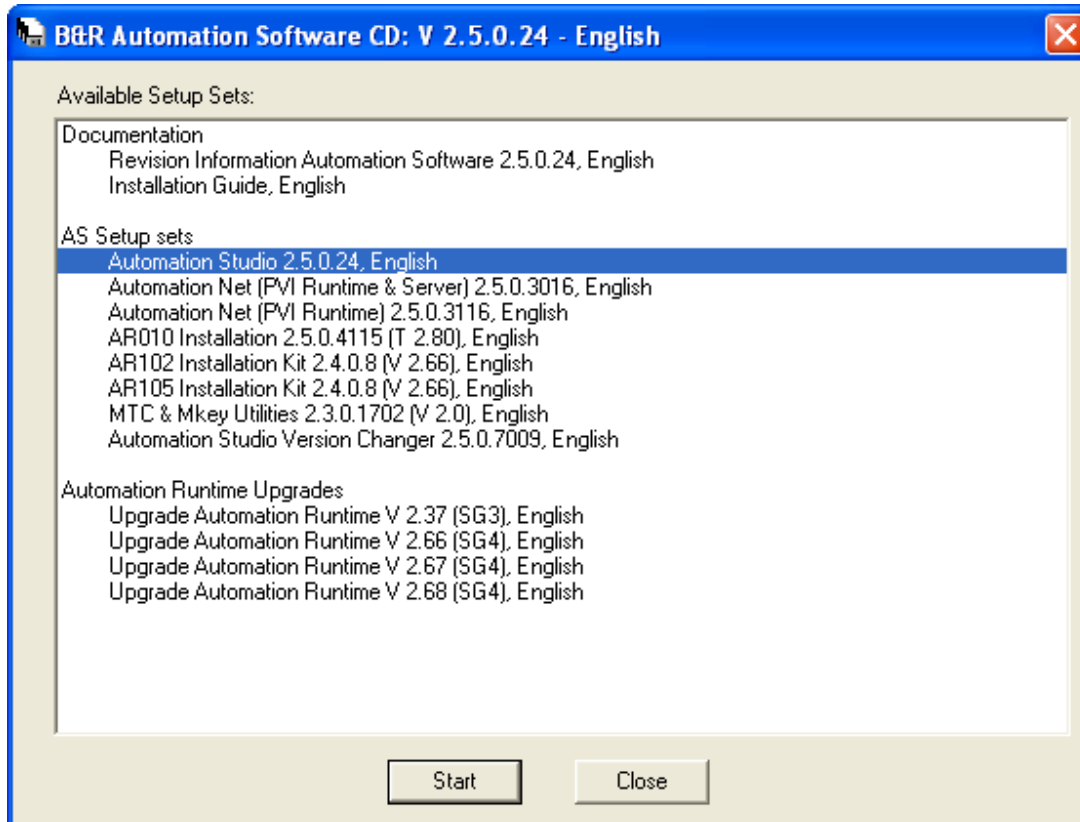


**Note:**

You need Administrator rights to install software with the Windows NT operating system! Contact your network administrator if necessary.

To install Automation Studio, you must first install [Automation Net eV2.3.0.1](#), [Internet Explorer eV6.0](#), and [.NET Framework eV1.1](#). If these are not installed, Automation Studio installation is cancelled with a message indicating the versions/software necessary.

1. Start the operating system.
2. Insert the language-specific installation CD in the computer's CD-ROM drive.
3. If the "AutoPlay" option is activated on your computer, then the B&R Automation Installation window appears. If "AutoPlay" is not enabled, run the **BrMenu.exe** file from the root directory of your installation CD.

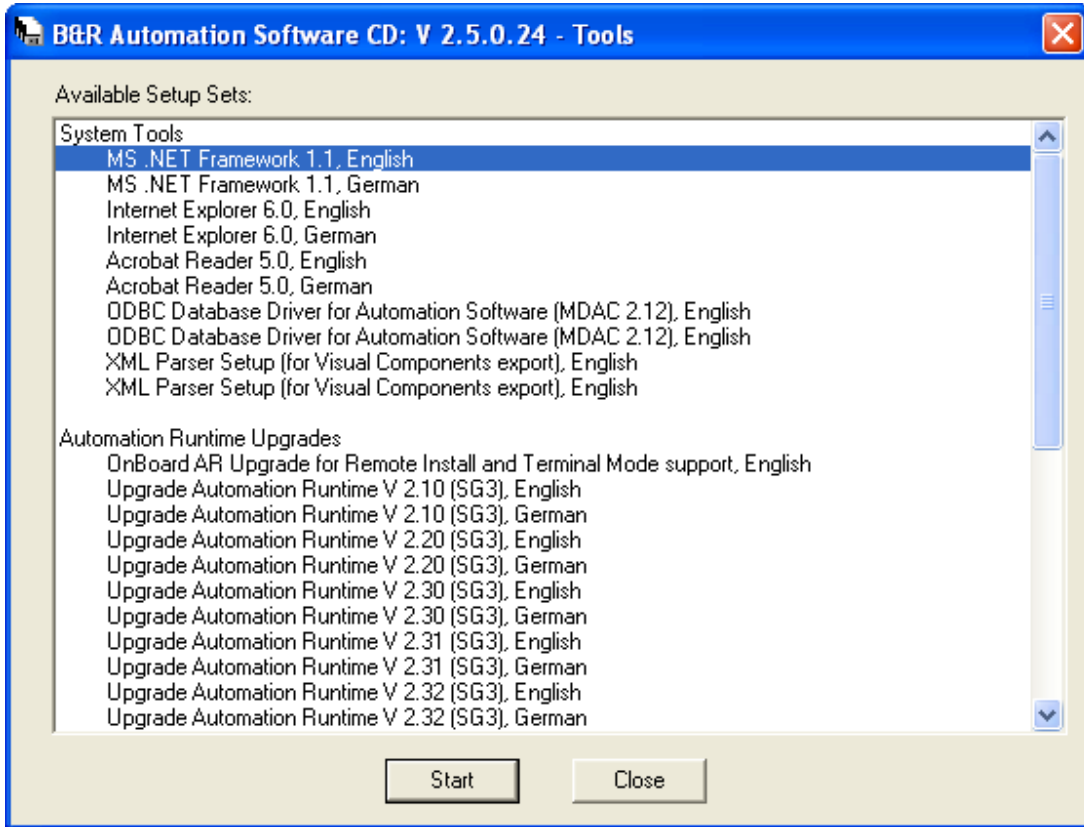


Choose the desired language for Internet Explorer with the cursor keys and then begin the installation by pressing the **Start** button.

#### 1.2.2.5.1 Installing .NET Framework

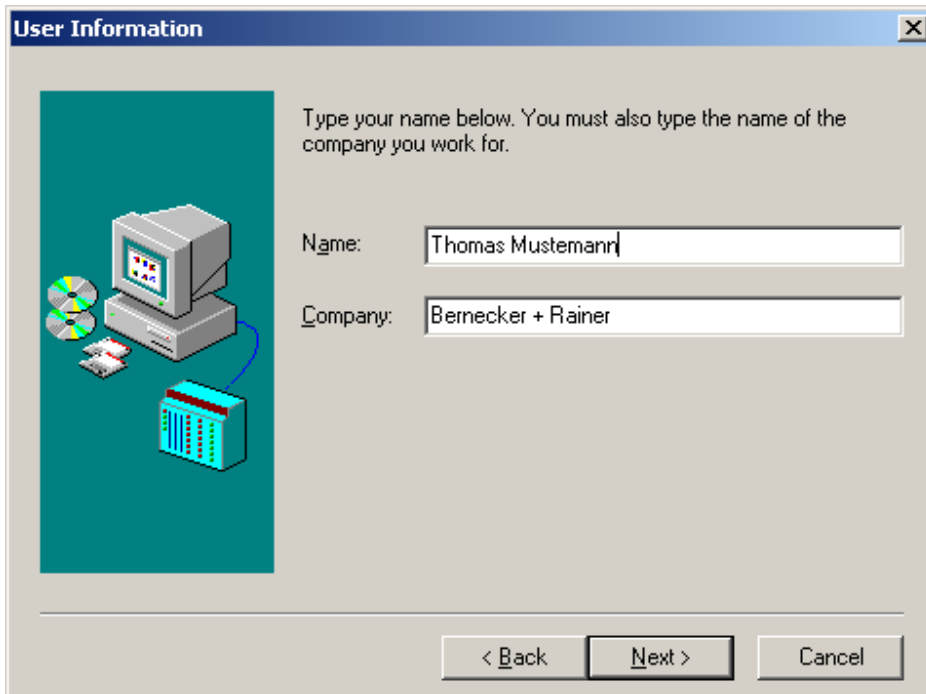
If .NET Framework isn't installed on your system yet, it can be done as follows:

1. Place the "AS Tools" installation CD in the CD-ROM drive on your computer.
2. If the "AutoPlay" option is activated on your computer, then the B&R Automation Installation window appears. If "AutoPlay" is not enabled, run the BrMenu.exe file from the root directory of your installation CD.



Choose the desired language for .NET Framework with the cursor keys and then begin the installation by pressing the **Start** button. The entire installation is menu driven. Installing Automation Software can continue once the installation is completed successfully.

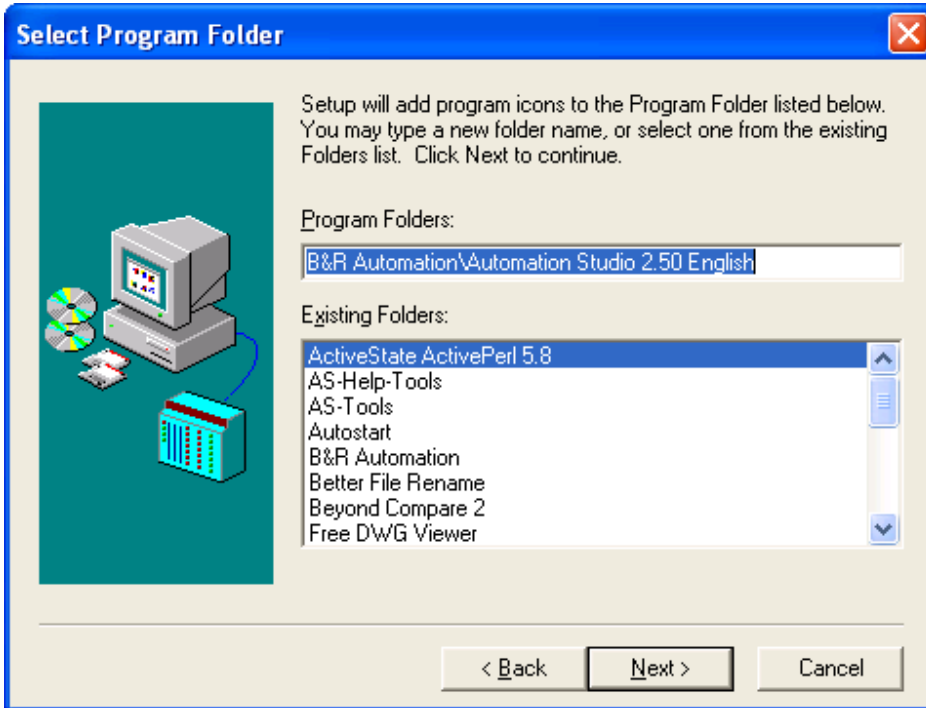
### 1.2.2.6 Entering user data



Enter the user and company name and click on **Next >** to confirm the information.

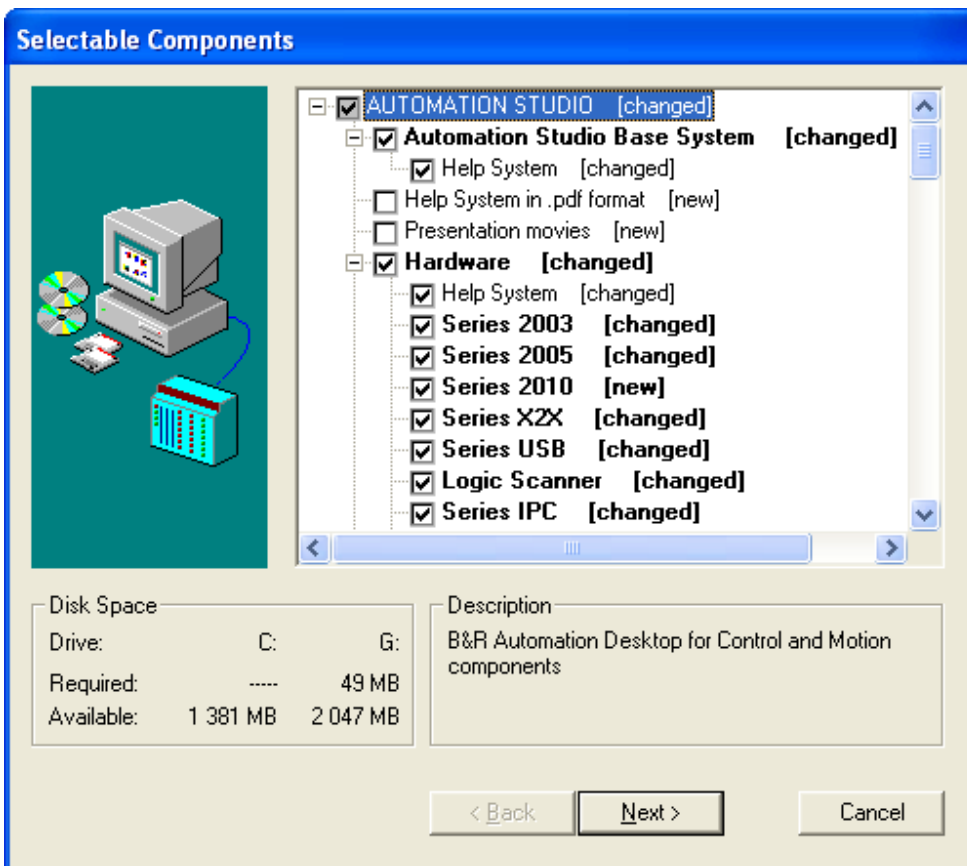
### 1.2.2.7 Setting the program folder





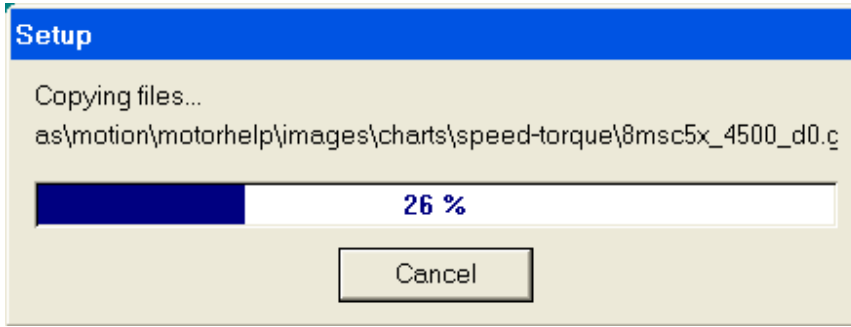
The program is copied to the assigned folder when the **Next >** button is pressed. If the target folder should be changed, click on the **Browse** button to set a different location.

### 1.2.2.8 Selecting software components



Select the software components to be installed and click on the **Next >**.

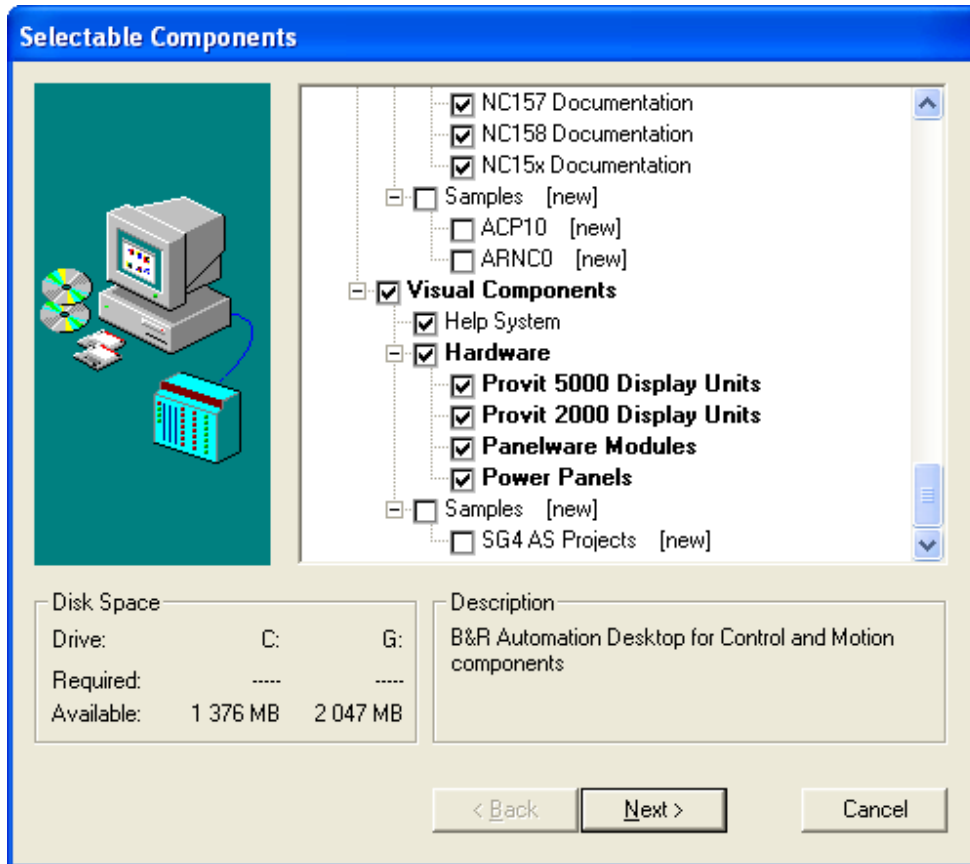
The archive file is now unpacked and copying begins.



Progress for the copying procedure is displayed on the screen.

### 1.2.2.9 Adding or installing software components

If the Setup program is started on a system which already has Automation Studio installed, all of the components which can be installed are shown in a component tree. Components which have not already been installed on the system are identified with **[new]**.



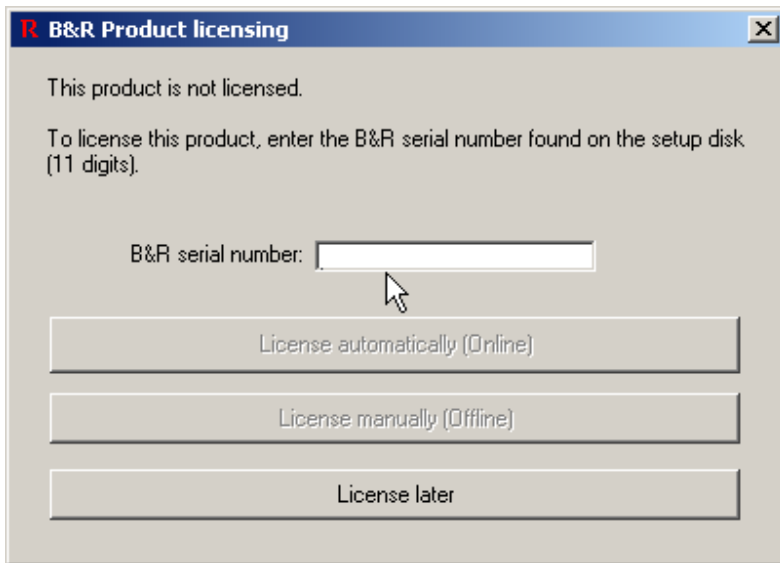
You can now select those components that should be installed to your system.

**Warning:**

If check marks are removed from the component tree, it means that you would like to automatically **uninstall** those components.

### 1.2.2.10 Licensing

The following dialog box is displayed as long as B&R Automation Studio™ has not yet been registered:



Enter the B&R serial number (included on the CD insert) into the first input field.



Once a valid serial number is entered, all buttons become active.

