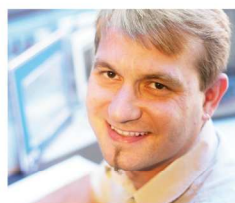


PVI OPC TM730



Perfection in Automation
www.br-automation.com



Requirements

Training modules: TM710 – PVI Communication

Software: Windows XP SP02 / Windows VISTA
PVI Development
Automation Studio 3

Hardware: PC

Table of contents

1. INTRODUCTION	4
1.1 Objective	5
2. OPC	6
2.1 OPC installation	6
3. REQUIREMENTS AND PREPARATIONS	7
3.1 Starting Automation Studio 3	7
3.2 OPC help	7
3.3 OPC terminology	8
3.4 The controller project	8
4. CREATING THE OPC CONFIGURATION	10
4.1 Creating and managing OPC tags	11
4.2 OPC alarms and OPC properties	14
4.3 OPC mapping file	15
4.4 OPC configuration summary	18
5. BR.OPC.SERVER RUNTIME SYSTEM	19
5.1 Creating the OPC configuration file .opcs	19
5.2 Runtime configuration of the OPC server	20
6. TESTING THE OPC CONFIGURATION	21
6.1 Description of the coffee machine process sequence	21
6.2 Testing environment – OPC sample client	22
7. SUMMARY	25

1. INTRODUCTION

OPC (OLE for Process Control) is the standard interface, based on the DCOM component model from Microsoft, for **accessing Windows-based SCADA packages** (Supervisory Control and Data Acquisition) on different controller systems.

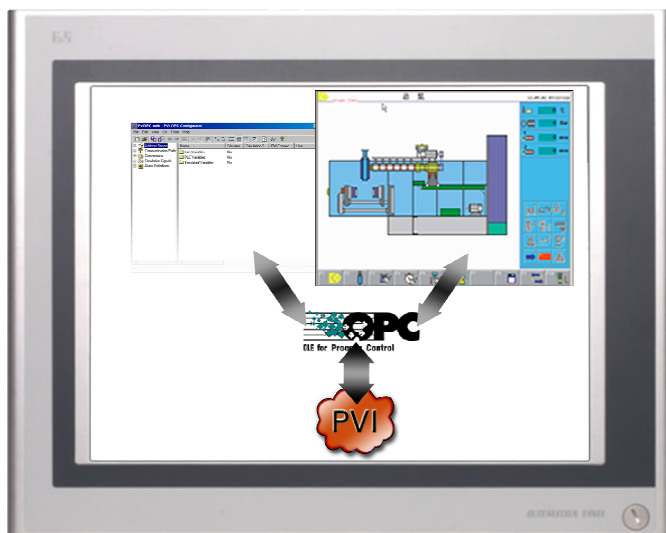


Fig. 1: BR.OPC.Server application

The OPC Server allows you to use SCADA packages from a variety of manufacturers and can also work with a number of **programming languages** including Visual Studio.NET, C++, Visual Basic and script languages.

With Automation Studio 3, creation and management of the OPC configuration is completely integrated in Automation Studio. This integration provides the following advantages:

- The OPC configuration is created together with the B&R Automation Studio project.
- For each configuration you can create and manage one or more OPC configurations.
- All OPC tags, OPC alarms and IEC variables are managed centrally in a single tool.

This training module will explain how an OPC configuration in Automation Studio 3 is created for accessing a SCADA package and will then test it using an OPC Diagnostic Client.

1.1 Objective

Participants will be able to create an OPC configuration in Automation Studio 3 after working through the samples and tasks and will know the possibilities of the OPC configuration and its access possibilities.

Access to the controller via OPC can be tested using the OPC Diagnostic Client, or any other OPC Client.

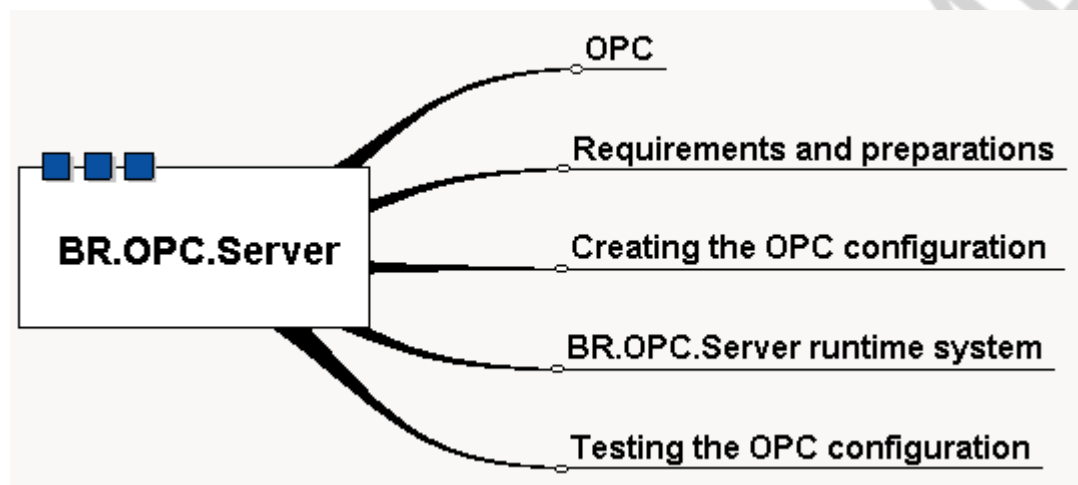


Fig. 2: Overview

With the aid of the Automation Studio help, users will get to know all of the OPC components – from the process variable to the OPC connection – and will learn how create and test an OPC configuration using an existing Automation Studio project.