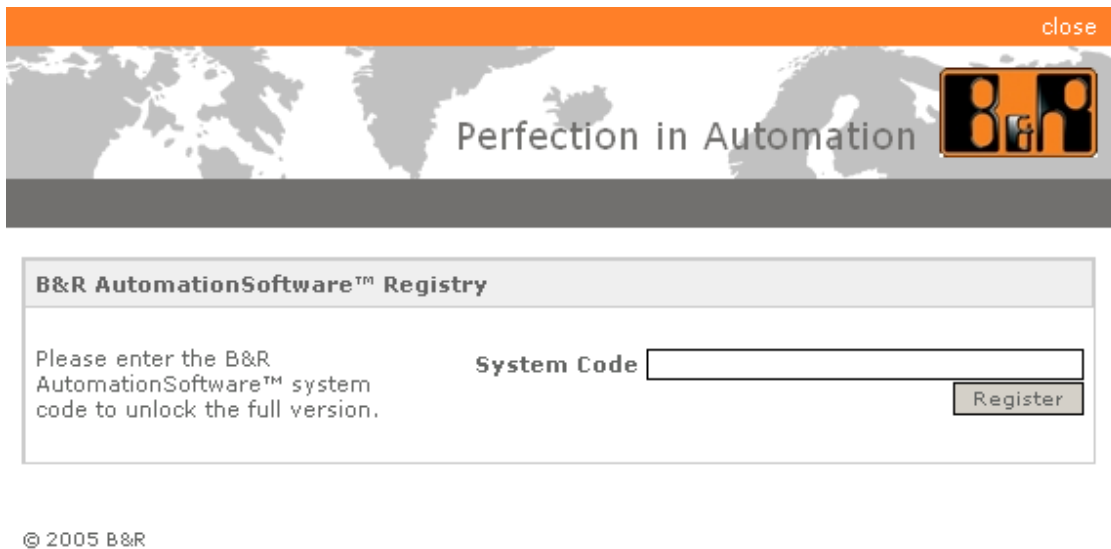
**1.2.2.10.1 License automatically (online)**

Pressing the **License automatically (online)** button automatically licenses the B&R Automation Studio™ Version online if there is an existing Internet connection.

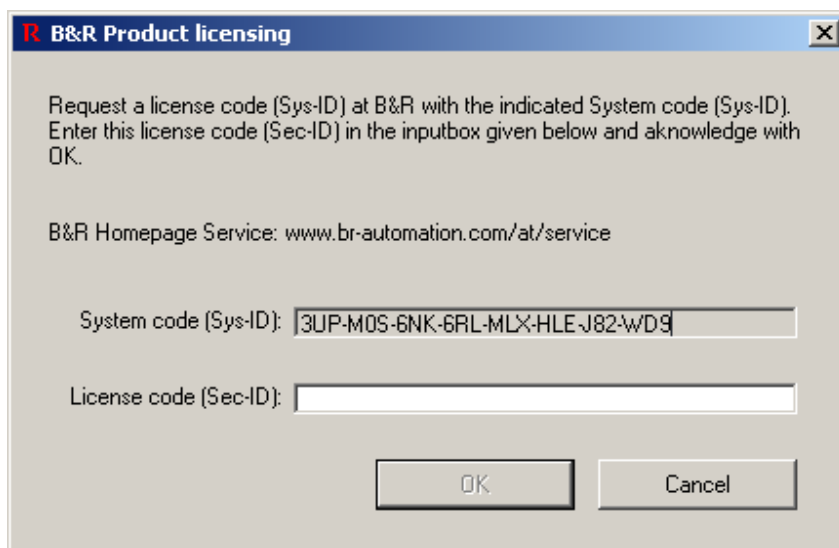
An error message is output if a connection cannot be established to the B&R server.

**1.2.2.10.2 License manually (offline)**

Pressing the **License manually (offline)** button opens up a licensing dialog box. You can use the system code (Sys-ID) shown here to apply for the actual activation code from the B&R Homepage Services site.



Entering the system code (Sys-ID) and pressing the **Register** button displays the activation code, which must then be entered into the lower input field in the Automation Studio licensing window (License code (Sec-ID)).



If the correct code has been entered, then the licensing procedure can be completed with the **OK** button.

If a valid B&R serial number is not input, the dialog box can still be closed with **License later**, which starts the 30 day evaluation period.

The time left in the evaluation period is shown each time Automation Studio™ is started. Operation is possible during this period without limitations.

If the evaluation period expires, then the software must be licensed to be able to continue using it.

### 1.2.2.11 Uninstalling B&R Automation Software

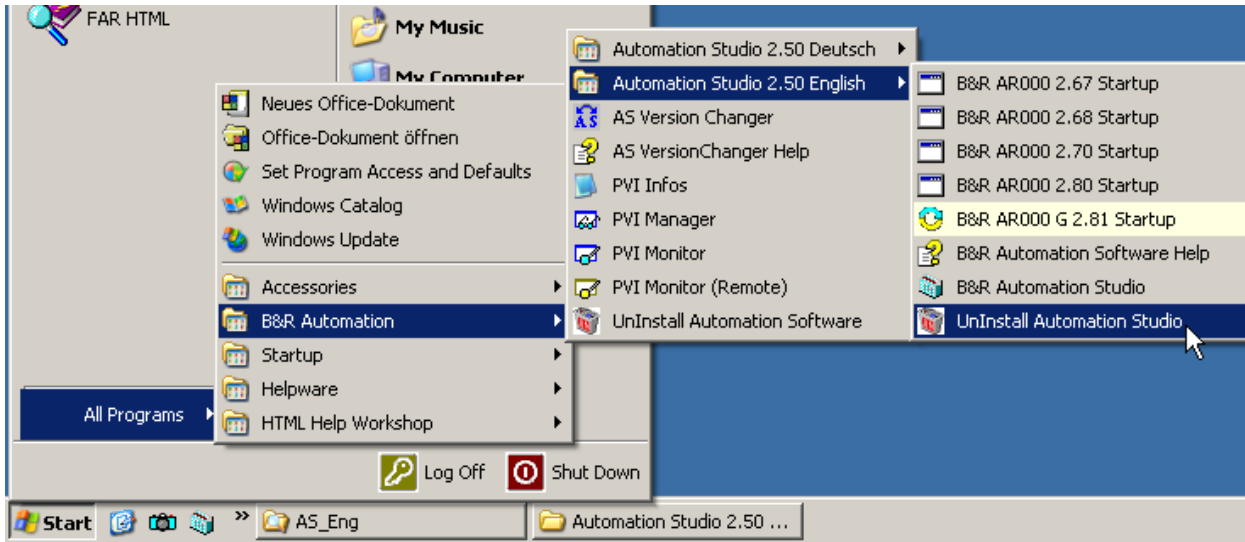


**Note:**

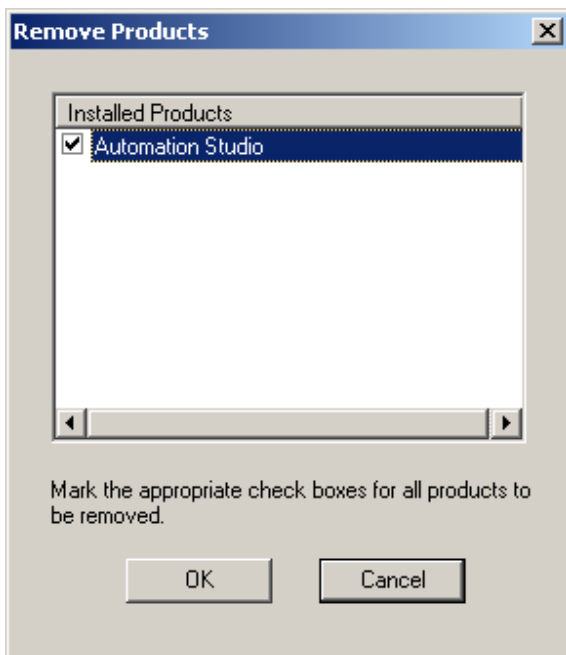
If there are several version of B&R Automation Studio™ being used on your computer simultaneously, only the **active** version can be uninstalled.

If no versions are active, the Version Changer can be used to activate the one you want to uninstall (see Version Changer help).

If you want to remove B&R Automation Software™ from your hard drive, proceed as follows:



If the Uninstall Wizard is started, then you have the possibility to choose which program(s) to remove in the next dialog box.



However, if the uninstall finished with an error message, then the removal of any remaining unused files has to be done manually.



**Applies to multiple installations:**

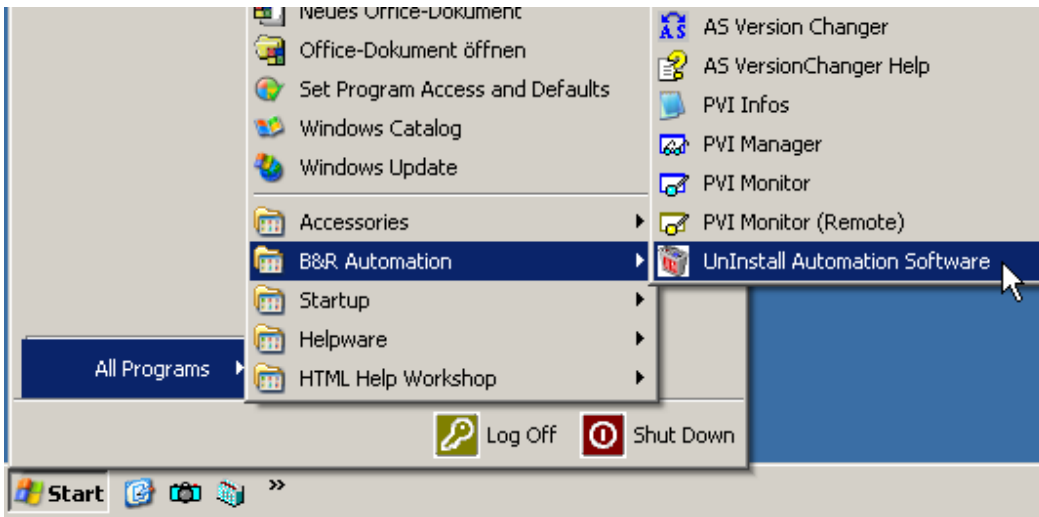
If there is still a B&R Automation Studio™ version present on your computer after one has been uninstalled, it has to be activated using the Version Changer.

**1.2.2.11.1 Uninstalling B&R Automation Net™ (PVI)**

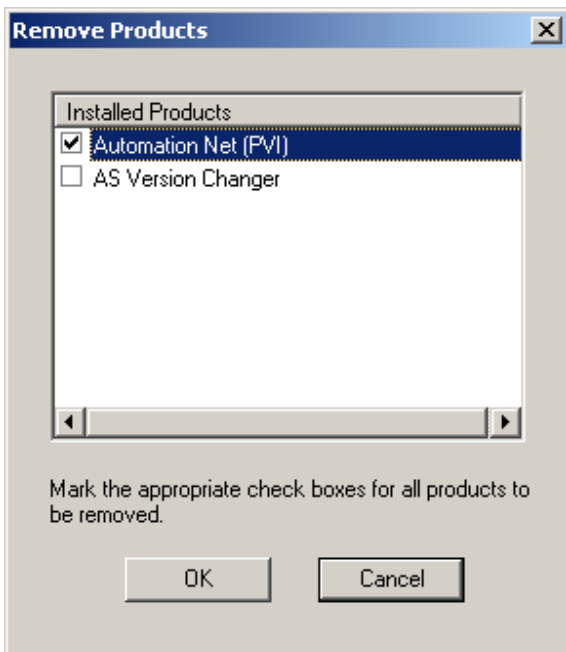


Before B&R Automation Net™ (PVI) can be uninstalled, all B&R Automation Studio™ versions need to be uninstalled beforehand.

If you want to remove B&R Automation Net™ from your hard drive, proceed as follows:



If the Uninstall Wizard is started, then you have the possibility to choose which program(s) to remove in the next dialog box.



Uninstallation of the selected components begins when the **OK** button is clicked.

### 1.2.2.12 Installing using a response file ("silent installation")

#### 1.2.2.12.1 General information

Components of the B&R Automation Software can also be installed on the target system by means of an automatic installation (response file installation).

The installation is then achieved without user input with information from a response file. This file must first be created with a "sample installation".

A normal installation (without a response file) gets the required information from the user in the form of dialog box responses. A response file installation doesn't require any user input. Here the installation gets its user entries from another source – the response file (.iss file).

A response file contains information similar to that which the user enters in dialog boxes during a normal installation. When run, the installation program reads the necessary information from the response file.

#### 1.2.2.12.2 Recording a response file

Carries out a "sample" installation. The "**Setup.exe**" program is started with the "**r**" parameter. All of the installation selection data and the settings in the file "Setup.iss" (in the Windows directory by default) are recorded.

#### 1.2.2.12.3 Response file installation

After the response file has been created using a "sample installation", it can be used for the automated installation. During the installation, there is no output on the screen; instead, information about the installation is recorded in the Setup.log file. It also records whether the installation has been successful or not ("ResultCode" value in the [ResponseResult] part of the log file).

In order to carry out the response file installation, run the **Setup.exe** program with the **-s** option.

The name and path of the response file and the log file can be set using the command line parameters **-f1** and **-f2**.

Examples see [SETUP usage examples](#).

##### 1.2.2.12.3.1 SETUP usage examples

Example	Remark
setup r	Starts the installation program for the sample installation for creating the response file in the Windows directory
setup s	Starts the installation program in response file mode and attempts to load Setup.iss from the same directory that also contains Setup.exe. The Setup.log file is also produced in this directory if it is not write-protected.  If the directory is write-protected, then the Setup.log file will be stored in the Windows directory (if no log file path (-f2) has been specified).  If the log file path is specified (-f2 parameter), then it will be placed there.
setup -s -f1c:\mydir\mydir.iss	Starts the installation program in response file mode and attempts to load mydir.iss from the c:\mydir directory for use as a response file. The Setup.log file is placed in the same directory as the response file (c:\mydir).

##### 1.2.2.12.3.1.1 Some Tips for Using "Setup.exe"

- No spaces are allowed between command line switches and options.
- If response file installation is carried out, a log file is created in the same directory as the response file. The response file has the default name Setup.iss if switch -f2 is not specified together with -f1.



- If the -f1 switch is not used during a response file installation, the Setup.iss file is searched for in the same directory as Setup.exe. The log file is also created in this directory.
- Command line switches and options are not case sensitive.

**1.2.2.12.4 "Setup.exe" parameters**

Setup.exe is the start program for creating an installation set with InstallShield. A few command line parameters are provided for a response file installation. Parameters are not case sensitive.

Parameters for installing using a response file:

Parameters	Usage	Meaning
-f1<path\response_file>	Optional	The path and name of an alternative response file (.iss file) is specified with this option. If this option is used with a response file installation (silent installation), then the response file with the specified path/name <path\response_file> is used.
-f2<path\log_file>	Optional	This parameter is used to specify an alternative path and name for the log file created by the response file installation  By default, the Setup.log file will be created in the same directory as the Setup.exe file as long as the directory is not write-protected.  If the directory is write-protected, then the Setup.log file will be stored in the Windows directory (if no log file path (-f2) has been specified).  If the log file path is specified (-f2 parameter), then it will be placed there.
-r	Required entry	This parameter causes Setup.exe to automatically generate an installation response file (.iss file) for recording all input and selection settings. This file is stored in the <Windows> directory.
-s	Required entry	Executes InstallShield using a response file (no user input).

**1.2.2.12.5 Setup.log file**

Since there is no output on the screen during installation, installation information is recorded in the Setup.log file.

The log file contains three sections:

- [InstallShield Silent](#)
- [Application](#)
- [ResponseResult](#)

**1.2.2.12.5.1 [InstallShield Silent] section**

The version of InstallShield Silent is entered in this section. It also marks this file as a log file.

**1.2.2.12.5.2 [Application] section**

The name and version of the installed system as well as the company name are entered in this section.

**1.2.2.12.5.3 [ResponseResult] section**

This section contains the status code of the installation and indicates whether the response file installation has been successful. The numeric value of the "ResultCode" entry in the [ResponseResult] section provides

information about the installation execution.

Possible values for "ResultCode":

0	Installation successful
-1	General error
-3	Necessary data not found in the "Setup.iss" file
-4	Insufficient memory available
-5	File does not exist
-6	Not possible to write to response file
-9	Invalid list type (string or numeric)
-10	Invalid data type
-11	Unknown error during installation
-12	Dialog boxes are not in the correct order

Setup.log is the standard name for the [response file](#). By default, this is placed in the same directory as the Setup.ins file. The name and path of these log files can be changed using command line parameters (-f1, -f2) with Setup.exe

The content of the Setup.log file after a successful response file installation is as follows:

```
[InstallShield Silent]
Version=v3.00.000
File=Log File
```

```
[Application]
Name=BrAutSetup
Version=0.00
Company=BR_Automation
```

```
[ResponseResult]
ResultCode=0
```

### 1.2.2.13 AR installation disks

## 1.2.3 Getting started

### 1.2.3.1 Starting B&R Automation Studio

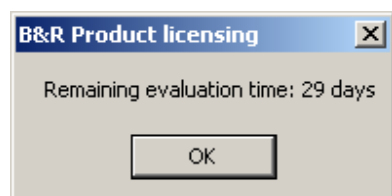
B&R Automation Studio™ is started as follows:

- Click on the Start button and move the mouse pointer over "Programs".
- Then move the mouse pointer over the B&R Automation group and click on the the B&R Automation Studio program.

After starting B&R Automation Studio™, the start screen is shown for a short time before B&R Automation Studio™ is opened.

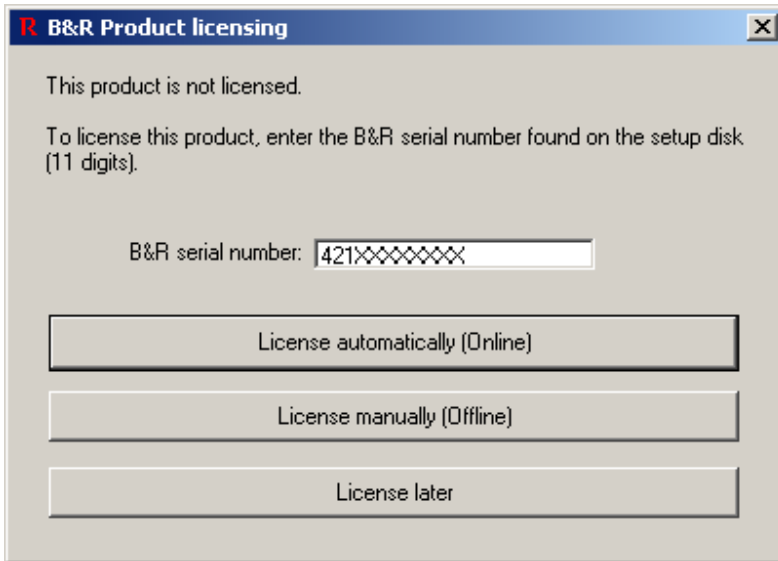
#### 1.2.3.1.1 Starting the evaluation version

If B&R Automation Studio™ has not been licensed, it can only be used for 30 days. The remaining evaluation period is shown each time B&R Automation Studios™ is started. Operation is possible during this period without limitations.



The licensing dialog box is opened by clicking on the **OK** button. You can either license the software or open the evaluation version by clicking on the **License later**.

After 30 days have passed, B&R Automation Studio™ can no longer be started. "Evaluation ended" is shown. The licensing dialog box is opened by clicking on the **OK** button.