2. OPC

OPC (**O**LE for **p**rocess **c**ontrol) is an industrial standard that was created with the participation of numerous worldwide leaders in automation and hardware manufacturers in cooperation with Microsoft.



Administration and specification of the OPC interface is handled by the **OPC Foundation**. This foundation is made up of members from well-known visualization and controller manufacturers.

www.opcfoundation.org

OPC is based on Microsoft's **OLE** (**O**bject **L**inking and **E**MBEDding) and **DCOM** (**D**istributed **C**omponent **O**bject **M**odel) technology and is made up of a set of standard interfaces, properties and methods that are used in the process controller and setup of automation clients.

The OLE/COM technologies determine how individual software components work together and exchange data. OPC provides a general interface for communication with various process control devices, independent of the control software used in the process.

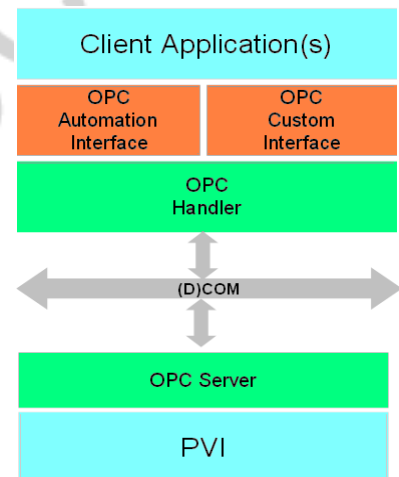


Fig. 3: PVI OPC

The standardized interface enables the user to select any SCADA package that supports OPC or to create his own OPC client based on the VC++, VB or Visual Studio.NET programming languages.

2.1 OPC installation

The PVI OPC Server DA 3.0 (BR.OPC.Server) is part of the **PVI Developer** installation from Version 2.6.0.3003 and later.

All necessary components (e.g. OPC Core Components, etc.) are installed during the installation.

3. REQUIREMENTS AND PREPARATIONS

Automation Studio provides all that is needed to create an OPC configuration.

- Creating the OPC runtime configuration with the Automation Studio project
- Creating and managing multiple OPC configurations
- Using the active hardware configuration to define the PVI connection settings

3.1 Starting Automation Studio 3

The first step is to open Automation Studio.

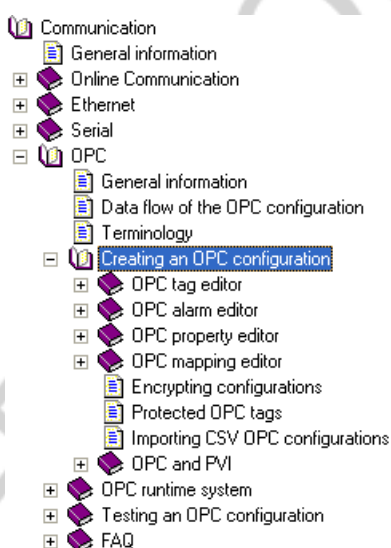
Do not close Automation Studio during the exercises and tests.

3.2 OPC help

The OPC help system is an important tool for creating an OPC configuration in Automation Studio or for manually changing entries.

Many references to the help system will be made throughout this training module because this guide is always available, most importantly while creating the configuration.

When OPC Editors are open, you can use the F1 key to open the OPC help, otherwise the Start Menu can be used at any time.



In this training manual, all references to the OPC help will be indicated using the following icon and the structure of the chapters starting with the **OPC** chapter.



The help system can remain open. Use **<ALT> + <Tab>** to switch between Automation Studio and the help system.

3.3 OPC terminology

The following names and terms are used when creating and managing OPC configurations:

OPC term	Description
OPC tag	Describes the properties of a process variable with the corresponding OPC-specific properties.
OPC alarm	Describes the properties of an alarm definition for a digital or limit alarm.
OPC custom properties	Properties or information that users can define on their own
OPC mapping	The PVI-specific properties are assigned to each OPC tag.

3.4 The controller project

The exercises in this training module are based on the "CoffeeMachine" sample project installed with the Automation Studio installation.

It contains all of the process variables required for this training module that are needed to create and to access the OPC configuration.

3.4.1 Opening the sample project

The project is opened using the menu <File – Open Project> from the directory "Program Files\BrAutomation\AS30071\Samples\CoffeeMachine.zip".

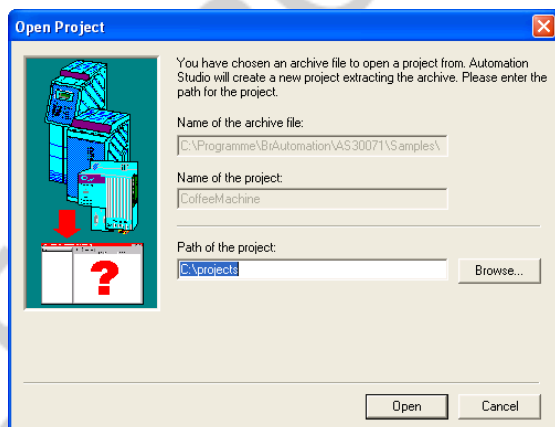


Fig. 4: Opening the "CoffeeMachine" sample project

3.4.2 Transferring the project to the target hardware

The configuration for the AR000 simulation is active after opening the project.

The process of transferring the project to the AR000 (online settings, start AR000, transfer project) is described in the Automation Studio help – Visualization section.