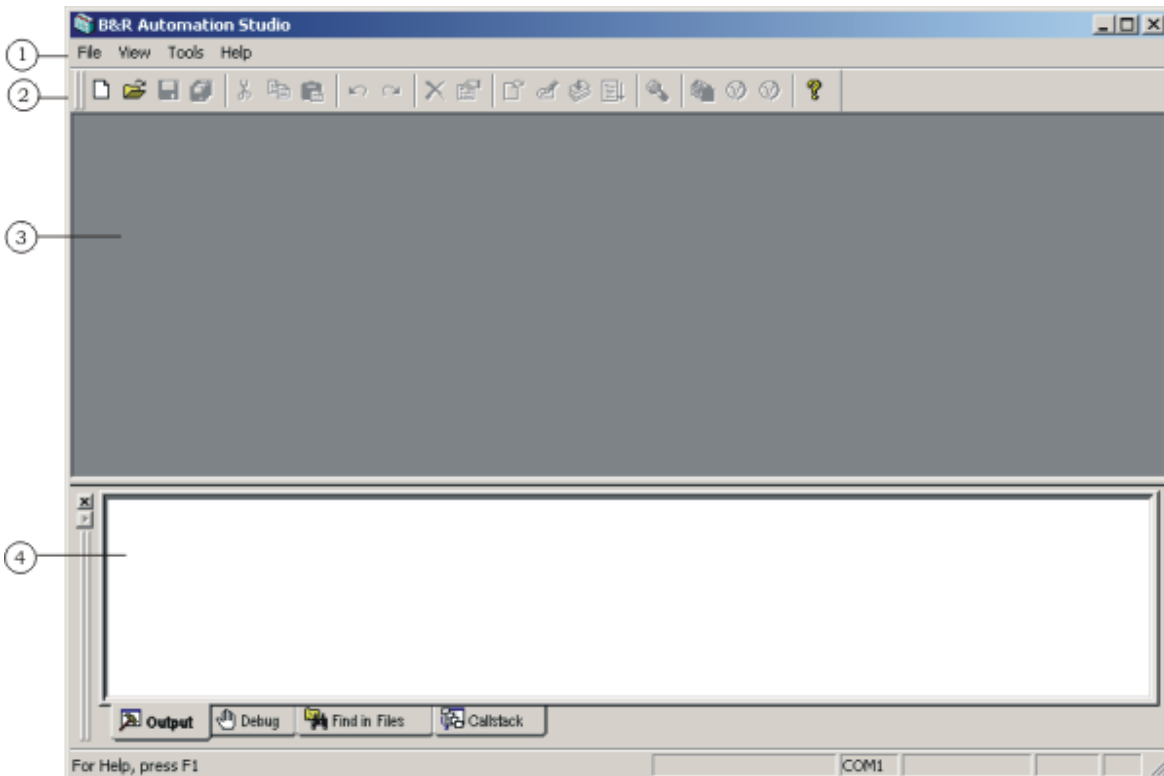


Clicking on the **License automatically (online)** button automatically carries out the automatic licensing process if there is an existing Internet connection.

More information about this can be found in the online help.

**1.2.3.1.2 B&R Automation Studio**

Layout of the B&R Automation Studio™ window:



No.	Name	Description
-----	------	-------------

1	Main menu	The B&R Automation Studio main menu adapts itself to the active editor. In this way, only functions which are necessary for the current editor are available.
2	Toolbar	Contain icons that provide fast access to several different commands and functions. Placing the mouse pointer over an icon displays its function. Use the <b>Toolbars...</b> command in the <b>View</b> menu to show or hide toolbars.
3	Program workspace	The window for an open project is shown in this area. The project window can either be maximized to fit into this area or sized accordingly.
4	Message window	The message window is shown in the lower part of the program window after B&R Automation Studio™ is started the first time. It is used to display compiler and debugger messages, etc.
5	Status bar	The status bar at the bottom of the window shows information such as: <ul style="list-style-type: none"> <li>• Brief help about menu commands or toolbar icons</li> <li>• Brief information concerning editing procedures</li> <li>• Current position in a list, tree, or text</li> <li>• Status of the online connection between the programming device and the target system (see the <a href="#">Creating a new project</a> section).</li> <li>• Status of the target system (RUN, STOP, DIAG)</li> <li>• Status of various keys (CapsLock, Num, Scroll)</li> </ul>

#### 1.2.3.1.3 Closing B&R Automation Studio

B&R Automation Studio™ can be closed by...

...Clicking on **Close** in the **File** menu, or

...Clicking on the button in the top right corner of the B&R Automation Studio™ window.

#### 1.2.3.1.4 Keyboard and mouse operation

The B&R Automation Studio™ workspace complies with Microsoft design guidelines for Windows programs. That means keyboard and mouse operation also correspond to the Windows standard.

Additional information concerning the operation of B&R Automation Studio™ can be found in the online help. To start the online help, click on **Help** in the **?** menu or press F1.

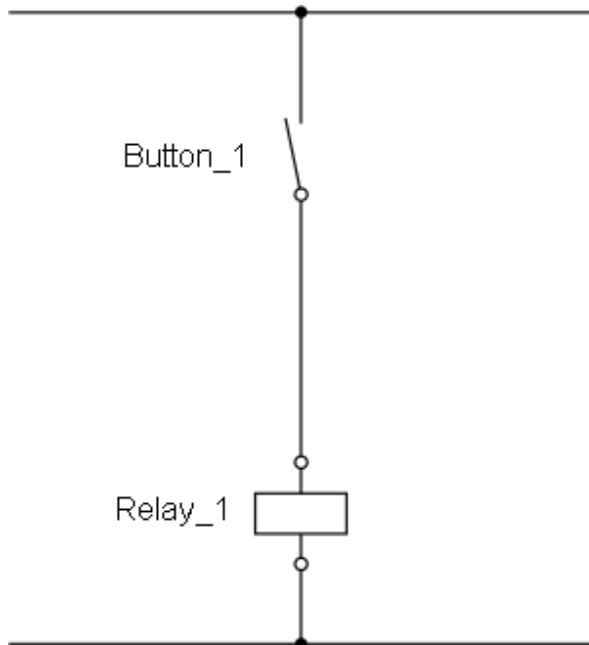
The [Shortcut keys](#) section contains an overview of all key combinations used in B&R Automation Studio™ to directly call commands.

### 1.2.3.2 Example program

In the following sections, we will get the ball rolling by writing a short example program in B&R Automation Studio™ using the ladder diagram (LAD) programming language.

#### 1.2.3.2.1 Task definition

We will create a project called "AS\_QS" in the "C:\ASProjects" directory. Then we will open the project and enter the circuit plan shown in the following illustration using the Ladder Diagram programming language.



This example program be saved in the project "AS\_QS" under the name "logic1".

#### 1.2.3.2.2 Hardware requirements

The following target system hardware is used for all example programs:

Much of the text in the following sections refer to this hardware configuration. However, you can also use different hardware. To run this example program, you will need at least the following hardware components:

- Power supply module
- CPU
- Digital input module
- Digital output module

Without the hardware components listed above, you can program the example but you can't test it.

#### 1.2.3.3 How do I create a project?

A project contains all the information needed by a system/machine, including programs and corresponding databases. In order to be able to write an example program, you first have to create a project.

##### 1.2.3.3.1 Preparation

B&R Automation Studio™ is a hardware-oriented programming system. That means the hardware has to be defined when a project is created. B&R Automation Studio™ can automatically recognize hardware when a project is created. For inexperienced users, this is the simplest way of defining hardware.

The online connection has to be configured properly so that B&R Automation Studio™ can determine the hardware configuration on the target system. Before creating a project, check the connection between the programming device (computer) and target system (controller).

In order to establish an online connection, the B&R Automation Runtime operating system has to be installed on your target system.



B&R Automation Runtime™ is an adjustable, deterministic, real-time multitasking operating system.

#### 1.2.3.3.2 How do I connect the programming device to the PLC?

The connection between the programming device and the controller is established with an online cable (model no. 0G0001.00–090).

You should take the following points into consideration when establishing an online connection using this cable:

- COM1, COM2, COM3, or COM4 must be free on your computer.
- The online cable has two 9-pin DSUB sockets. Serial interfaces on some computers use 25-pin DSUB plugs; you might therefore need an adapter (available from specialized dealers).
- The online cable is only for a connection between two RS232 interfaces. Do not use an adapter to connect the serial interface COMx with an RS485/RS422 interface. A special interface converter must be used for this!

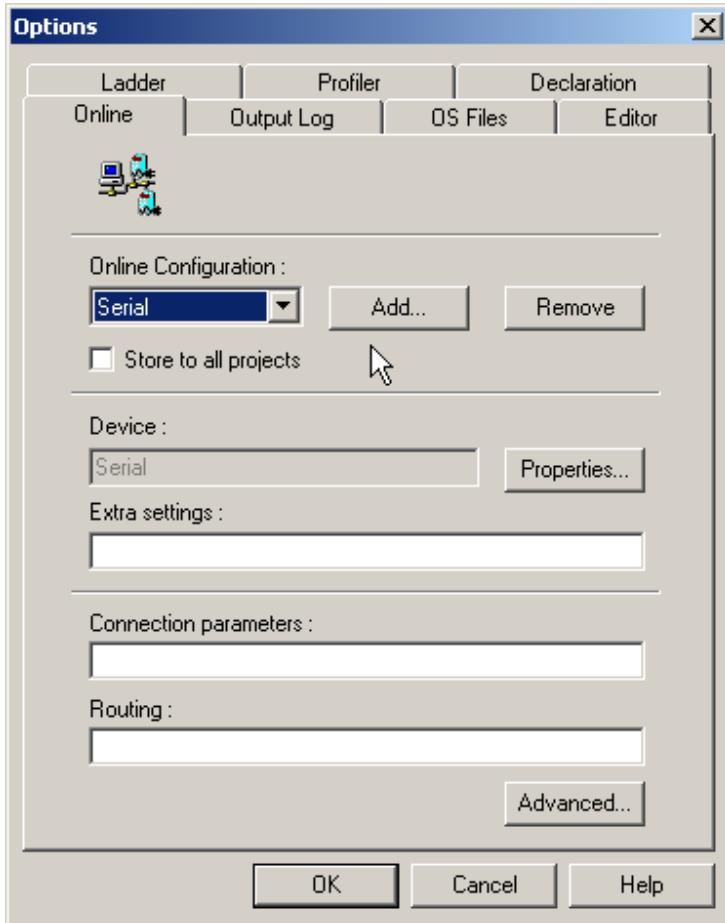
Communication can also take place via an ETHERNET interface. The CAN interface on B&R industrial PCs can also be used as an online interface. When using a LS251 logic scanner CPU, the online connection is made via the PCI bus and SRAM on the LS251.

#### 1.2.3.3.3 Establishing an online connection

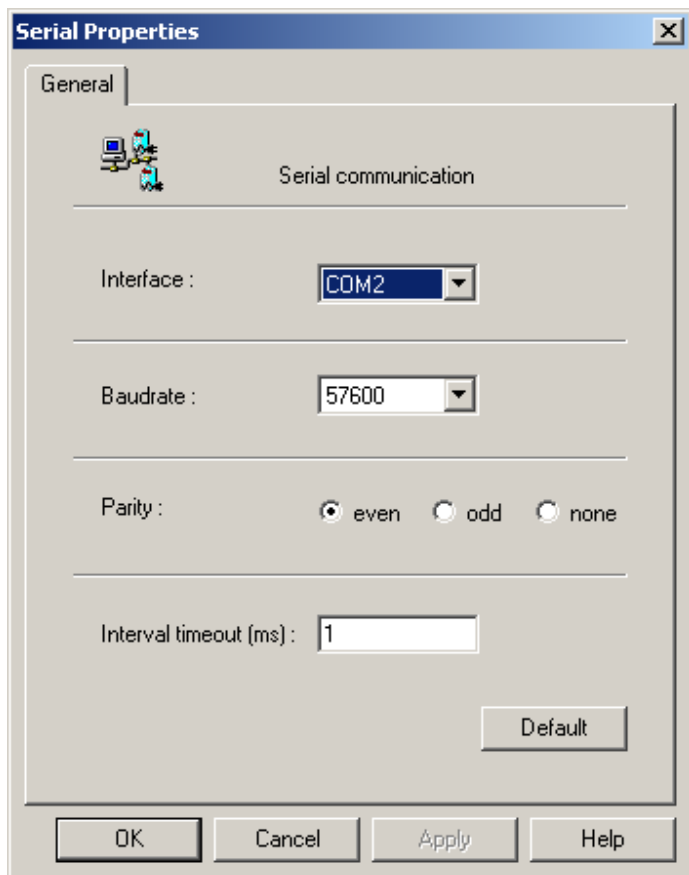
In our example, communication with the controller should take place via the COM2 serial interface on the programming device.

Before the programming device and the controller are connected to each other, make sure that the controller is turned on. Then connect the RS232 interface on the CPU with the COM2 serial interface of the programming device:

The online connection also has to be configured. To do this, select **Options...** from the **Tools** menu in B&R Automation Studio™. The online interface can be configured in the dialog box shown.



After installing B&R Automation Studio™, the serial interface COM2 is selected as default. Then click on the **Properties** button to modify the interface configuration. Settings can be changed in the Serial Properties dialog box. For our example, the default settings do not need to be changed.



Descriptions for other settings in the Options dialog box can be found in the online help.

The defined parameters are applied when this dialog box is closed with the **OK** button.

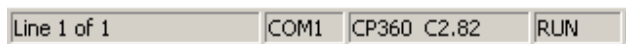
If the online interface is configured correctly, the online connection between the programming device and the controller will be established automatically when the project is opened (see [Creating a new project](#)).

#### 1.2.3.3.4 Connection status

If the online interface is configured correctly, the online connection between the programming device and the target system is established automatically when a project is opened. The connection status is shown in the status bar:

##### 1.2.3.3.4.1 Active connection

Active connection between the programming device and the target:



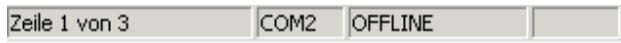
The text in the status bar contains the following information:

Status bar	Information
COM1	Shows the interface used on the computer as the online interface.
CP360 C2.82	Shows the processor type (name of the CPU) and the operating system version used.
RUN	Mode that the processor is in: RUN = RUN mode, SERV = SERVICE mode, DIAG = DIAGNOSTICS mode



## 1.2.3.3.4.2 Offline

No connection between the programming device and the target system:

**Important:**

If a connection cannot be established even though the interface configuration is correct, recheck whether the operating system is installed on the system or whether the interface cable is connected to the correct interface! Refer to the [Preperation](#) section.

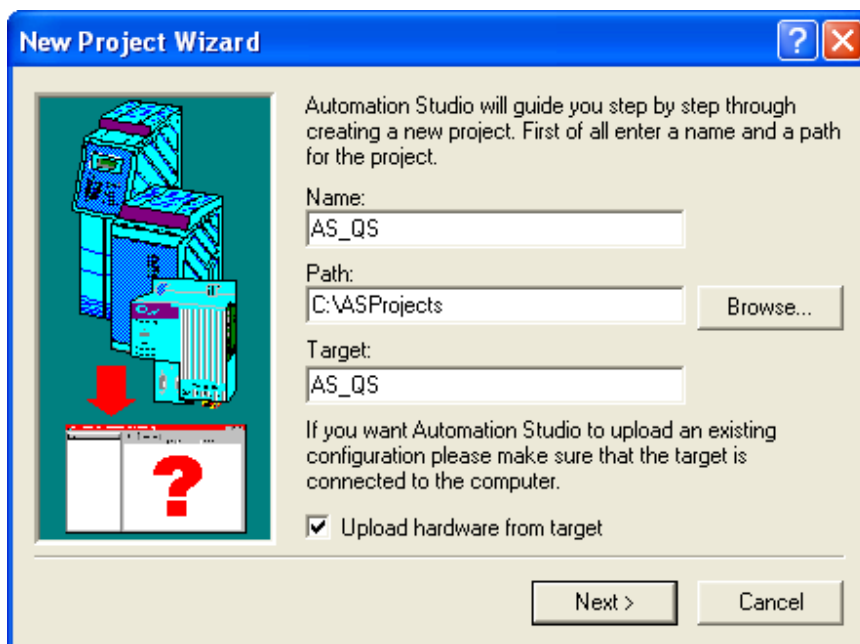
## 1.2.3.3.5 Creating a new project

B&R Automation Studio™ is a hardware-oriented programming system. That means the hardware has to be defined when creating a project. We differentiate between automatic hardware recognition and manual hardware definition.

## 1.2.3.3.5.1 Automatic hardware recognition

B&R Automation Studio™ can carry out automatic hardware recognition when creating a project. For inexperienced users, this is the simplest way of defining hardware.

To create a new project, select the **New Project...** command from the **File** menu. This will open the New Project Wizard. In this dialog box, you can enter the desired project name (Name) and the path to the project directory (Path). For our example project, we will use the project name "AS\_QS", and the project should be saved in the "C:\ASProjects" directory:



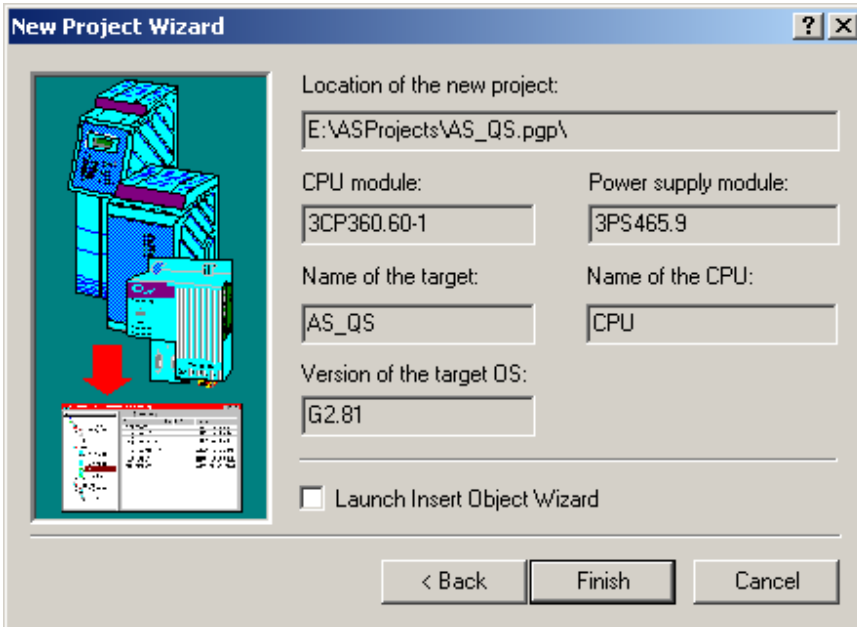
If you want to use a project name other than "AS\_QS", read the limitations for project names given in the context-sensitive help!

**Important:**

In order for the hardware configuration to be uploaded from othe controller automatically, the **Upload Hardware from Target** box needs to be checked.

**If the directory "C:\ASProjects" doesn't exist, you will be asked if it should be created when you press the Next > button. Close this message box with OK.**

B&R Automation Studio™ then automatically recognizes your hardware. The progress is shown in a dialog box. Once the hardware has been detected, clicking on the **Next >** dialog box opens up a new dialog box.



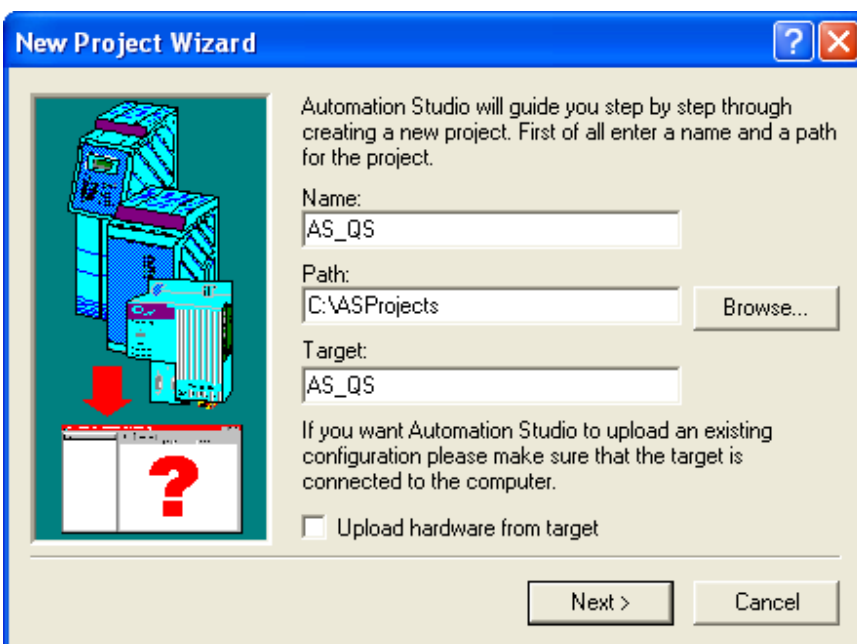
This dialog box shows all previously defined data along with the CPU found during hardware recognition. Power supplies do not have a hardware code, therefore they cannot be automatically recognized by B&R Automation Studio™. They can be added manually later in the hardware configuration (see [How can I work without hardware?](#)).

After checking all entered data, you can use the **Back** button to go back and correct any data if necessary. Once you click on **Finish**, the project is created and opened.