Help - Visual Components VC4 section:



Introduction - Sample programs

3.4.3 Testing the sample project

The sample project is tested on the AR000 simulation using the VNC Viewer or with the help of the Automation Studio variable monitor.

We recommend using the VNC Viewer because this allows testing the OPC configuration with live values.

The VNC Viewer (e.g. RealVNC) can be downloaded for free by going to:

RealVNC – URL: <http://www.realvnc.com/>



...with a little practice, this section should just take a few minutes, provided that Automation Studio is already started...

4. CREATING THE OPC CONFIGURATION

This section describes how to create the OPC configuration in Automation Studio.

The following steps are necessary for creating an OPC configuration:

- An OPC tag is created from a process variable using OPC-specific properties.
- An OPC tag is configured in the OPC Mapping with PVI-specific properties.

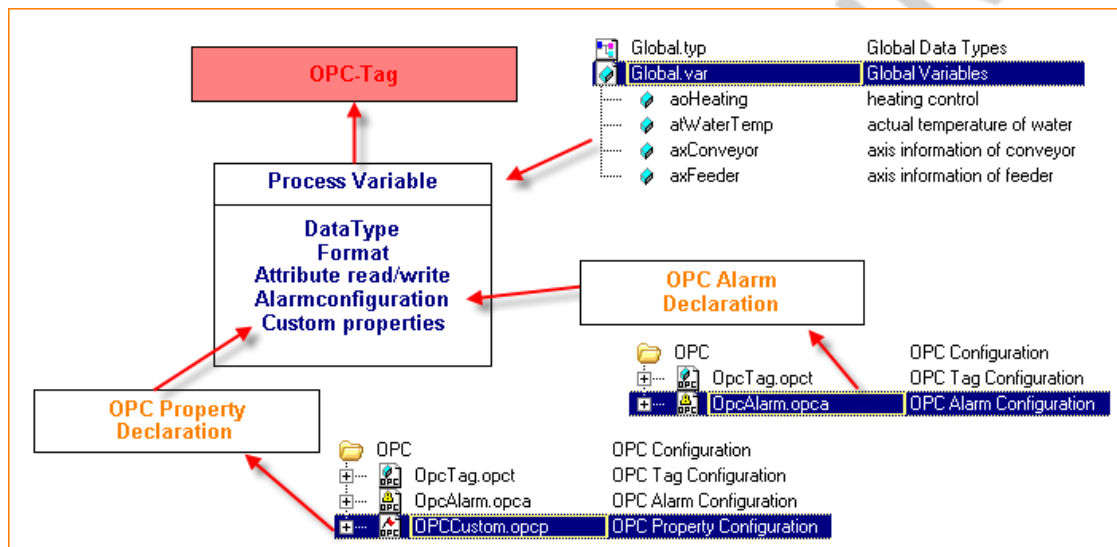


Fig. 5: Creating an OPC configuration

Help - OPC section:

Creating an OPC configuration

Create and manage the OPC configuration directly in the logical view of the Automation Studio project.

Multiple OPC configurations in different packages can be created according to the project tree - however, it is recommended to manage the OPC configuration (OPC tags, OPC alarms and OPC properties) in one package for reasons of clarity.

4.1 Creating and managing OPC tags

An OPC tag describes the properties of a process variable in the controller software, including its OPC-specific properties.

4.1.1 Process variables for the exercises

The following process variables are required for this exercise:

Action	Value range	Process variables
Select coffee type	0-2	gMainLogic.par.coffeeType
Coffee ingredients	0-100	gMainLogic.par.recipe.coffee gMainLogic.par.recipe.milk gMainLogic.par.recipe.sugar gMainLogic.par.recipe.water
Coffee price	-	gMainLogic.par.recipe.price
Payment	0-10	gMainLogic.par.givenMoney
Switching on/off	0 / 1	gMainLogic.cmd.switchOnOff
Start preparation	1	diStartCoffee
Water temperature	-	gHeating.status.actTemp
Messages	-	gMainLogic.cmd.vis.messageIndex
Process sequence	-	gMainLogic.status.progressStep

4.1.2 Create OPC tag configuration file

An OPC tag configuration file must first be made before an OPC tag can be created.

Exercise: Creating an OPC tag configuration file



Carry out the following steps to create an OPC tag configuration file:

- Create a new package in the project
- Add the OPC tag configuration file
- Add the process variables described above to the editor
- Save the configuration

An OPC tag configuration file can be created in the shortcut menu in the project's logical view - **Add Object** with the type **OPC - New OPC Tag Declaration**.

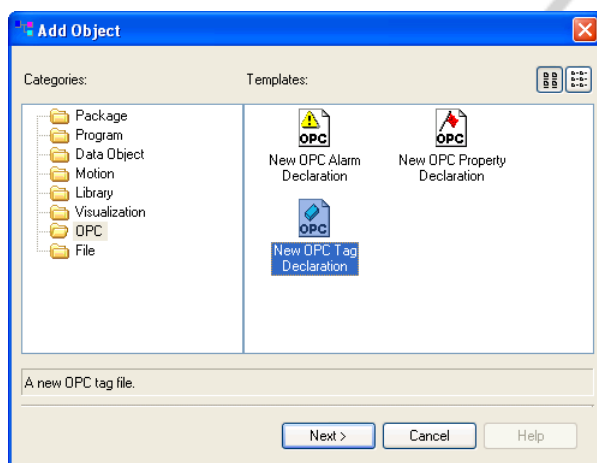


Fig. 6: Insert OPC tag configuration file

Click on **<Next>** to open the window for entering a configuration file name (e.g. **OPCTags**) and Finish to complete the process.