#### 1.2.3.3.5.2 Manual hardware definition

Manual hardware definition is mostly used when the target system is not available. The following section describes the procedure for project creation using manual hardware definition.

To create a new project, select the **New Project...** command from the **File** menu. In this dialog box, you can enter the desired project name (Name) and the path to the project directory (Path). For our example project, we will use the project name "AS\_QS", and the project should be saved in the "C:\ASProjects" directory:



#### Important:

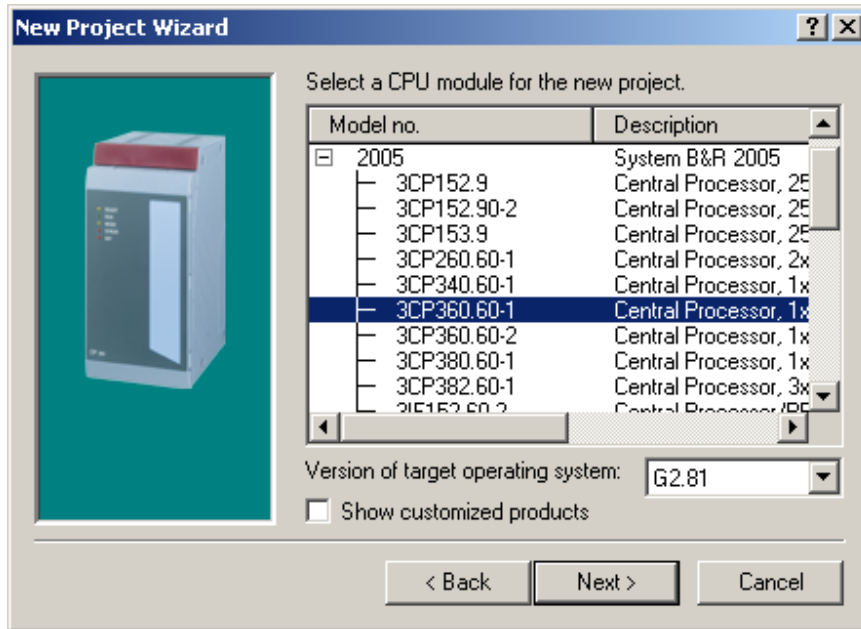
If you want to use a project name other than "AS\_QS", read the limitations for project names given in the

context-sensitive help!

To define hardware manually, the **Upload Hardware from Target** box needs to be unchecked.

If the directory "C:\ASProjects" doesn't exist, you will be asked if it should be created when you press the **Next >** button. Answer this question with **OK**.

The next step is defining the CPU on the target system. A list of all possible modules is shown:



Select the desired CPU from the list and click on **Next >**. Another dialog box is then opened where you can make a selection from a list of all power supply modules that can be used with the selected CPU. Select the desired power supply and then click on **Next >**. The next dialog box shows all previously defined data and the modules defined manually (model number of the CPU and power supply modules):

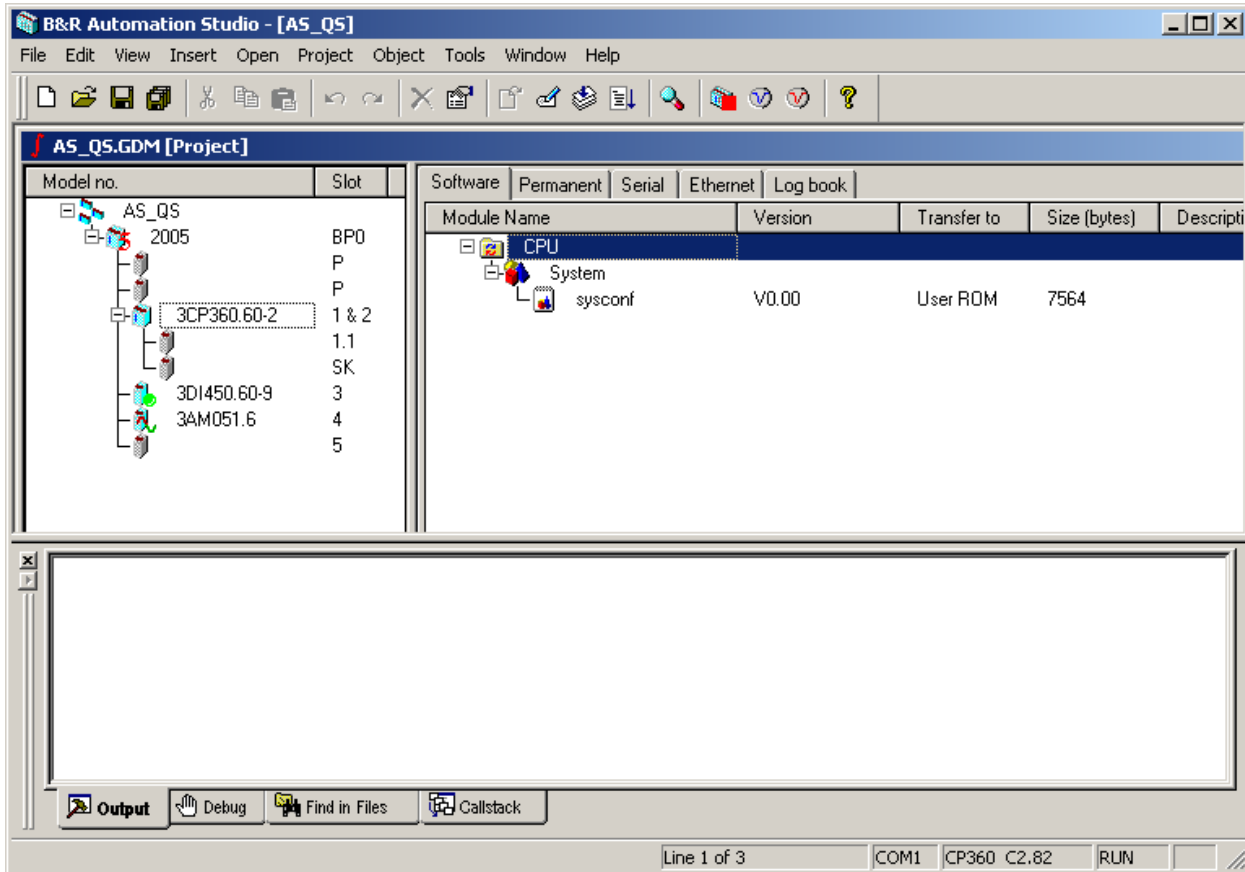
After checking all entered data, you can use the **Back** button to go back and correct any data if necessary. Once you click on **Finish**, the project is created and opened.

**Important:**

To define hardware manually, the **Upload Hardware from Target** box needs to be unchecked.

**1.2.3.3.6 Desktop environment**

A "project window" is opened when a project is opened. The project window is the heart of each project. It is shown as a two part window:



The left side of the window gives an overview of the hardware used in the project (hardware configuration). The right section of the window provides additional information and settings for the marked module.

#### 1.2.3.3.6.1 Left side of the window

The left side of the window gives an overview of the hardware used in the project.





The left side of the project window is also called the hardware configuration.

The easiest way to define hardware in a project is to activate automatic hardware recognition (see [Creating a new project](#)):

- When creating a new project, the user can elect to upload the hardware present from the controller being used.
- If there is an active connection between the programming device and the target system, the current hardware on the target system is compared with the hardware configuration (shown in the project window).

If there are hardware differences, the modules in the current project that are different from the modules on the controller are marked:

Symbol	Description
--------	-------------

	A different module is in this slot on the controller! You can replace the module by marking it and calling the shortcut menu (right mouse button).
	No module is in this slot on the controller!

**1.2.3.3.6.2 Right side of the window**

The right side of the window contains additional information and settings for the module which is marked on the left side. The right section is divided into different tabs in order to show the information more clearly.

Which tab is shown depends on which module you have marked in the left section.

<b>Tab</b>	<b>Description</b>
Software	This tab allows access to the software configuration (see below).
Logbook	This tab shows entries from the logbook on the PLC.



The software configuration is integrated in the right side of the project window. Each module that can run software (CPU or PP) has the "Software" tab. This tab is also called the software configuration.



You can configure Automation Runtime operating system performance to suit your application using the software configuration (you can find an in–depth description of the software configuration in the online help).

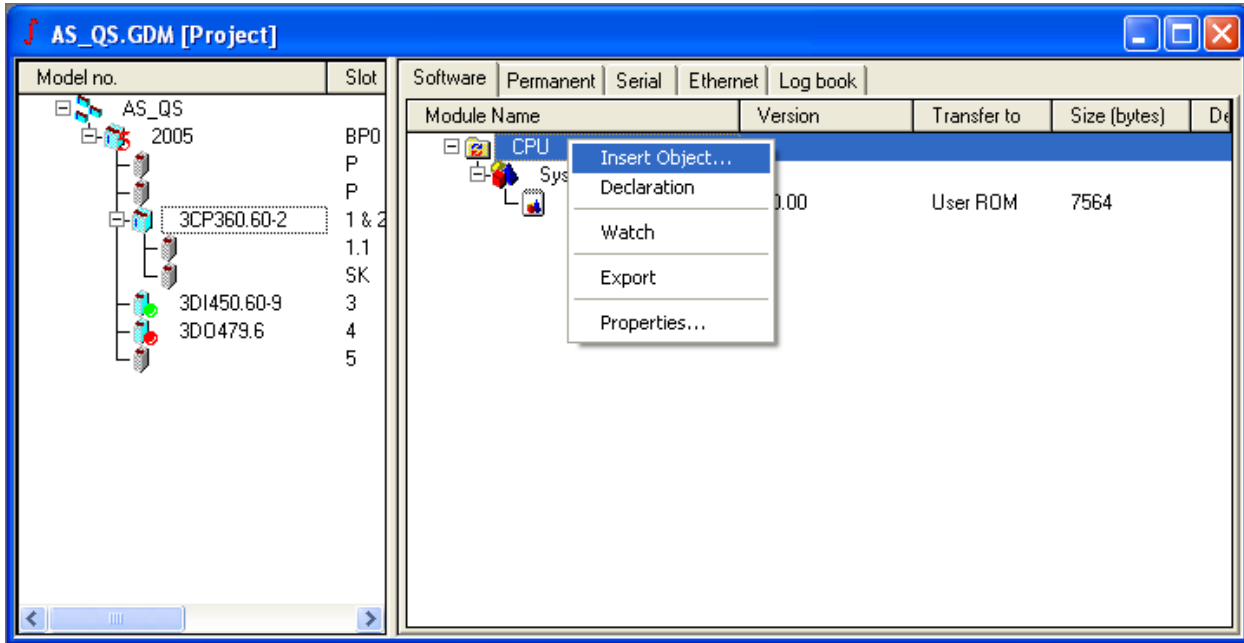
<b>Tab</b>	<b>Description</b>
I/O mapping	I/O mapping refers to the assignment of controller program variables to I/O channels on the PLC system. The Automation Runtime I/O system provides the controller program with the values of the respective I/O channels in the mapped variables.
I/O configuration	Module configuration: Displays all data points on the module. This list varies according to the properties of the module.

**1.2.3.4 Creating a cyclic object**

Now we will enter our example program using the Ladder Diagram (LD) programming language. Since our example program should be executed cyclically, we have to create a cyclic object for it.

**1.2.3.4.1 Inserting a cyclic object**

A cyclic object is inserted in the following manner:



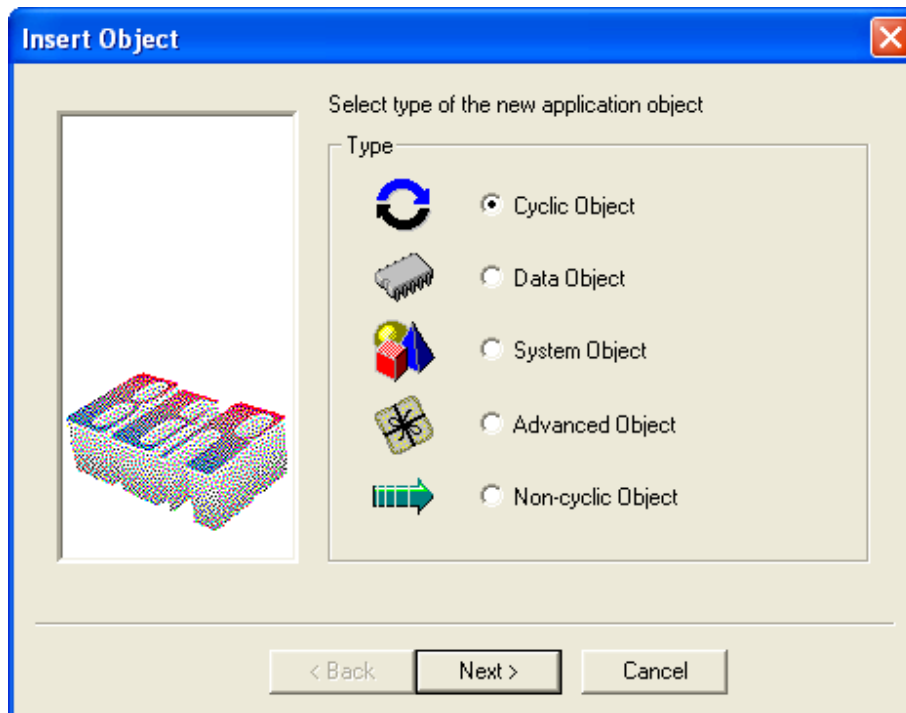
- On the left side of the project window, select a module that can contain a software object (CPU or parallel processor). In our example, this is the CP360 module.
- Select the Software tab in the right-hand window.
- Insert a new object by...

...Clicking on **New Object...** in the **Insert** menu, or

...Clicking with the right mouse button on the CPU symbol and selecting **Insert Object...** from the shortcut menu which appears, or

...Clicking on the CPU symbol and then pressing the INS key.

- Select the **Cyclic Object** resource in the dialog box which appears and then click **Next >**.



## 1.2.3.4.1.1 Resource types

The following resource types can be selected in the **Insert Object** dialog box:

Resource	Description
Cyclic Object	<p>A cyclic object is executed in a defined time frame (cycle time). The operating system monitors to make sure the cycle time is not exceeded. To assign priorities, cyclic objects can be assigned various resources (different cycle times). Cyclic objects that carry out important, time critical tasks are assigned a resource with a shorter cycle time and are therefore executed more often than objects assigned a resource with a longer cycle time. Cyclic objects include:</p> <ul style="list-style-type: none"> <li>• Cyclic objects</li> <li>• Timer objects</li> <li>• Interrupt objects</li> <li>• Exception objects</li> </ul>
Data Object	Data objects are modules used to store data.
System Object	<p>B&amp;R modules (*.BR files) can be transferred to the controller as system objects. System objects include libraries or objects with additional system functions. Many of these objects are loaded by B&amp;R Automation Studio™ automatically and don't have to be inserted manually.</p> <p>Library: A collection of standard functions is represented in B&amp;R Automation Studio™ by a B&amp;R module (library). Each library contains functions that can save you time and effort solving standard problems. If a function from a library is used, that library is automatically imported into the project and installed on the controller during the next transfer.</p>
Advanced Object	NC axes, CAM profiles, etc.
Non-cyclic Object	Idle time objects – these objects use the system idle time. This means they are only executed if neither the operating system nor the cyclic objects require CPU time (system time).

## 1.2.3.4.2 Defining object names, object types, and resources

After defining the resource type in the **Insert Object** dialog box, the **New Object** dialog box used to define the object name, object type and resource is opened:

Enter the desired name in the Name field ("logic1" in our example) for the new object. Please be aware of the limitations for object names given in the context-sensitive help. Now select Ladder Diagram for the object type and Cyclic #1 – [10 ms] for the resource.



You can assign various resources to objects to define priority between the individual cyclic objects. A separate cycle time can be set for each cyclic resource and each timer resource. All objects assigned this resource are executed once within this cycle time. Objects that carry out important, time-critical tasks are assigned a resource with a shorter cycle time and are therefore executed more often than objects assigned a resource with a longer cycle time.

After acknowledging these entries by clicking on the **Finish** button, the cyclic object is entered in the software configuration and the LAD editor is opened.

#### 1.2.3.4.2.1 Object types

The following object types are presently possible for cyclic objects (see "Type" in the "New Object" dialog box):

B&R Automation Basic™	..... formerly PL2000
C program (C language)	..... ANSI C
Ladder Diagram	..... IEC 1131 Ladder Diagram (LD)
Sequential Function Chart	..... IEC 1131 Sequential Function Chart (SFC)
Instruction List	..... IEC 1131 Instruction List (IL)
Structured Text	..... IEC 1131 Structured Text (ST)

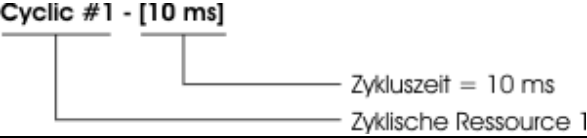
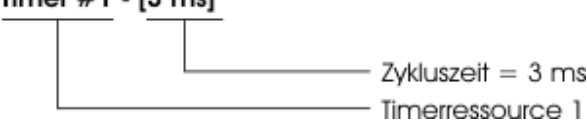
#### 1.2.3.4.2.2 Resources

The following resources are possible for cyclic objects (see **Resource** in the **New Object** dialog box):

Time-controlled resources:

Resource	Description
----------	-------------



<p>Cyclic #x – [...ms]</p>	<p>Objects assigned this resource are executed cyclically. The Automation Runtime operating system calls the object. Cyclic resources have a fixed name (Cyclic) which the cycle time is added to (time for one cycle). Example:</p> <p><b>Cyclic #1 - [10 ms]</b></p> 
<p>Timer #x – [...ms]</p>	<p>Objects with this resource are carried out cyclically using special hardware timers. Timer resources offer a high degree of consistency and have higher priority than cyclic resources. Timer resources have a fixed name (Timer) which the cycle time is added to (time for one cycle). Example:</p> <p><b>Timer #1 - [3 ms]</b></p> 

**Important:**

All resources described up to this point (cyclic and timer resources) are time-controlled. However, there are also resources that are not time controlled, but instead are carried out when a certain event occurs. We will differentiate between two types of event-controlled resources: interrupt resources and exception resources.

Event-controlled resources

Resource	Description
<p>Interrupts</p>	<p>Interrupts are asynchronous events (triggered by hardware) that interrupt cyclic program execution.</p> <p>Interrupts can only be generated by special hardware. An example of an interrupt-capable module is the DI400 digital input module (B&amp;R SYSTEM 2010). 8 digital inputs can be used as CSI inputs (CSI = "Change of State Interrupt") on this module. If the state of an enabled CSI input changes, an interrupt is triggered on the CPU.</p> <p>An object can be created for each interrupt capable module to handle the event.</p>
<p>Exceptions</p>	<p>Exceptions are signals that come from the PCCSW operating system and indicate certain exception states. They can be compared to interrupts triggered by the system to indicate a fatal error. Objects which are assigned an exception resource can be used to react to these fatal errors (e.g. inserting and removing I/O modules, power failure for RIO slave, etc.).</p>

1.2.3.4.3 Creating the ladder diagram

Now we will program the circuit diagram (see the [Task definition](#) section) in the Ladder Diagram editor (LD editor).

1.2.3.4.3.1 Setting the digital inputs

Digital inputs can be inserted in the LD editor as follows:

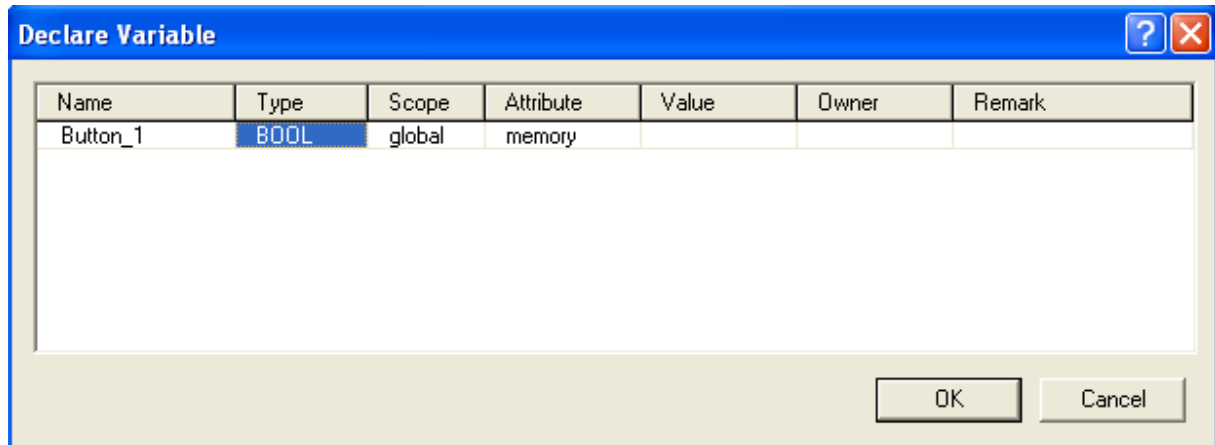
- Insert a symbol for the digital input by...

...Clicking on the  button in the toolbar, or

...Pressing "c", or

...Clicking on the **Contact** command in the **Insert** menu and selecting **Normally open contact** from the submenu.

- You now have to assign a name to the digital input. In this case, it will be called "Button\_1". After confirming the input with the Enter key, a declaration dialog box is opened.



### Name

The name of the variable is defined by IEC 1131-3.

Name length:

- ◆ max. 10 characters (up to operating system software version 2.0)
- ◆ max. 32 characters (from operating system software version 2.0 onwards)

### Type

The data type of a variable can either be entered directly or selected from a dialog box. This dialog box is opened by pressing the spacebar.

Any characters already entered act as a filter.

### Scope

The user can define the scope of the variable in a drop-down box. The scope is predefined as **global** in the global editor. The global declaration can only be called if the first element in the tree structure is selected in the software configuration.

The declaration can be called in such a way that it relates only to an object. Only variables used within this object are displayed. For objects, the scope of variables can also be defined as **local**. An object-specific declaration is called if ...

- ◆ ... The editor of an object is open and the window is in the foreground (e.g. LD editor open), or
- ◆ ... An object (e.g. LD object) is selected in the software configuration.

The drop-down box is called up by double-clicking or by pressing the spacebar.

Selection is made using the cursor keys or mouse.

### Attribute

The attribute determines the type of variable. Ultimately it also describes the way in which the variable is used in the system. In this field too you can either type in directly from the keyboard or select from a dialog box (called by pressing the spacebar).

### Value

For variables this value corresponds to the INIT value of internal variables after a warm restart.

For constants this value defines the constant.

### Owner

If library-specific variables are made available, the name of the library is shown in this column.

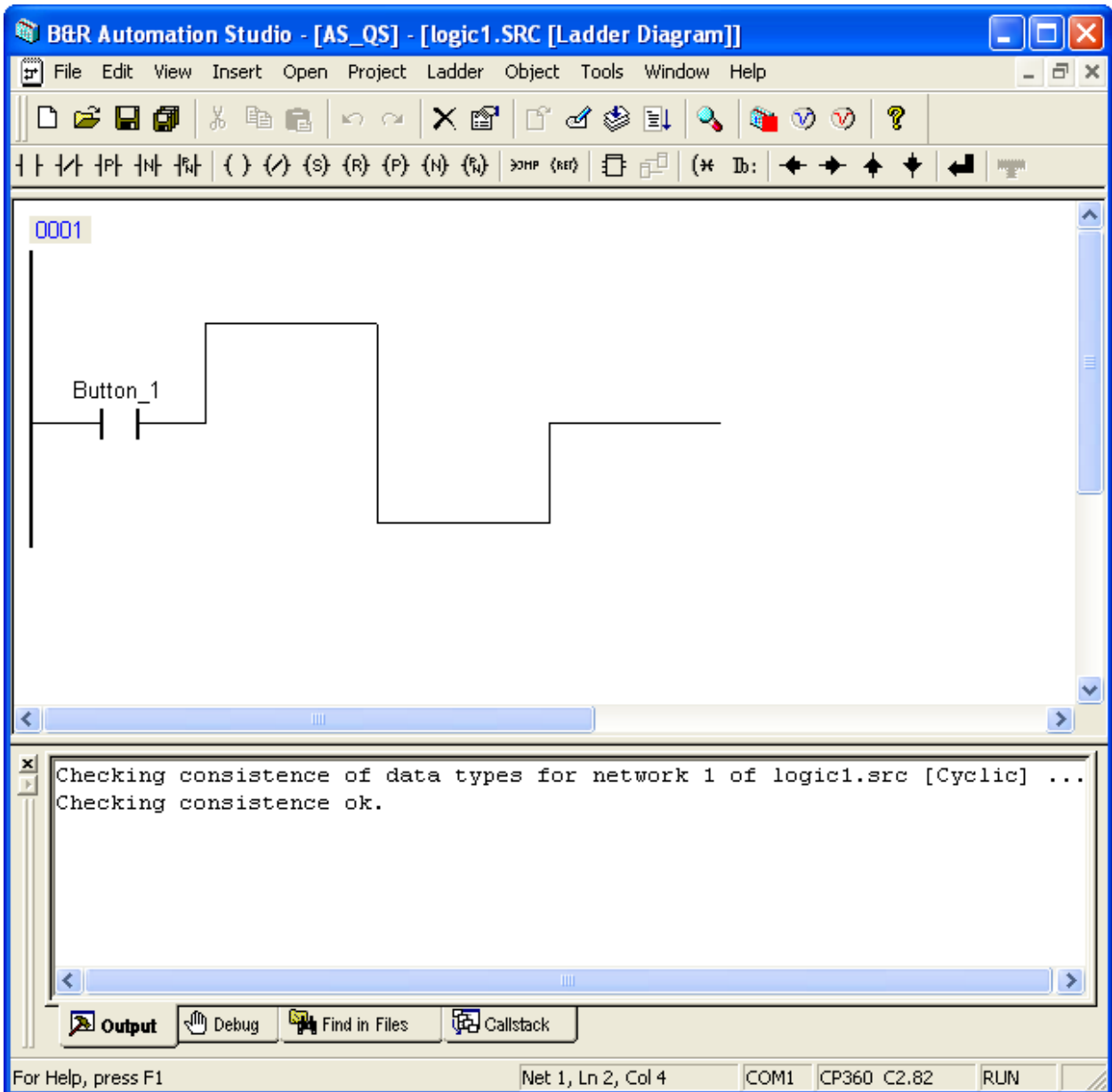
### Remark

The remark for a variable is for descriptive purposes only. A description of up to 34 characters can be entered here for each variable.

...If global PVs have already been defined in another cyclic task, they can also be selected by pressing the spacebar.

#### 1.2.3.4.3.2 Drawing connection lines


Connection lines can be drawn with the ALT + CURSOR keys or with the , , , and  icons.



#### 1.2.3.4.3.3 Setting digital outputs

Digital outputs can be inserted in the LD editor as follows:

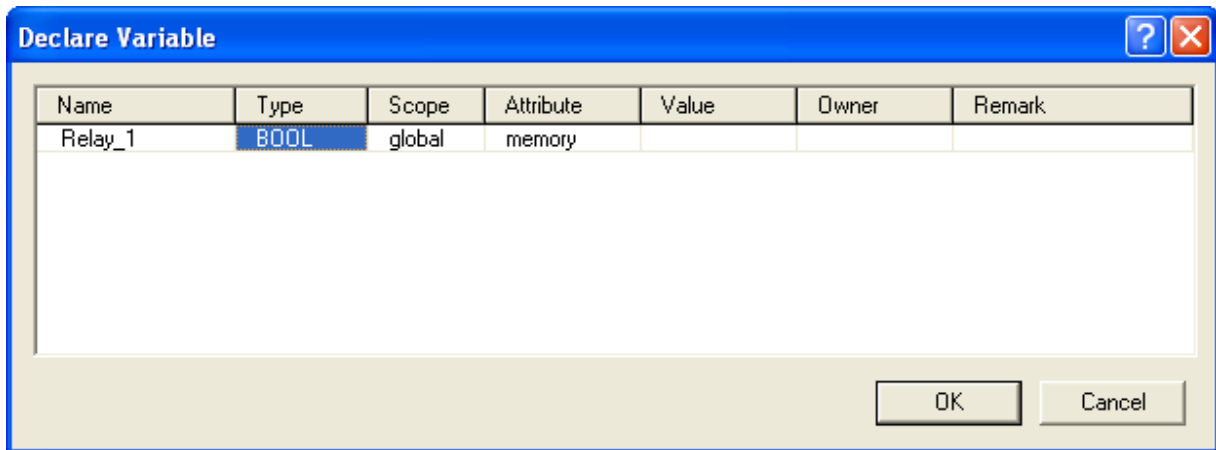
- Insert a symbol for the digital output by...

...Clicking on the  icon in the toolbar,

...Pressing the SHIFT+C shortcut key, or

...Clicking on **Coil** in the **Insert** menu and selecting Coil from the submenu shown.


- You now have to assign a name to the digital output. In this case, it will be called "Relay\_1". After confirming the input with the Enter key, a declaration dialog box is opened.



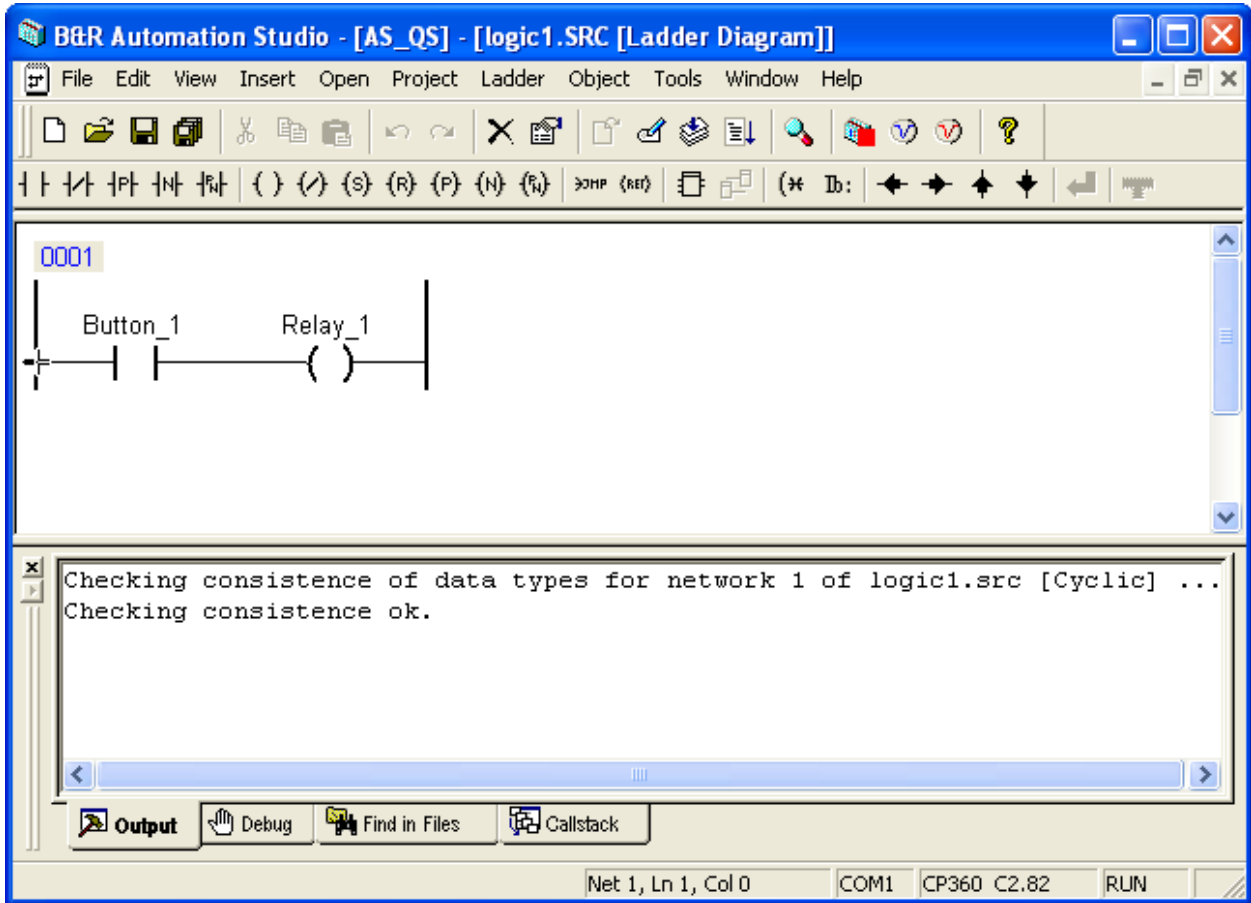
...If global PVs have already been defined in another cyclic task, they can also be selected by pressing the spacebar.

The list only contains variables that are compatible with the data type of the new contact!

#### 1.2.3.4.3.4 Checking and optimizing LD

If you click on the  button in the toolbar or press the ENTER key, the ladder diagram is checked for open connection lines or other errors and optimized (extra connection lines are deleted, etc.).

Checked and optimized ladder diagram:



#### 1.2.3.4.3.5 Showing/Hiding information

It is possible to show additional information in the ladder diagram. The Type, Scope, and Remark commands in the "View" menu and the shortcut menu of the LD editor (opened with the right mouse button in the LD editor) can be used to select which information is to be shown. A check mark to the left of the menu command means that this information is being displayed.

Meaning of the additional information:

Information	Description
Type	Data type of process variables (detailed information concerning data types can be found in the online help).
Scope	Valid range of process variables (detailed information about scope can be found in the online help).
Comment	Up to 34 character long description of the process variable (see the "Variable declaration" section).

#### 1.2.3.4.3.6 Saving and closing LD

To save the ladder diagram "logic1" on the hard drive of the programming device (under C:\ASProjects\AS\_QS\...),

...Click on the  icon in the toolbar,

...Pressing the shortcut key **CTRL+ S**, or

...Select **Save** from the File menu.

When saved, the ladder diagram is checked and optimized again. Once saved, the Ladder Diagram editor can be closed.

The LD editor can be closed by...

...Clicking on **Close** in the File menu, or

...Clicking on the  button in the upper right corner of the LD editor window.

#### 1.2.3.4.4 Mapping variables

Now the variables that we've defined will be mapped to an input or output, i.e. we will establish a dependency with the hardware.

Proceed as follows to map the digital input "Button\_1":

- In the left section of the project window, click on the symbol for the DI450 module (digital input module). The right-hand window will then display the I/O Mapping and I/O Configuration tabs.
- Clicking on the I/O Mapping tab displays a list of all data points for the DI450 module.
- To assign the variable name "Button\_1" to the first digital input on the DI450 module...

...Double-click on the first line of the **PV Name** column and enter the variable name "Button\_1". The entry for the variable name is completed when the Enter key is pressed. Or

...Place the cursor in the first line of the list in the **PV Name** column and press the spacebar. Then the "Button\_1" PV can be selected from the drop-down menu. Press Enter to complete the entry.

- The task class must also be defined where the variable should be read/written. To do so, click on the **Task class** column in the corresponding line and select the desired task class from the menu.

I/O Mapping		I/O Configuration		
Logical Name	Data Type	Task Class	PV Name	Description
+ ModuleOk	BOOL			Module status (1 = module present)
+ DigitalInput01	BOOL	Cyclic#1	Button_1	24 VDC, <1.2 ms switching delay, sink
+ DigitalInput02	BOOL			24 VDC, <1.2 ms switching delay, sink
+ DigitalInput03	BOOL			24 VDC, <1.2 ms switching delay, sink
+ DigitalInput04	BOOL			24 VDC, <1.2 ms switching delay, sink
+ DigitalInput05	BOOL			24 VDC, <1.2 ms switching delay, sink
+ DigitalInput06	BOOL			24 VDC, <1.2 ms switching delay, sink
+ DigitalInput07	BOOL			24 VDC, <1.2 ms switching delay, sink
+ DigitalInput08	BOOL			24 VDC, <1.2 ms switching delay, sink
+ DigitalInput09	BOOL			24 VDC, <12 ms switching delay, sink / s
+ DigitalInput10	BOOL			24 VDC, <12 ms switching delay, sink / s

Proceed as follows to map the digital output "Relay\_1":

- In the left section of the project window, click on the symbol for the DO479 module (digital output module). The right-hand window will then display the I/O Mapping and I/O Configuration tabs.
- Clicking on the I/O Mapping tab displays a list of all data points for the DO479 module.
- To map the variable name "Relay\_1" to the first digital output on the DO479 module...

...Double-click on the first line of the **PV Name** column and enter the variable name "Relay\_1". The entry for the variable name is completed when the Enter key is pressed. Or

...Place the cursor in the first line of the list in the **PV Name** column and press the spacebar. Then the "Relay\_1" PV can be selected from the drop-down menu. Press Enter to complete the entry.

- The task class must also be defined where the variable should be read/written. To do so, click on the **Task class** column in the corresponding line and select the desired task class from the menu.

I/O Mapping		I/O Configuration		
Logical Name	Data Type	Task Class	PV Name	Description
+ ModuleOk	BOOL			Module status (1 = module present)
+ DigitalOutput01	BOOL	Cyclic#1	Relay_1	24 VDC / 0.5 A, source
+ DigitalOutput02	BOOL			24 VDC / 0.5 A, source
+ DigitalOutput03	BOOL			24 VDC / 0.5 A, source
+ DigitalOutput04	BOOL			24 VDC / 0.5 A, source
+ DigitalOutput05	BOOL			24 VDC / 0.5 A, source
+ DigitalOutput06	BOOL			24 VDC / 0.5 A, source
+ DigitalOutput07	BOOL			24 VDC / 0.5 A, source
+ DigitalOutput08	BOOL			24 VDC / 0.5 A, source
+ DigitalOutput09	BOOL			24 VDC / 0.5 A, source
+ DigitalOutput10	BOOL			24 VDC / 0.5 A, source
+ DigitalOutput11	BOOL			24 VDC / 0.5 A, source

The "Remark" column usually contains a short function description of the input/output of the selected module.

This column can also be used to enter an additional description. Double-clicking on the "Remark" column will allow text to be entered there. Input is completed when the Enter key is pressed.

This additional description is only for documentation and is not transferred to the controller.

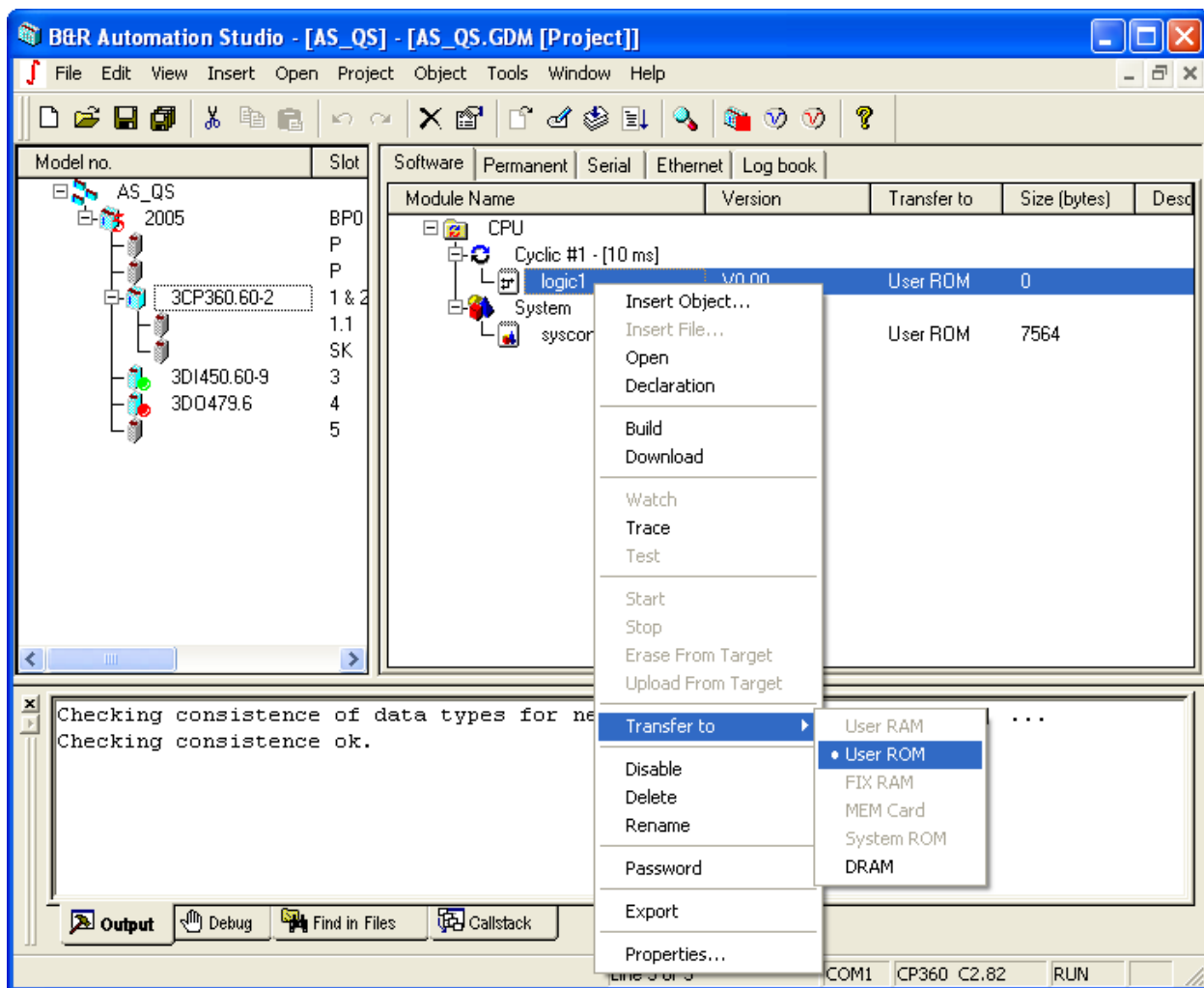
### 1.2.3.5 Building the object and transferring it to the controller

The ladder diagram has to be converted to the processor's machine language for the controller. B&R Automation Studio™ has a compiler for this purpose. The compiler provides an executable program module (B&R module) in machine language for the processor.