The OPC tag file is inserted in the logical view with the file extension opct.



Double click on the file to open it in the tag editor and add process variables.












Help – OPC section:



Create OPC configuration – OPC tag editor – Insert OPC tags

Results from the exercise:

After the necessary process variables have been added, they will appear in the configuration file as follows:

OPC Tag Name ▲	Data Type	Description
 gMainLogic.cmd.switchOnOff	VT_BOOL	
 gMainLogic.par.givenMoney	VT_R4	
 gMainLogic.par.coffeeType	VT_UI1	
 gMainLogic.cmd.vis.messageIndex	VT_UI1	
 gMainLogic.par.receipe.sugar	VT_R4	
 gMainLogic.par.receipe.milk	VT_R4	
 gMainLogic.par.receipe.price	VT_R4	
 gMainLogic.status.progressStep	VT_UI1	
 dStartCoffee	VT_BOOL	
 gMainLogic.par.receipe.coffee	VT_R4	
 gMainLogic.par.receipe.water	VT_R4	

The OPC tag declaration (or any other configuration file) must be saved before it can be used.

The OPC properties are described in the OPC help.

Help – OPC section:



Create OPC configuration – OPC tag editor – Properties of OPC tags

4.1.3 Naming OPC tags

In the last exercise we created an OPC tag from a process variable.

The OPC tag is displayed on the OPC client with the name of the folder (package).
e.g. CoffeeMachine/OPC/gMainLogic.status.progressStep.

If the logical name of the OPC tag (as it is displayed in the visualization) is different from the name of the process variable, then there are two possibilities for the configuration:

- The OPC tag name can be changed after inserting the process variable to the OPC tag editor. The link to the process variable is retained in the background. This is also displayed in the OPC mapping.
- A new OPC tag can be created with any name. The link between this OPC tag and the process variable is then made in the OPC mapping editor.

4.2 OPC alarms and OPC properties

Information about creating OPC alarms and OPC properties can be found in the OPC help.

These specific OPC configurations are not described in the training manuals.

Help - OPC section:



Create OPC configuration – OPC Alarm Editor
Create OPC configuration – OPC Property Editor

4.3 OPC mapping file

The OPC tags are assigned to a PVI process object in the OPC mapping configuration.

- Mapping multiple tags to a PVI process object
- Make PVI specific settings
- Set up special PVI objects

This assignment enables a detailed distribution of the OPC configuration to a runtime system.

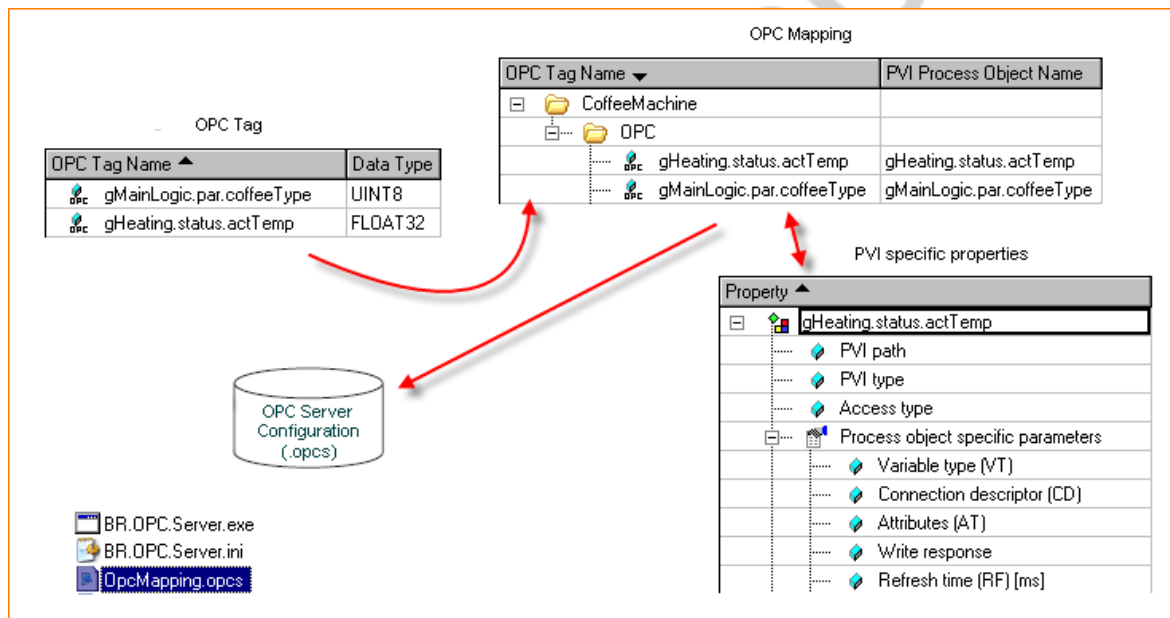


Fig. 7: OPC mapping overview

The connection settings between the runtime system and the controller are handled automatically by Automation Studio – Online Settings.

In Automation Studio 3.0.80 and later, detailed configuration of the connection is possible in the OPC mapping file.

4.3.1 Creating the OPC mapping file

The OPC mapping file can be created using the shortcut menu in the configuration view of the active project - **Add Object** of type **OPC - New OPC Mapping Declaration**.