Before compiling the example program "logic1" and transferring it to the controller, you have to define the memory area that will be used as target for object transfer.

#### 1.2.3.5.1 Defining the memory area

To define the target memory area for the object "logic1", click on the object symbol in the software configuration and select the **Transfer to** command from the **Object** menu. This displays a list of all possible memory areas for this object:





Overview of possible memory areas:

Memory area	Description
User RAM	User RAM is a conditionally nonvolatile memory area (buffered when power is not applied to the controller) and is available for application software (cyclic objects, data objects, etc.).
User ROM	User ROM can be used to store cyclic objects, data objects, etc. in nonvolatile memory. User ROM is executed as FlashPROM (electrically erasable read-only memory).
FIX RAM	Using B&R Automation Studio™ (software configuration), you can configure a part of User RAM to behave like User ROM during a COLD RESTART. This memory area is called FIX RAM. Objects stored in FIX RAM are not deleted during a COLD RESTART.  COLD RESTART boot mode corresponds to the first time a processor is initialized. During a COLD RESTART, the operating system is completely restarted and all RAM data is deleted (exception: objects stored in FIX RAM).
MemCard	Some processor modules can be equipped with a PCMCIA memory card (also called a MemCard). The MemCard can be used to store cyclic objects, data objects, etc. in nonvolatile memory. That means this memory corresponds to either the User ROM (MemCard as FlashPROM) or FIX RAM (MemCard as SRAM).
SYS ROM	The operating system B&R Automation Runtime™ is stored in System ROM. System ROM is a nonvolatile memory area in the form of a FlashPROM (electrically erasable read-only memory).
DRAM	The object is stored directly in DRAM when transferred to the controller. DRAM is a nonvolatile memory area and is available for application software (cyclic objects, data objects, etc.).

1.2.3.5.2 Building and transferring

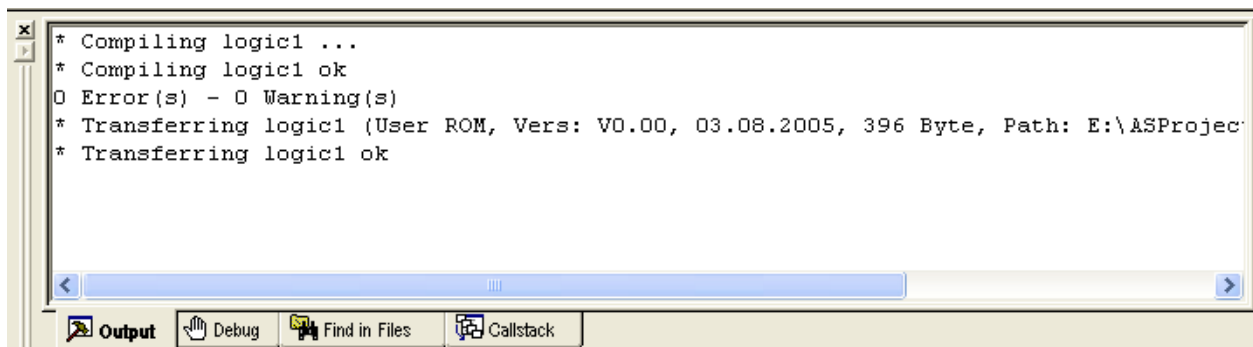
To compile the example program and transfer it to UserROM...

...Click on the  icon in the toolbar, or

...Select the **Transfer to Target** command in the **Project** menu.

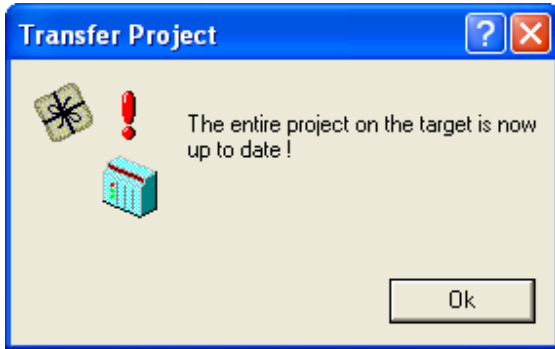
When run, this command first builds the entire project. It then transfer objects to the defined memory areas where they are started.

The progress and completion status of the build procedure is shown in the message window:



To avoid errors, B&R Automation Studio checks the structure of the target system hardware, the version of the operating system, and all previously installed software objects during a transfer. Any problems identified are clearly signaled so that additional errors can be avoided.

The following message is given after the project has been successfully transferred:



The example program is now running in UserROM. The digital output is set or reset according to the logical state of the digital input.


### 1.2.3.6 Testing the object

B&R Automation Studio™ recognizes many problems during programming and indicates them using compiler messages; however, it is necessary to have a comfortable debugger for large projects. Because of this, powerful diagnostics tools have been integrated into B&R Automation Studio™.

#### 1.2.3.6.1 LD monitor

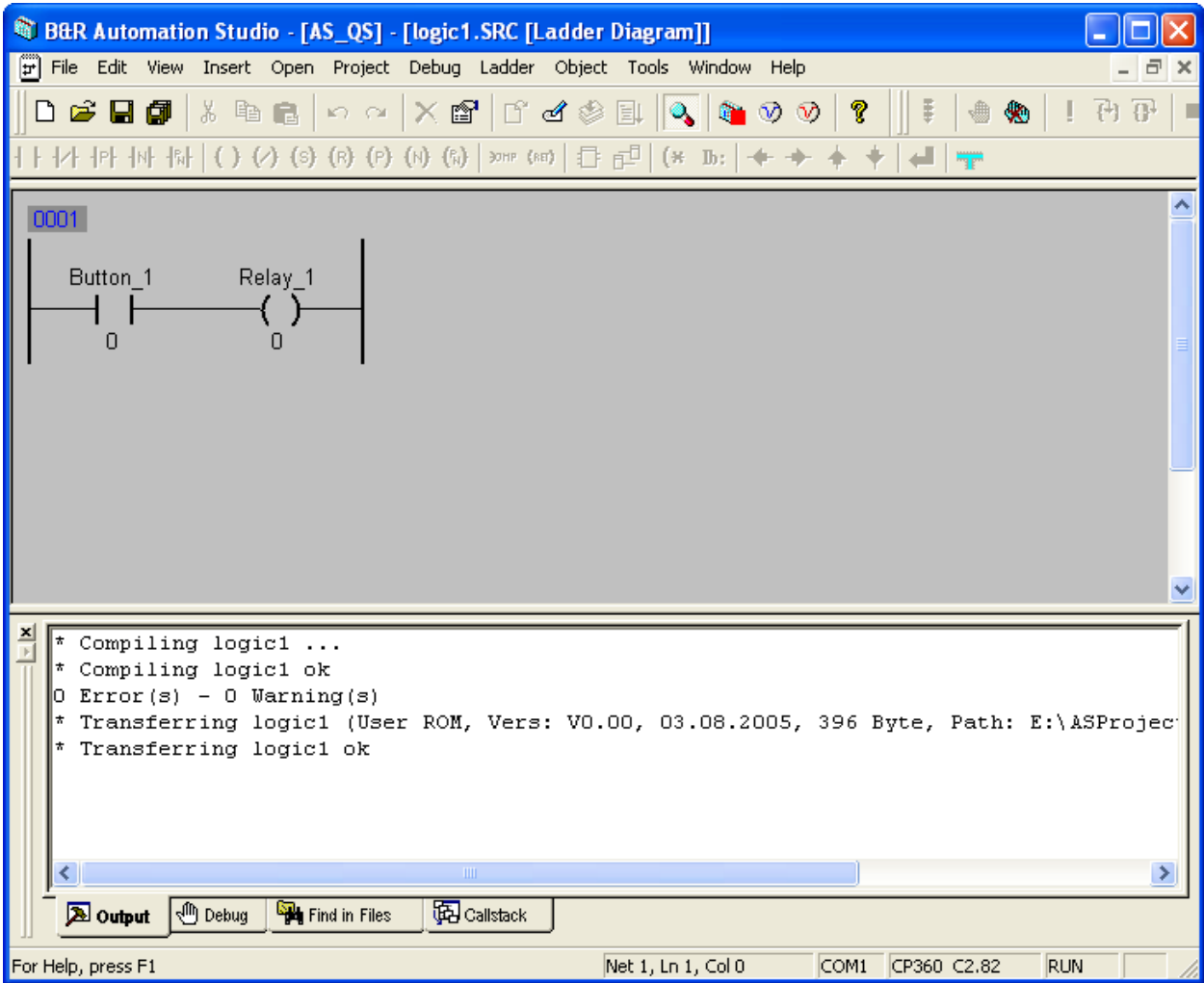
With the ladder diagram monitor (or LD monitor), it is possible to show and define process variables from a ladder diagram running on the controller directly in the LD editor itself.

To open the LD monitor, you have to first open up the LD editor. To do this, double-click on the symbol for example program "logic1" in the software configuration. You can then open the LD monitor from the LD editor by...

...Clicking on the  icon in the toolbar,

...Pressing the shortcut key **CTRL+ M**, or

...iSelecting the **Monitor** command from the **View** menu.



1.2.3.6.1.1 Defining a variable

Using the Force function in the LD monitor, you can change the state of digital inputs from B&R Automation Studio™. This function can be used to test the example program "logic1".

**Note:**

If a process variable is forced, it is assigned a specific value and remains at this value. If an object writes to a forced process variable, the changed value is only valid for the current cycle. Then the variable is overwritten with the force value entered.

The Force function is also available in other B&R Automation Studio™ diagnostics tools.

To force the variable "key\_1", click on the symbol for the digital input "key\_1" with the right mouse button and select **Force** from the shortcut menu. Then a dialog box is shown asking if the variable should be forced. Close this message box with **OK**.

**Important:**

A check mark to the left of the **Force** menu item in the shortcut menu means that the variable is forced. The values of forced variables are also shown in red in the LD monitor.

The value of the digital input "Button\_1" can then be set to 1 by...

...Clicking on the symbol for the digital input with the right mouse button again and selecting "Value" from the shortcut menu. The value "1" can then be entered. Press Enter to complete the entry. Or

...Clicking on the symbol for the digital input and pressing the spacebar. The value "1" can now be entered. Press Enter to complete the entry.


The forced value is then shown in RED and the digital output "Relay\_1" is set:



#### 1.2.3.6.1.2 Closing the LD monitor

You can close the LD monitor by...

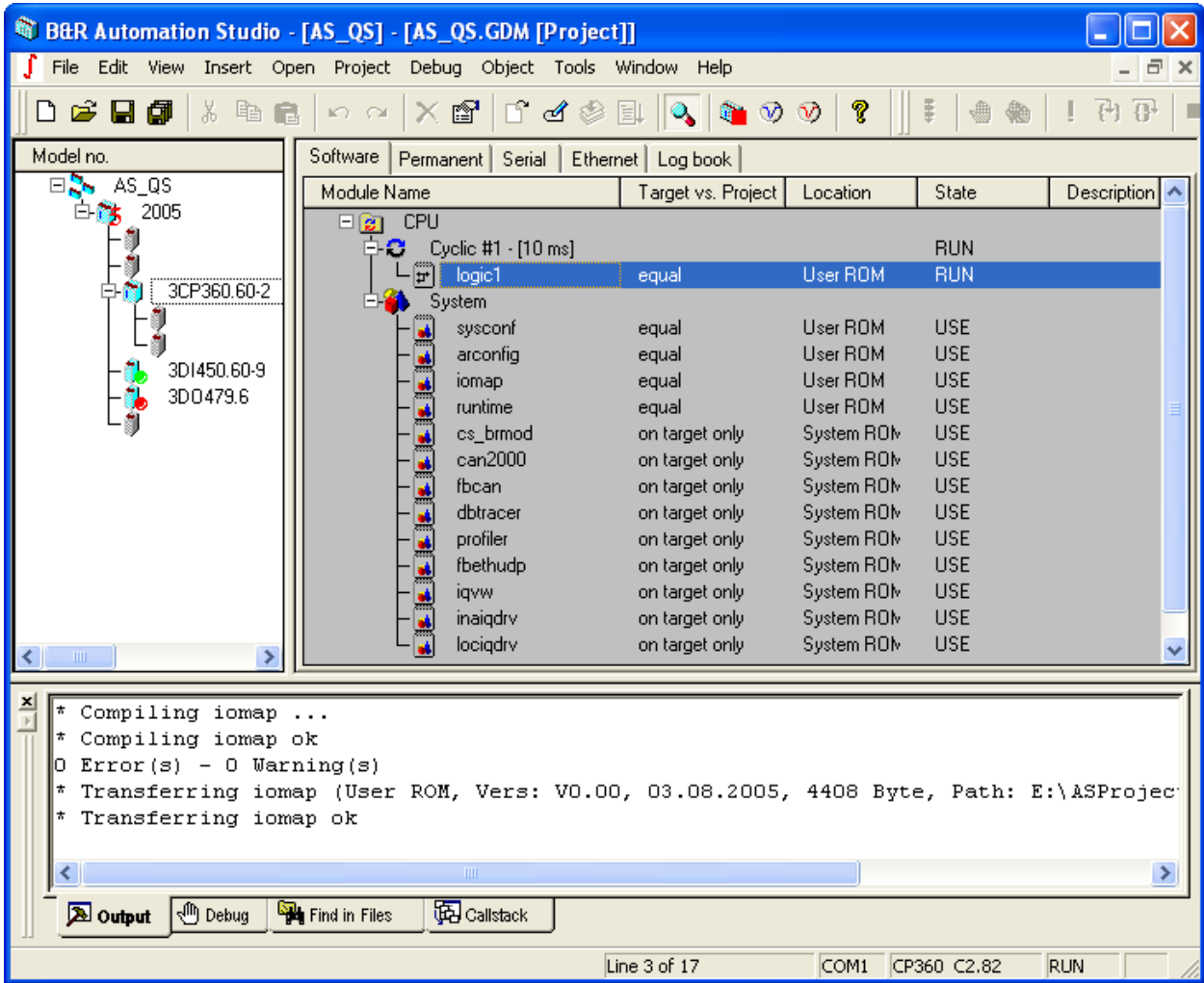
...Clicking on **Close** in the **File** menu, or

...Clicking on the  button in the upper right corner of the LD monitor window.

Additional information concerning the LD monitor can be found in the B&R Automation Studio™ online help.

#### 1.2.3.6.2 System monitor

When you close the LD monitor window, B&R Automation Studio™ remains in monitor mode. The system monitor (system monitor = software configuration in monitor mode) is shown:



To turn monitor mode on or off,

...Click on the  icon in the toolbar,

...Press the shortcut key **CTRL+ M**, or

...Select the **Monitor** command from the View menu.

Different information is shown in the software configuration depending on whether B&R Automation Studio™ is in Monitor mode or not.

#### 1.2.3.6.2.1 Monitor mode is turned off

In this mode, all objects in the current project are shown. Objects only found on the controller are not shown in this mode. The following information is displayed in the right-hand part of the project window:

Module Name	Version	Transfer to	Size (bytes)	Description
<ul style="list-style-type: none"> <li>[-] CPU               <ul style="list-style-type: none"> <li>[-] Cyclic #1 - [10 ms]                   <ul style="list-style-type: none"> <li>[-] logic1 V0.00 User ROM 396</li> </ul> </li> <li>[-] System                   <ul style="list-style-type: none"> <li>[-] sysconf V2.81 User ROM 7616</li> <li>[-] arconfig V0.00 User ROM 1012</li> <li>[-] iomap V0.00 User ROM 4408</li> <li>[-] runtime V1.09 User ROM 31244</li> </ul> </li> </ul> </li> </ul>				

Column	Description
Module name	Symbol and name of the software object.
Version	Version number of the software object.
Transfer to	Target memory for the software object (valid for the next transfer).
Size (bytes)	Size of the compiled software object in bytes. If the object has not been compiled yet, then the value 0 is shown.

#### 1.2.3.6.2.2 Monitor mode is turned on

In this view, the differences between the current project (computer) and the controller are shown. The following information is provided in addition to the symbol and name of an object:

Module Name	Target vs. Project	Location	State	Description
CPU				
Cyclic #1 - [10 ms]			RUN	
logic1	equal	User ROM	RUN	
System				
sysconf	equal	User ROM	USE	
arconfig	equal	User ROM	USE	
iomap	equal	User ROM	USE	
runtime	equal	User ROM	USE	
cs_brmod	on target only	System ROM	USE	
can2000	on target only	System ROM	USE	
fbcan	on target only	System ROM	USE	
dbtracer	on target only	System ROM	USE	
profiler	on target only	System ROM	USE	
fbethudp	on target only	System ROM	USE	
iqvw	on target only	System ROM	USE	
inaiqdrv	on target only	System ROM	USE	
lociqdrv	on target only	System ROM	USE	

Column	Description
Module name	Symbol and name of the software object.
Target vs. project	Objects in the project and on the controller are compared and the result is shown: <ul style="list-style-type: none"> <li>• On target only Object is only on the controller (target system).</li> <li>• Not on target Object is not on the controller (only in project).</li> <li>• Older on target Object in the project is more current than the one on the controller.</li> <li>• Newer on target Object on the controller is more current than the one in the project.</li> <li>• Identical Object in the project and on the controller are identical.</li> </ul>
Location	Memory where the object can be found on the controller.
State	Status of the object. <p>The status of cyclic objects (cyclic objects, timer objects, interrupt and exception objects) and idle time objects:</p> <ul style="list-style-type: none"> <li>• <b>RUN</b> Software object is being executed.</li> <li>• <b>STOP</b> Software object stopped (NOT being executed).</li> </ul> <p>The status of system and data objects:</p> <ul style="list-style-type: none"> <li>• <b>RDY</b> The software object (system or data object) is ready to be used by other objects.</li> <li>• <b>USE</b> The software object (system or data object) is being used by other objects.</li> </ul>
Description	A data definition can be entered here by the user. The length of the text cannot exceed 50 characters. <p>Description text can be modified as follows:</p>

- Mark the text and open the **Properties dialog box** with the right mouse button.
- Once there, open the **General** tab.
- In the "Description" section, enter the desired text.

Additional information concerning the system monitor can be found in the B&R Automation Studio™ online help.

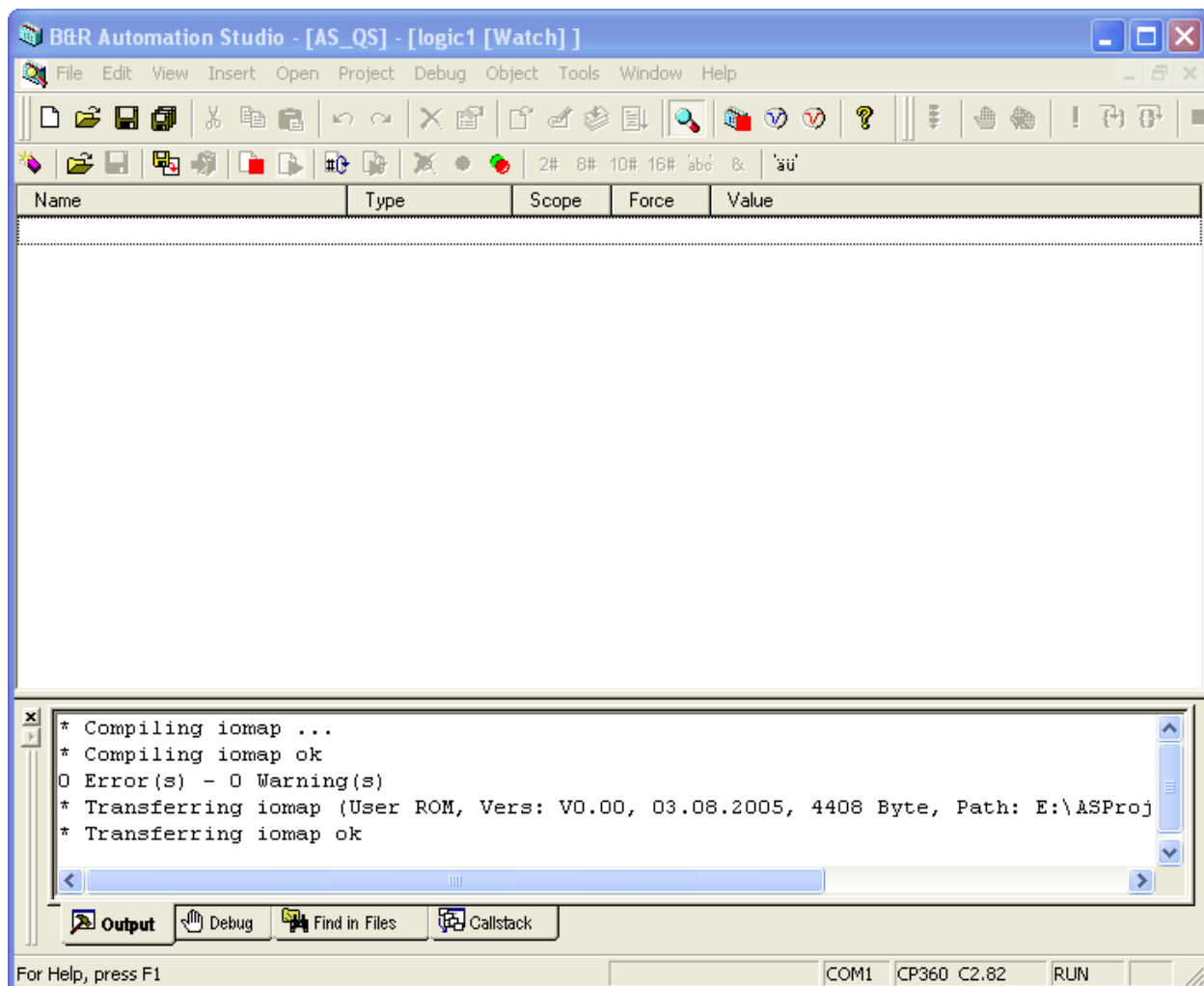
### 1.2.3.6.3 Watch

Using the Watch function, you can view, monitor, and change the values of process variables found on the controller. Open the Watch window for the object "logic1" by...

...Clicking on the symbol "logic1" in the software configuration and selecting the **Watch** command in the **Open** menu, or













...Clicking on the symbol "logic1" with the right mouse button in the software configuration and selecting the "Watch" command from the shortcut menu which appears.

The following information is shown in the Watch window:



Column	Description
Name	Names of process variables are shown in this column.
Type	The data type of each variable is shown in this column.



	The data type defines the memory requirements and value range for a process variable (detailed information can be found in the online help).																		
Scope	<p>The user can define the scope of the variable in a drop-down box. The scope is predefined as <b>global</b> in the global editor.</p> <p>The declaration can be called in such a way that it relates only to an object. Only variables used within this object are displayed. For objects, the scope of variables can also be defined as <b>local</b>. An object-specific declaration is called if ...</p> <ul style="list-style-type: none"> <li>• ... The editor of an object is open and the window is in the foreground (e.g. LD editor open), or</li> <li>• ... An object (e.g. LD object) is selected in the software configuration.</li> </ul> <p>The drop-down box is called up by double-clicking or by pressing the spacebar.</p> <p>Selection is made using the cursor keys or mouse.</p>																		
Force	<p>The following information is shown in this column:</p> <table border="1"> <thead> <tr> <th>Symbol</th> <th>Description</th> <th>Force status</th> </tr> </thead> <tbody> <tr> <td></td> <td>Force symbol for input (green)</td> <td>Inactive</td> </tr> <tr> <td></td> <td>Force symbol for input (green)</td> <td>Active</td> </tr> <tr> <td></td> <td>Force symbol for output (red)</td> <td>Inactive</td> </tr> <tr> <td></td> <td>Force symbol for output (red)</td> <td>Active</td> </tr> <tr> <td>No symbol</td> <td>Variable in memory</td> <td>--</td> </tr> </tbody> </table>	Symbol	Description	Force status		Force symbol for input (green)	Inactive		Force symbol for input (green)	Active		Force symbol for output (red)	Inactive		Force symbol for output (red)	Active	No symbol	Variable in memory	--
Symbol	Description	Force status																	
	Force symbol for input (green)	Inactive																	
	Force symbol for input (green)	Active																	
	Force symbol for output (red)	Inactive																	
	Force symbol for output (red)	Active																	
No symbol	Variable in memory	--																	
Value	<p>The value of the variable is shown in this column. You can choose from several display formats:</p> <ul style="list-style-type: none"> <li>• Binary</li> <li>• Octal</li> <li>• Decimal</li> <li>• Hexadecimal</li> <li>• String</li> </ul> <p>The commands Binary, Octal, Decimal, Hexadecimal and String in the View menu and in the shortcut menu for the Watch window (called with the right mouse button in the Watch window) can be used to select the view format. A dot · to the left of the menu item means that the variable values will be displayed in this format.</p>																		

1.2.3.6.3.1 Inserting a variable

All functions in the Watch window refer to the software object that was marked in the software configuration when the Watch function was called. That means you can only insert variables that exist in the selected object.

You can insert a variable in the Watch window as follows:

- The Watch window has to be the active window!
- Insert a new object by...

...Clicking on the  icon in the toolbar,

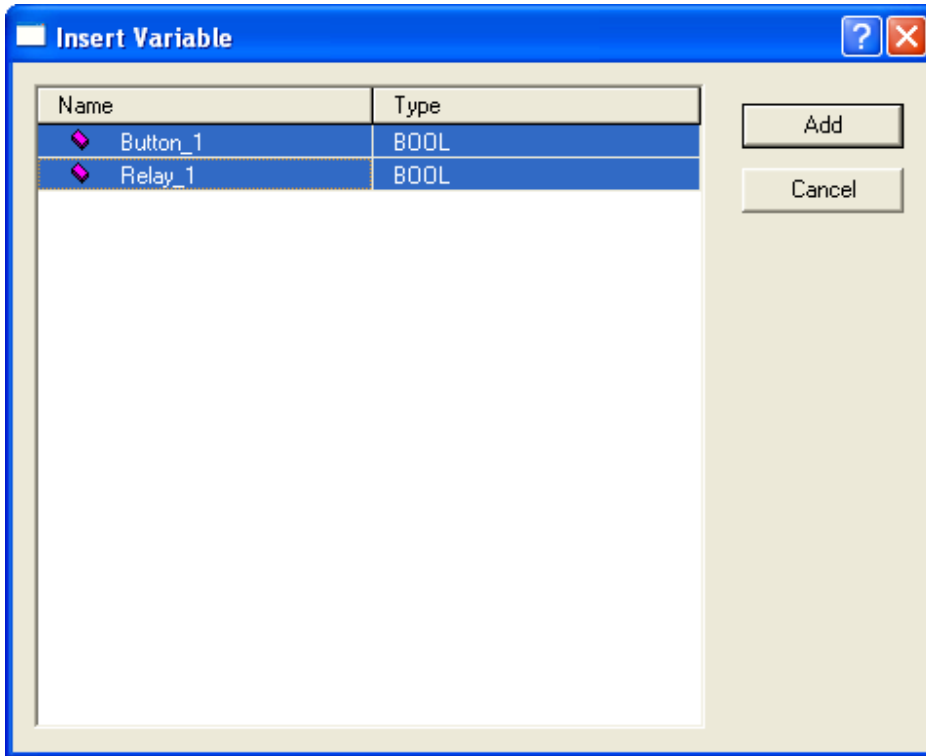
...Pressing INS and then the spacebar,

...Clicking "Variable" in the "Insert" menu, or

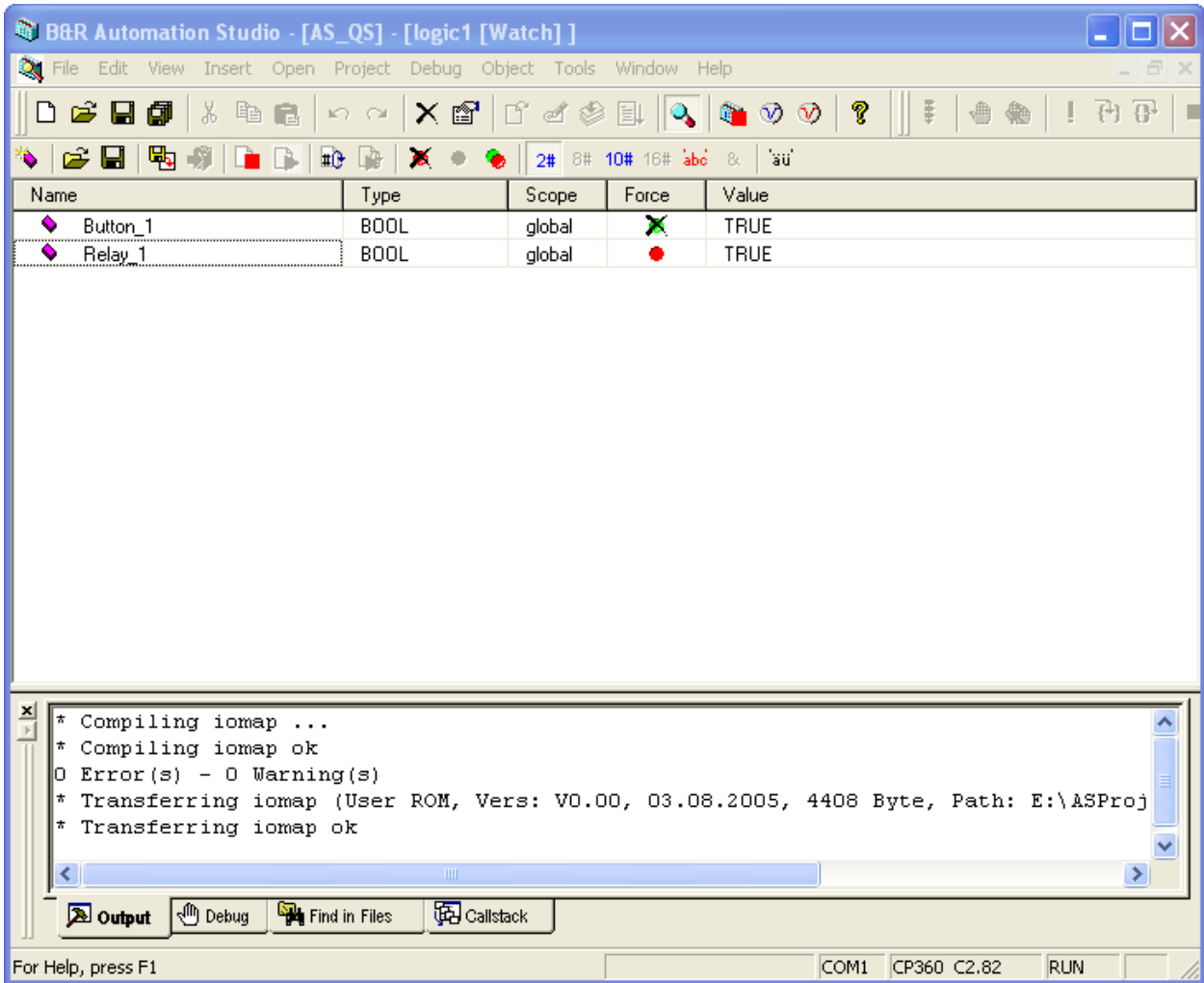
...Clicking in the Watch window with the right mouse button and selecting the **Insert variable** command in the shortcut menu which appears.

- In the **Insert variable** dialog box, select the desired variable (e.g. "Button\_1") and then click on the **Add** button.

You can also select several variable at the same time in the **Insert Variable** dialog box and insert them into the watch window. To do this, press and hold Shift when selecting the variables (with the mouse or cursor keys):



After closing the dialog box by pressing the **Add** button, all marked variables are inserted into the Watch window:



#### 1.2.3.6.3.2 Defining a variable

You can change the value of a variable as follows:

- Select the respective variable in the Watch window.
- Change the value of a variable by...

...Pressing the spacebar (the cursor has to be in the **Value** column) and entering the desired value,

...Clicking on "Variable" in the **Edit** menu and entering the desired value, or

...Entering the desired value directly (the cursor has to be in the **Value** column).

- Press Enter to accept the new value or ESC to disregard the change.


#### Important:

If the variable to be changed is an input or output as with our example (digital input "Button\_1" or digital output "Relay\_1"), the variable is automatically forced after acknowledgement.

### 1.2.3.6.3.3 Activating/Deactivating the force status

The force status of a variable can be activated manually.

To activate the force status of a variable, mark the variable in the Watch window and...

...Click on the  icon in the toolbar,

...Click on **Force off** in the **Object** menu, or



...Click on **Force off** in the shortcut menu (right mouse button).

A dialog box is then shown asking if the force status of the variable should be changed. Close this message box with **OK**.


#### Note:

A check mark to the left of the Force menu item in the Object menu and in the shortcut menu means that the variable is forced! The following symbols in the Force column of the watch window also indicate a forced variable.

Identifies an active force status

Symbol	Description	Force status
	Force symbol for input (green)	Active
	Force symbol for output (red)	Active

To deactivate the force status of a variable, mark it in the Watch window and click on **Force on/off** again in the **Object** menu or shortcut menu (right mouse button). A dialog box is then shown asking if the process variable should be forced. Close this message box with **OK**.

If you want to deactivate the force status of all variables in the Watch window at the same time, click on the  button in the toolbar or on **Force all off** in the **Object** menu.

### 1.2.3.6.3.4 Closing the Watch window

You can close the Watch window by...

...Clicking on **Close** in the **File** menu, or

...Clicking on the  button in the upper right corner of the Watch window.

Additional information concerning all functions of the Watch window can be found in the B&R Automation Studio™ online help.

## 1.2.4 Frequently asked questions

### 1.2.4.1 How do I install the operating system?

The Automation Runtime™ operating system is stored in the processor's System ROM (CPU or parallel processor).



System ROM is a nonvolatile memory area in the form of a FlashPROM (electrically erasable read-only memory).

You can download or update the B&R Automation Runtime™ operating system yourself using B&R Automation Studio™.

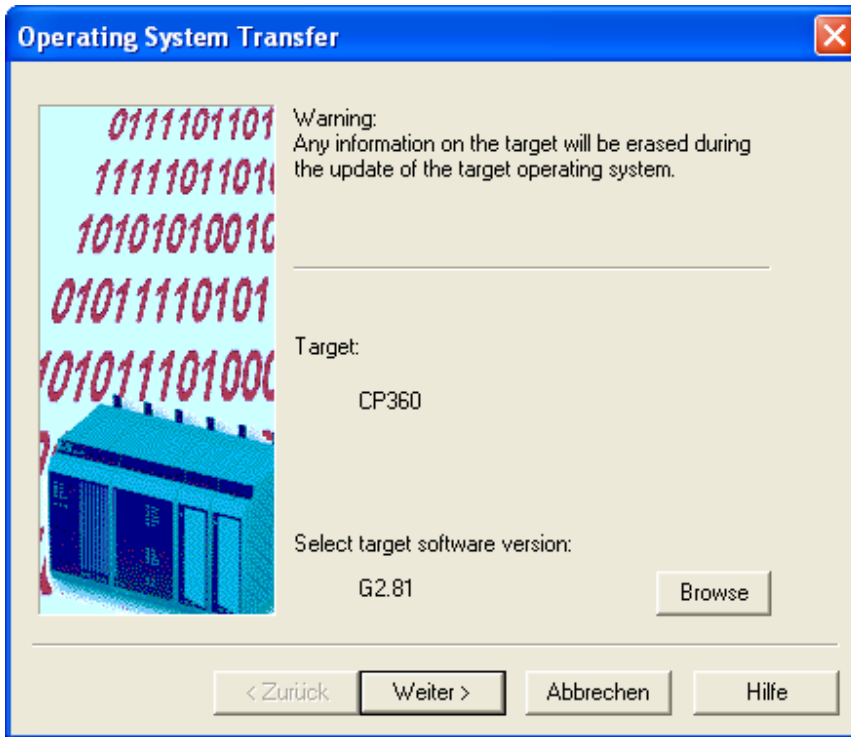
**Important:**

The operating system download or update is described for each module in the respective hardware users manual. You can also look in the module description integrated in B&R Automation Studio™. Please note the switch settings for bootstrap loader mode or RUN mode listed there.

**1.2.4.1.1 Processors with application memory**

Application memory is always delivered without an operating system installed. The first operating system installation (operating system download) is carried out as follows for processors with application memory:

1. Turn off power to the controller. This is absolutely necessary since the application memory can only be removed or inserted without power applied!
2. Insert new FlashPROM application memory and return power. Make sure that the write-protect switch on the front of the application memory is in the write position!
3. Take the online cable and connect the computer to the CPU where the operating system should be installed.
4. Start B&R Automation Studio™.
5. Open a project in B&R Automation Studio™. Use the **Open project...** command from the **File** menu for this.
6. Start the download procedure by opening up the **Services** item in the **Project** menu. Select **Transfer Operating System...** from the menu shown.
7. B&R Automation Studio™ now gathers all the information required for downloading the operating system from the controller (type and revision of the application memory, etc.).



8. If a different operating system version than the one displayed should be transferred to the target system, you can click on the **Browse** button to select another version.

9. FlashPROM is erased after clicking on **Next >** . The selected version of the operating system is then downloaded to FlashPROM. Download progress is shown on the screen. The download procedure can take a few minutes!

10. The controller is now ready for operation.

#### **Important:**

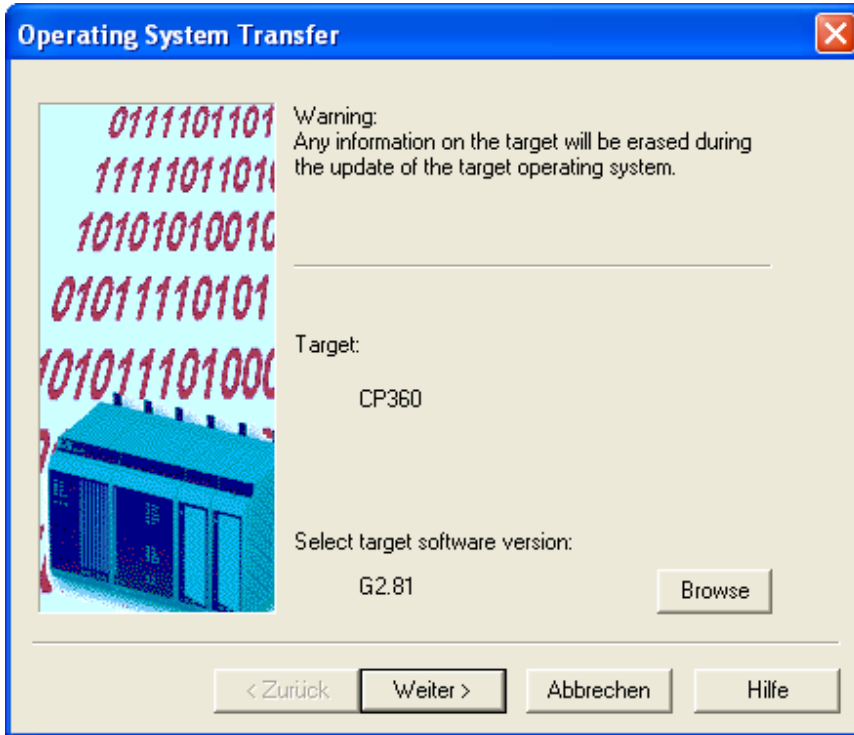
If the download procedure is interrupted, the operating system installation has to be started again!