Help – OPC section:

Create OPC configuration – OPC Mapping Editor

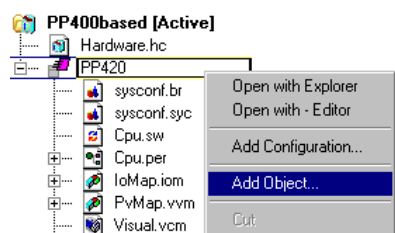


Fig. 8: Inserting an OPC Mapping

Click **<Next>** in the subsequent window to enter the name of the OPC mapping file (e.g. OPC mapping) and click **<Finish>** to close the window.

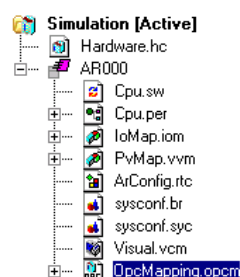


Fig. 9: Inserted mapping file

The OPC mapping file is added with the file extension .opcm in the configuration view of the active configuration.

Double click on the file to open it in the mapping editor and add the OPC tags.

Exercise: Creating an OPC mapping file



Carry out the following steps to create an OPC mapping file:

- Switch to the configuration view in Automation Studio
- Add the OPC mapping file
- Add the OPC tags (via drag & drop)
- Save the mapping file

Results from the exercise:

The file should appear as follows after the OPC tags have been inserted to the OPC mapping:

OPC Tag Name	PVI Process Object Name
CoffeeMachine	
OPC	
gMainLogic.status.progressStep	gMainLogic.status.progressStep
gMainLogic.par.receipe.water	gMainLogic.par.receipe.water
gMainLogic.par.receipe.sugar	gMainLogic.par.receipe.sugar
gMainLogic.par.receipe.price	gMainLogic.par.receipe.price
gMainLogic.par.receipe.milk	gMainLogic.par.receipe.milk
gMainLogic.par.receipe.coffee	gMainLogic.par.receipe.coffee
gMainLogic.par.givenMoney	gMainLogic.par.givenMoney
gMainLogic.par.coffeeType	gMainLogic.par.coffeeType
gMainLogic.cmd.vis.messageIn...	gMainLogic.cmd.vis.messageIndex
gMainLogic.cmd.switchOnOff	gMainLogic.cmd.switchOnOff
diStartCoffee	diStartCoffee

4.4 OPC configuration summary

An OPC tag is the basis for the OPC configuration. The OPC tag consists of a process variable and has additional OPC properties.

OPC alarms and OPC properties are linked together as one of these OPC properties.

OPC tags can be created by simply dragging & dropping from the process variable declarations.

One or more OPC mapping files can be created for each configuration of the Automation Studio project (e.g. for different connection settings or tag configurations).

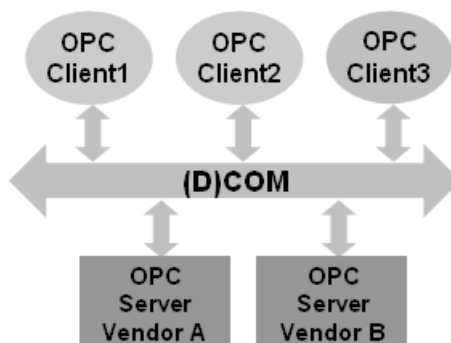
OPC tags are added to the mapping by simply dragging & dropping.

The connection settings for PVI communication are automatically applied from the Automation Studio – Online Settings.

The OPC configuration (.opcs file), which is required for the OPC server, is automatically created when making the build for the active project.

5. BR.OPC.SERVER RUNTIME SYSTEM

The B&R PVI OPC Server DA 3.0 (BR.OPC.Server) is an OPC-compatible server that can communicate with various I/O devices and protocols as well as send data to OPC Clients when using PVI.



Help – OPC section:

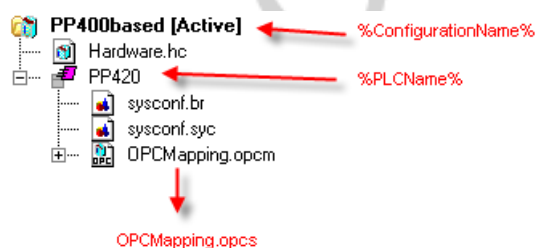


OPC runtime system – Runtime behavior of the OPC server

5.1 Creating the OPC configuration file .opcs

After successfully making the project build, the OPC configuration file created for the target system in the training module is generated with the name of the mapping file and the file extension .opcs.

The file is created in the following directory:



%ProjectPath&\Temp\Objects\%**ConfigurationName**%\%**PLCName**%\MappingName.opcs

Help – OPC section:



OPC runtime system – Structure of the OPCS configuration file

5.2 Runtime configuration of the OPC server

Runtime system configuration requires the following:

- Install PVI Development with the BR OPC Server 3.0 on the target system
- Copy the created .opcs configuration file to the target system
- Adjust the **BR.OPC.Server.ini** file with the cross-reference to the .opcs configuration file

Help – OPC section:



OPC runtime system – OPC server INI file

Exercise: Configuring the target system and adjusting the INI file



In this exercise, the generated file **OPCMapping.opcs** will be copied from the Build directory to the BIN directory and the file **BR.OPC.Server.ini** will be adjusted according to the entry **CfgFile0 =**.

Results from the exercise:

The created configuration can now be accessed from the OPC application and communication between the controller and application can be tested.

6. TESTING THE OPC CONFIGURATION

In order to test the OPC configuration created in this training module, either the OPC Diagnostic Client installed during the PVI setup or a different OPC test client must be used.

Help – OPC section:

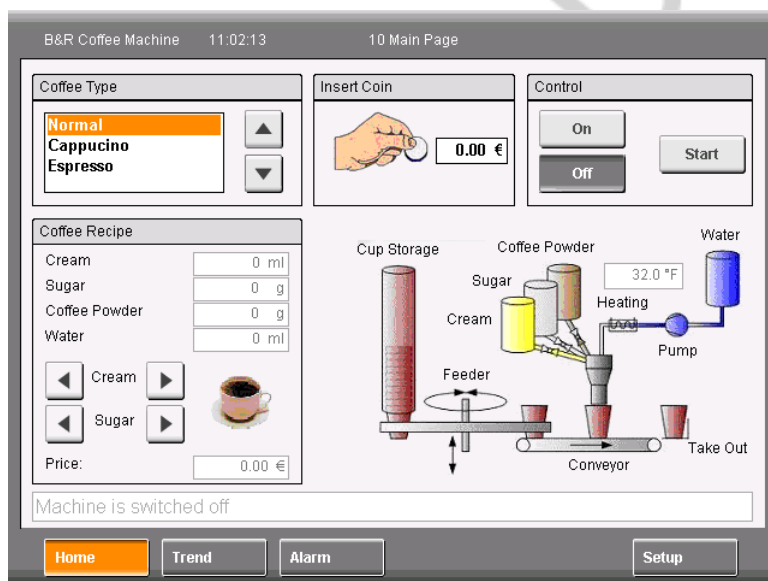


Testing an OPC configuration

The goal of this exercise is to operate the coffee machine not only via VNC, but also via OPC.

6.1 Description of the coffee machine process sequence

The coffee machine is operated using the process variables specified in 4.1.



The VNC Viewer can be used to monitor operation of the coffee machine via the OPC application. The variable monitor in Automation Studio can also be used to do this.

After starting the coffee machine (gMainLogic.cmd.switchOnOff), the water is then heated up (gHeating.status.actTemp).

After a certain temperature has been reached, preparation of the selected coffee type can begin (gMainLogic.par.coffeeType).

Preparation is then started (diStartCoffee) after simulating the insertion of a coin (gMainLogic.par.givenMoney) either equal to or greater than the coffee price (gMainLogic.par.recipe.price).

The status of the process sequence is output via status process variables (gMainLogic.cmd.vis.messageIndex / gMainLogic.status.progressStep).

The ingredients (i.e. the coffee recipe) can be pre-selected with the coffee type (gMainLogic.par.coffeeType). Sugar (gMainLogic.par.recipe.sugar) and milk (gMainLogic.par.recipe.milk) values can be modified.

6.2 Testing environment – OPC sample client

In principle, any OPC client (application) can be used to test the coffee machine.

The **OPC sample client** will be used in this training manual.

6.2.1 Start the sample client and connect to the OPC server

After starting the OPC sample client a connection to the installed OPC server is established by pressing the **<Connect>** button.

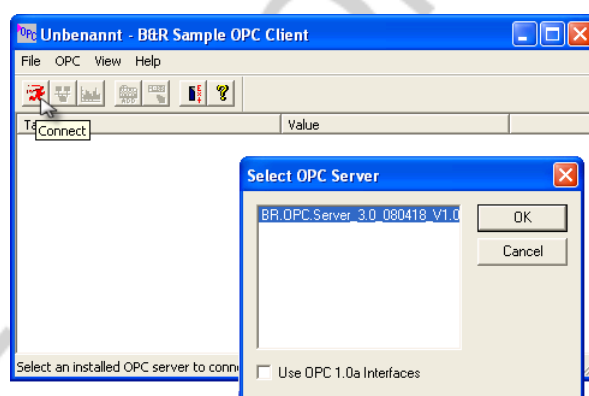


Fig. 10: Establishing connection to the OPC server

6.2.2 Inserting the OPC tags

The <Add Item> button is used to display all of the OPC tags in the runtime configuration in the subsequent dialog box.

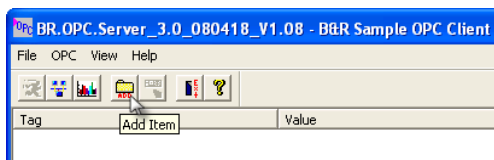


Fig. 11: Inserting OPC tags

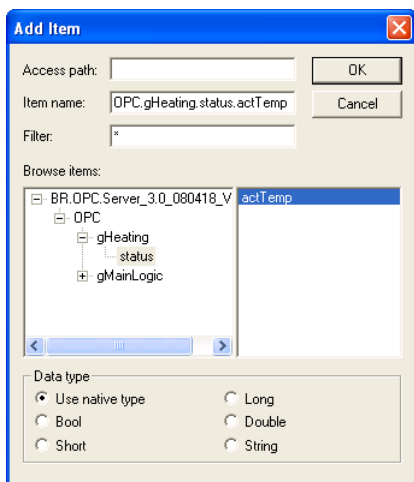


Fig. 12: Add Item dialog box

Task: Inserting all OPC tags in the sample client

Insert all OPC tags from the OPC configuration into the sample client.