#### **1.2.4.1.2 Processors without application memory**

An operating system is already installed on processors without application memory (CP260, IF260, CP360, IP161, XP152, CP430, CP47x, CP77x, AR102, AR105, ARxxx) before they are delivered.

Updating the operating system is only possible in RUN mode. Information about the RUN mode setting can be found in the hardware documentation. The following steps are necessary for an operating system update:

1. Take the online cable and connect the computer to the processor (CPU or PP) where the operating system should be updated.
2. Start B&R Automation Studio™.
3. Open a project in B&R Automation Studio™. Use the **Open project...** command from the **File** menu for this.
4. Start the download procedure by opening up the **Services** item in the **Project** menu. Select **Transfer Operating System...** from the menu shown.
5. B&R Automation Studio™ will now get all of the information necessary for the operating system download from the controller:



6. If a different operating system version than the one displayed should be transferred to the target system, you can click on the **Browse** button to select another version.

7. FlashPROM is erased after clicking on **Next >** . The selected version of the operating system is then downloaded to FlashPROM. Download progress is shown on the screen. The download procedure can take a few minutes!

8. The controller is now ready for operation.

#### **Important:**

If the download procedure is interrupted, the operating system installation has to be started again!

#### **1.2.4.2 How can I work without hardware?**

B&R Automation Studio™ is a hardware-oriented programming system. That means the hardware has to be defined when creating a project. We differentiate between automatic hardware recognition and manual hardware definition.

If you want to work without hardware, hardware must be **defined manually** when the project is created.

##### **1.2.4.2.1 Defining additional modules**

After creating a project, additional modules can be defined on the left side of the project window (e.g. digital or analog I/O modules). Proceed as follows to do so:

- Place the cursor in the hardware configuration at the position where a new module should be inserted.
- Insert a new module by...

... Clicking on **Module...** in the **Insert** menu, or

... Clicking on the **Insert...** command in the shortcut menu (right mouse click), or

... Pressing the INS key.

- Select the desired module in the module selection window.
- Confirm the selection by clicking on **OK**.

You can create any hardware configuration by simply repeating this procedures for all desired modules. Then you can declare variables, create objects, etc. in the usual way.

**Please note:**

Be aware that you can program a project without hardware, but you can't test it!



### 1.2.4.2.2 Manual hardware definition

A detailed description of the steps required for manual hardware definition can be found in the [Creating a new project](#) section.

Of course, if there is no target system hardware present, then a connection cannot be made between the programming device and the target system. In other words, the programming device remains OFFLINE.

### 1.2.4.3 Kann ich mehrere B&R Automation Studio Versionen installieren?

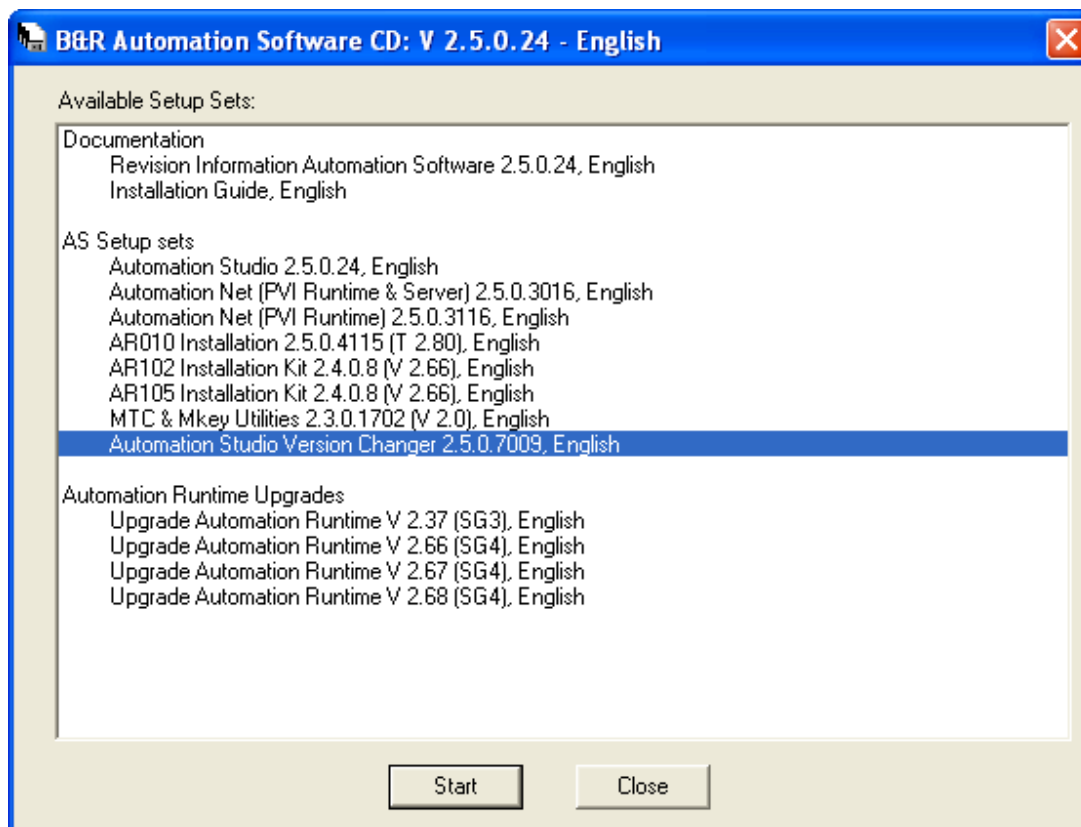
Multiple installations of B&R Automation Studio have been possible since V2.2. Changing to different versions takes place by using the Version Changer. However, only one version at a time can be active.

Older B&R Automation Studio versions have had their installation programs modified (as far back as V1.4.1) so that they can also be installed alongside newer versions.

#### 1.2.4.3.1 Installing the Version Changer

The Version Changer must be installed on your system before it can be selected from the Start Menu. To do this, run "BrMenu.exe" from the root directory of your language-specific installation CD.

The following dialog box opens:



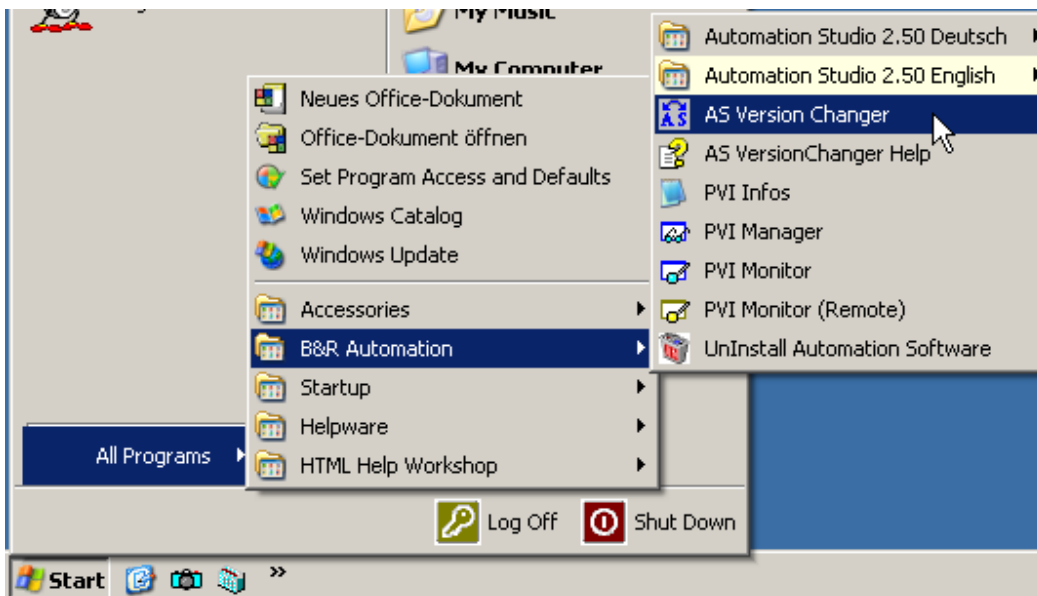
Select "Automation Studio Version Changer" and begin the installation procedure by clicking on the **Start** button.

Menus will lead you through the installation process. If installation is successful, then the Version Changer is displayed under 'B&R Automation' in the Start Menu.

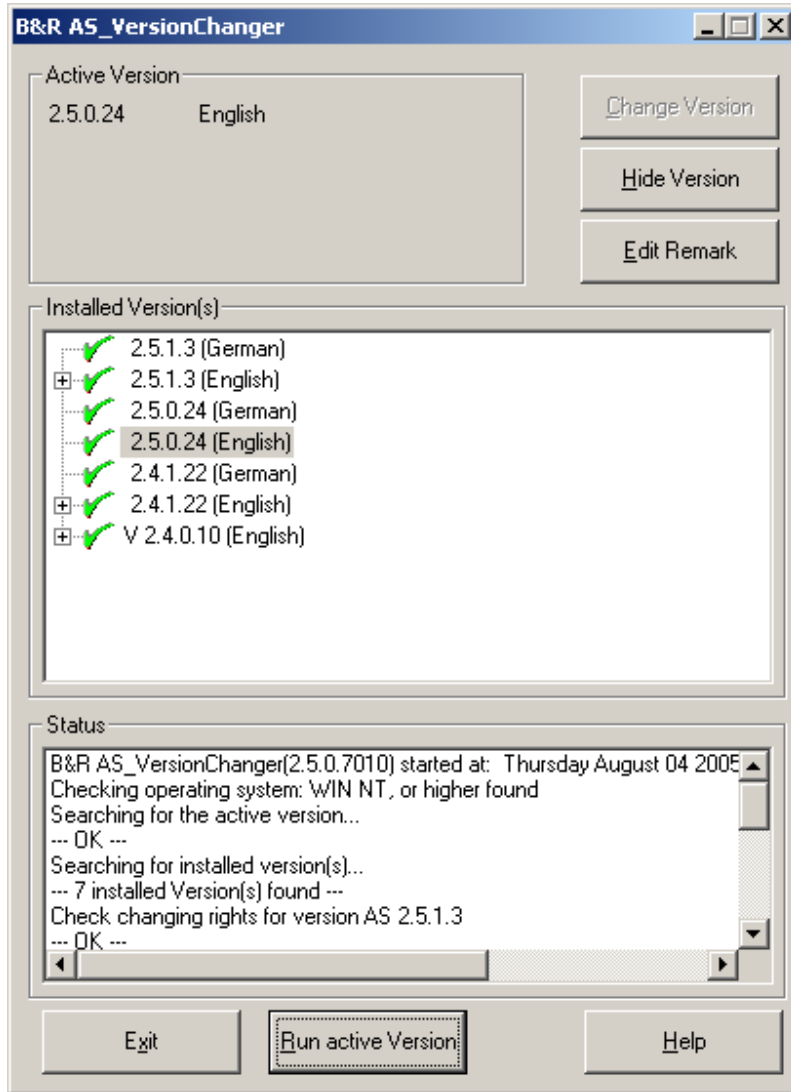
### 1.2.4.3.2 Changing AS versions

If several versions are installed at one time, the Version Changer can be used to switch between them.

Start the Version Changer from the Start menu:



The dialog box which appears displays the active version by highlighting it. The up/down cursor keys can be used to highlight the version which should now be used. Clicking the **Change Version** button activates the highlighted version. This process can take a few seconds.



The **Run active version** button starts the currently active version and **Exit** closes this window. Additional information about the version changer can be found in its [help documentation](#).

## 1.3 Shortcut keys

### 1.3.1 General

<b>Shortcut keys</b>	<b>Description</b>
+/-	Trees can be expanded and collapsed using the + and - keys on the number pad. As an alternative to number pad keys, CTRL+- and CTRL++ can be used on the normal keyboard.
ENTER	Opens the respective editor in the software configuration or closes the dialog box using the highlighted button (usually OK).
ALT+ENTER	With Alt+ENTER, you can enter the respective properties dialog box.
ESC	Cancels entries and dialog boxes.
F1	Opens context-sensitive help.
F4	Opens selection boxes (e.g. selection box for task class in the properties dialog box for tasks).
CTRL+F4	Closes current editor window.

ALT+F4	Closes B&R Automation Studio™.
F6	Switches between left and right panes of the project window.
CTRL+F6	Switches between the individual editor windows, an alternative to CTRL+TAB.
SHIFT+CTRL+F6	Like Ctrl+F6 but in the other direction, an alternative to SHIFT+CTRL+TAB.
SHIFT+F10	Opens the respective shortcut menu (like the right mouse button), the shortcut key can also be used (to the left of the rightmost CTRL key on a Windows keyboard).
ALT+SPACEBAR	Opens the AS window system menu.
ALT+-	Opens the system menu in the respective editor window.
INS	Inserts a new element in a table. The new element is inserted above the current cursor position.
CTRL+INS	If several hierarchies of elements exist in a table or an editor (main and sub-elements), the main elements are to be created with CTRL+INS and the sub-elements with INS. If the cursor is on a main element, the sub-element is inserted as the last element in the list below the main element.
DEL	Deletes the currently marked object.
SPACEBAR	With the spacebar, the currently marked cell in a table goes into input mode. In a cell that is already in entry mode, the SPACEBAR is used to go to selection dialog boxes (if available).
CTRL+D	Disables the object currently marked by the cursor.
CTRL+R	Target memory of the software object is User RAM.
CTRL+U	Target memory of the software object is User ROM.
CTRL+I	Target memory of the software object is FIXRAM.
CTRL+E	Target memory of the software object is a MEMCard.
CTRL+T	Target memory of the software object is SYSTEM ROM.
CTRL+X	Cuts the currently marked block and places it on the clipboard.
CTRL+C	Copies the currently marked block to the clipboard.
CTRL+V	Pastes a block from the clipboard.
CTRL+Z	Undo the last action.
CTRL+Y	Redo the last undo action.
CTRL+S	Saves.
CTRL+P	Prints.
CTRL+F	Calls the Find dialog box.
CTRL+H	Calls the Replace dialog box.
	Calls the Goto function.

CTRL+G	
CTRL+W	Opens the PV Monitor.

### 1.3.2 Desktop

<i>Shortcut keys</i>	<i>Description</i>
CTRL+O	Opens a project.
CTRL+N	Creates a new project.
CTRL+M	Activates monitor mode.
ALT+0	Switches to the project window.
ALT+1	Places focus in the message window (under the main window).
F4	Jumps to the next message in the message window. Either the next error message or the next result from <b>Edit / Find in Files</b> depending on which tab is active in the message window.
SHIFT+F4	Jumps to the previous message in the message window.

### 1.3.3 Text editors

<i>Shortcut keys</i>	<i>Description</i>
F3	Continues searching forward.
SHIFT+F3	Continues searching backwards.

### 1.3.4 Build & transfer

<i>Shortcut keys</i>	<i>Description</i>
STRG+F5	Transfer to PLC.
F7	Build
CTRL+F7	Build All

### 1.3.5 Debugger

<i>Shortcut keys</i>	<i>Description</i>
F5	Continue
SHIFT+F5	Cancels Debugger, allows task to continue.
F9	Set breakpoint.
SHIFT+F9	Show breakpoints

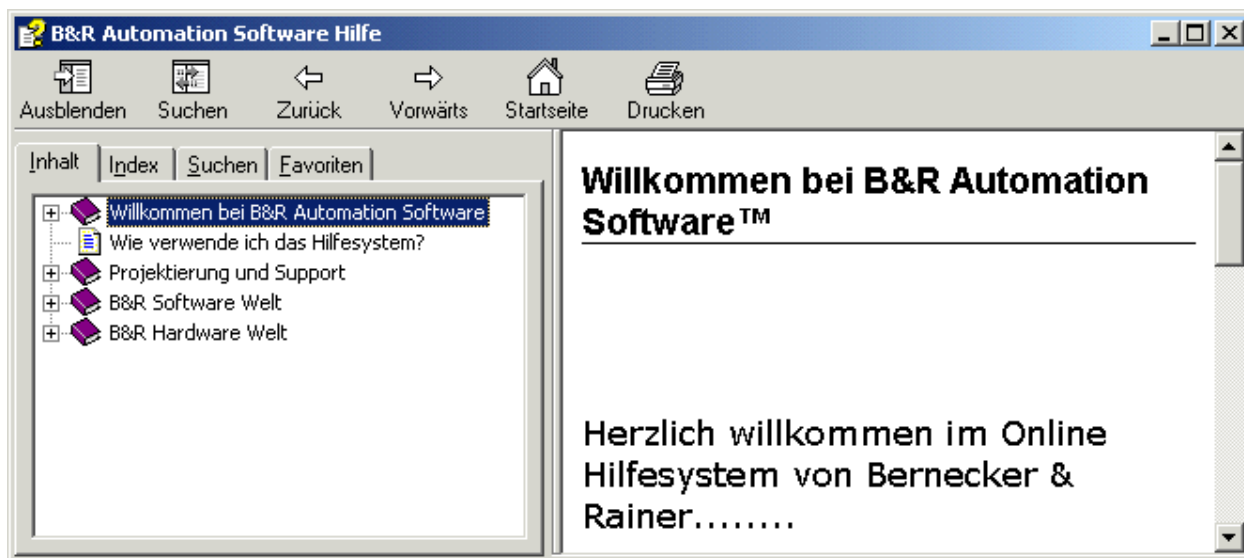
F11	Step into
F10	Step over







### 1.3.6 LAD editor

<b><i>Shortcut keys</i></b>	<b><i>Description</i></b>
Shortcut keys for the LD editor can be found in the tooltips on the toolbar or next to items in the menus.	

## 2 How do I use the help system?

### 2.1 Online help window – General




 Ausblenden	Hides/Shows the tabs on the left.	 Vorwärts	Switches to the previously viewed page if the "Back" button has been pressed.
 Suchen	Synchronizes the currently displayed help page with the navigation bar.	 Startseite	Switches back to the first page of the online help.
 Zurück	Switches to the page previously viewed.	 Drucken	Offers the following possibilities: <ul style="list-style-type: none"> <li>• Printing a selected topic</li> <li>• Printing a selected topic with all of its subtopics</li> </ul>

### 2.2 You can find information in the online help system in one of the following ways:

#### 1. Context-sensitive help

The context-sensitive help system in Automation Studio can be opened at any time. Pressing the F1 key calls up corresponding help about the current context (LD, hardware configuration, hardware modules, etc.).

#### 2. Help for menus, toolbars, etc.

You can obtain help about any specific element in the user interface. After pressing Shift+F1, a question mark appears next to the mouse pointer: . Then click on the element (menu item, toolbar, etc.) whose help should be displayed.

#### 3. Using the online help

The tabs above the table of contents can be used to select the following contents:

- ◆ **Contents (Alt-C):** Allows a topic to be selected from a section of the online help
- ◆ **Index (Alt-N):** Allows you to search for a topic using the index. The more letters that are entered, the more exact the topic selection. It is not recommended to place search terms one right after the other.
- ◆ **Search (Alt-S):** Allows a search for a topic by specifying a certain word or sentence in the online help.

◇ Fine-tuning your search:

- General tips: Since the search function only returns pages that contain all of the words in the query, simply add more words to the search terms that have already been entered.
  - You can exclude a word from your search with a hyphen ("-") placed directly before that word (a space must come before the hyphen).
  - Searching for word groups: You can search for word groups simply by adding quotation marks. Words surrounded by quotes ("like these") appear exactly this way in the texts which are returned.
- 
- ◆ **Favorites (Alt-F):** Allows favorite personal bookmarks to be saved.