Tag	Value
OPC.gHeating.status.actTemp	0
OPC.gMainLogic.cmd.vis.messageIndex	0
OPC.gMainLogic.par.receipe.coffee	60
OPC.gMainLogic.par.receipe.milk	100
OPC.gMainLogic.par.receipe.price	1.69
OPC.gMainLogic.par.receipe.sugar	30
OPC.gMainLogic.par.receipe.water	150
OPC.gMainLogic.cmd.switchOnOff	Off
OPC.gMainLogic.status.progressStep	0
OPC.dStartCoffee	Off
OPC.gMainLogic.par.coffeeType	0
OPC.gMainLogic.par.givenMoney	0

6.2.3 Operating the coffee machine

The coffee machine is operated using the <Write Item> button.

One item in the list must be selected to which a value can be written.

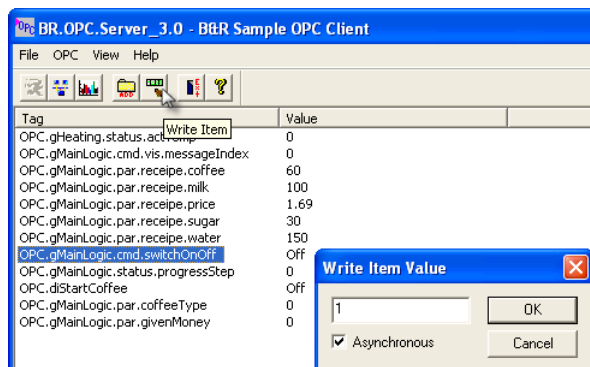
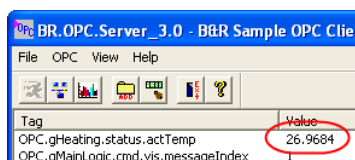


Fig. 13: Writing to an OPC tag



Writing the value 1 to the item **OPC.gMainLogic.cmd.switchOnOff** in this example will turn on the coffee machine. The water will then begin to be heated.

The item OPC.gMainLogic.cmd.vis.messageIndex shows the current status of the heating procedure. 0 = Turned off, 1 = Water temperature not yet reached, 2 = Water temperature reached.

After writing a value greater than or equal to the item OPC.gMainLogic.par.recipe.price to the variable OPC.gMainLogic.par.givenMoney, the brewing process can then begin by writing the value 1 to the item OPC.diStartCoffee.

The item gMainLogic.status.progressStep indicates the status of the brewing process. 0 = Cup being transferred to the brewing device, 1 = Preparation in progress, 2 = Finished!

The procedure can now be repeated.

7. SUMMARY

The BR OPC Server 3.0 provides the user with extensive access to the world of B&R control systems.

The possibility for the user to freely select finished **SCADA** visualizations for Windows means that there are essentially no limitations for accessing a controller's variables.

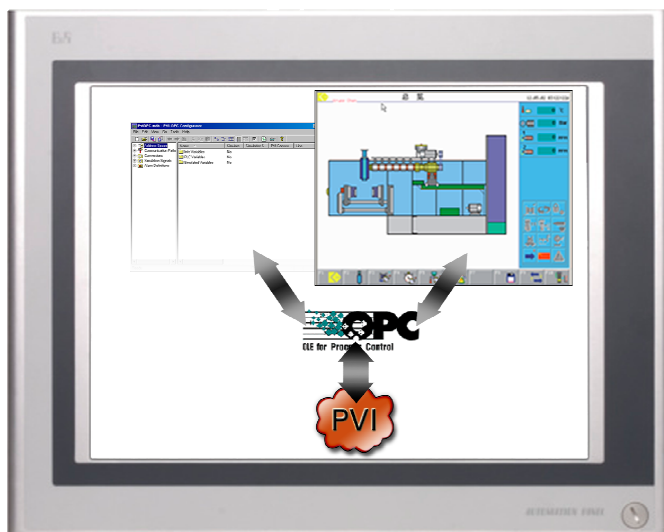


Fig. 14: BR.OPC.Server application

If the possibilities offered in a visualization are not sufficient, then the user still has the option to program the OPC Server (i.e. access to functions, methods and events via the standardized interface).

As a result, it should be possible to meet the demands of the OPC visualization for any application.

Notes

ELECTRONIC DOCUMENT

Overview of training modules

TM200 – B&R Company Presentation **
TM201 – B&R Product Spectrum **
TM210 – The Basics of Automation Studio
TM211 – Automation Studio Online Communication
TM212 – Automation Target **
TM213 – Automation Runtime
TM220 – The Service Technician on the Job
TM223 – Automation Studio Diagnostics
TM230 – Structured Software Generation
TM240 – Ladder Diagram (LAD)
TM241 – Function Block Diagram (FBD)
TM246 – Structured Text (ST)
TM247 – Automation Basic (AB)
TM248 – ANSI C
TM250 – Memory Management and Data Storage
TM260 – Automation Studio Libraries I
TM261 – Closed Loop Control with LOOPCONR

TM400 – The Basics of Motion Control
TM410 – The Basics of ASiM
TM440 – ASiM Basic Functions
TM441 – ASiM Multi-Axis Functions
TM445 – ACOPOS ACP10 Software
TM446 – ACOPOS Smart Process Technology
TM450 – ACOPOS Control Concept and Adjustment
TM460 – Starting up Motors
TM480 – Hydraulic Drive Control

TM500 – The Basics of Integrated Safety Technology
TM510 – ASiS SafeDESIGNER

TM600 – The Basics of Visualization
TM610 – The Basics of ASiV
TM630 – Visualization Programming Guide
TM640 – ASiV Alarm System
TM650 – ASiV Internationalization
TM660 – ASiV Remote
TM670 – ASiV Advanced

TM700 – Automation Net PVI
TM710 – PVI Communication
TM711 – PVI DLL Programming
TM712 – PVI Services
TM730 – PVI OPC

TM800 – APROL System Concept
TM810 – APROL Setup, Configuration and Recovery
TM811 – APROL Runtime System
TM812 – APROL Operator Management
TM813 – APROL XML Queries and Audit Trail
TM830 – APROL Project Engineering
TM840 – APROL Parameter Management and Recipes
TM850 – APROL Controller Configuration and INA
TM860 – APROL Library Engineering
TM865 – APROL Library Guide Book
TM870 – APROL Python Programming
TM890 – The Basics of LINUX

**) see Product Catalog

CORPORATE HEADQUARTERS

Bernecker + Rainer Industrie-Elektronik Ges.m.b.H.

B&R Strasse 1

5142 Eggelsberg

Austria

Tel.: +43 (0) 77 48/65 86 - 0

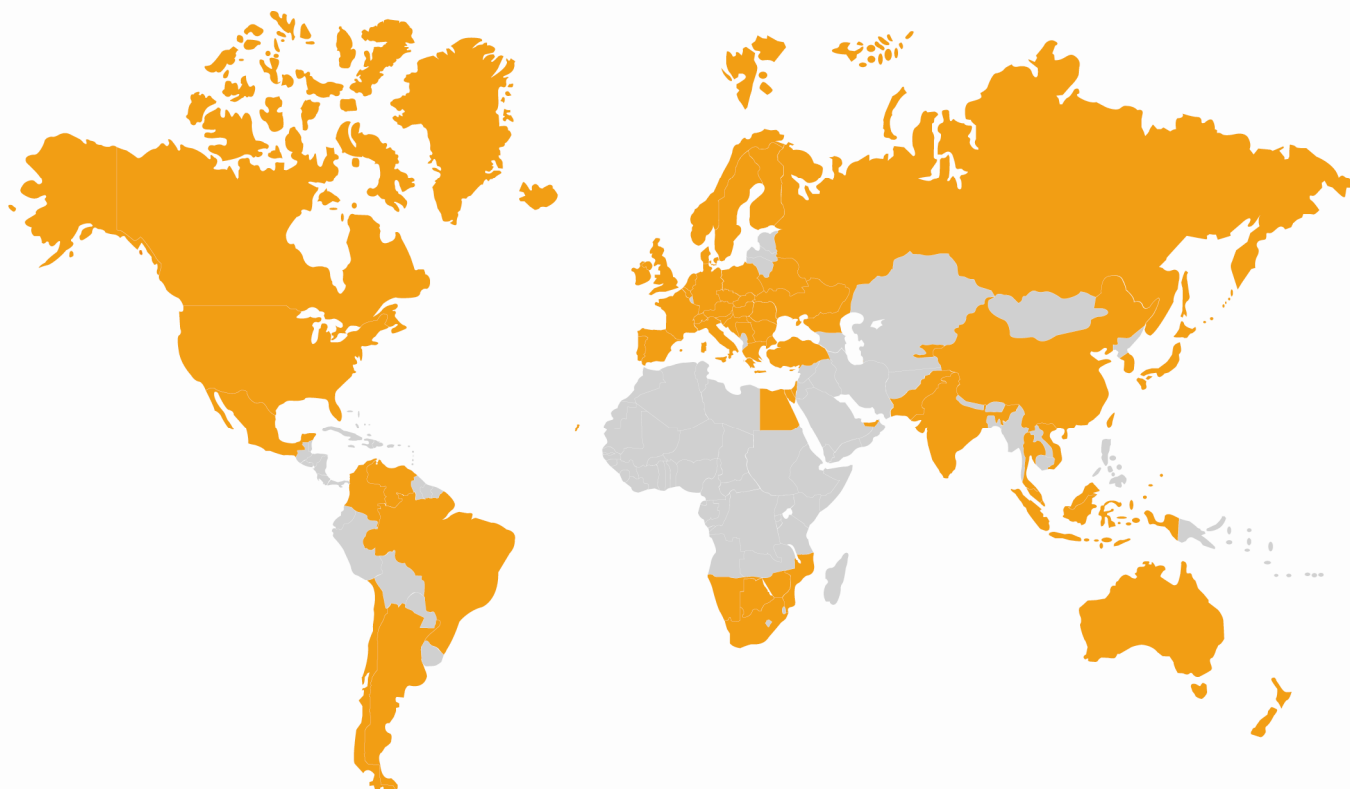
Fax: +43 (0) 77 48/65 86 - 26

info@br-automation.com

www.br-automation.com

TM730TRE-30-ENG 0109
©2009 by B&R. All rights reserved.
All registered trademarks are the property of their respective owners.
We reserve the right to make technical changes.

155 offices in 60 countries - www.br-automation.com/contact



Argentina • Australia • Austria • Belarus • Belgium • Botswana • Brazil • Bulgaria • Canada • Chile • China • Colombia • Croatia • Cyprus
Czech Republic • Denmark • Egypt • Emirates • Finland • France • Germany • Greece • Hungary • India • Indonesia • Ireland
Israel • Italy • Japan • Korea • Luxemburg • Colombia • Kyrgyzstan • Malaysia • Mexico • Mozambique • Namibia • The Netherlands
New Zealand • Norway • Pakistan • Poland • Portugal • Romania • Russia • Serbia • Singapore • Slovakia • Slovenia • South Africa
Spain • Sweden • Switzerland • Taiwan • Thailand • Turkey • Ukraine • United Kingdom • USA • Venezuela • Vietnam • Zimbabwe