OPC-Server Manual

E	5.99005.03

Edition: July 2001

BAUMÜLLER

MANUAL OPC SERVER

Edition: July 2001

Document no. 5.99005.03

BEFORE CARRYING OUT COMMISSIONING, CAREFULLY READ AND OBSERVE THE OPERATING INSTRUCTIONS AND SAFETY INFORMATION

This document contains all the information necessary to correctly use the products it describes. It is intended for specially trained, technically qualified personnel who are well-versed in all warnings and commissioning activities.

The equipment is manufactured using state-of-the-art technology and is safe in operation. It can safely be installed and commissioned and functions without problems if the safety information is followed.

You may not carry out commissioning until it has been established that the machine into which this component is to be installed complies with the specifications of the EC machine guidelines.

This technical description/these operating instructions invalidate all previous descriptions of the corresponding product. Within the scope of further development of our products, Baumüller GmbH reserves the right to change their technical data and handling.

Manufacturer and Supplier's Address:
Baumüller Nürnberg GmbH
Ostendstr. 80
D-90482 Nürnberg
Tel. ++49 (0)9 11/54 32 - 0
Fax ++49 (0)9 11/54 32 - 1 30

Country of Origin: Germany

Date of Manufacture: Determined from the serial number on the equipment

TABLE OF CONTENTS

1 INTRODUCTION

1.1 What is the OPC Server?

The OPC Server was especially designed to enable the communication between any OPC Client (e.g. a visualization) and your PLC. It allows an OPC Client to get current values from your PLC which can be used for example to visualize the running control processes. The OPC Server is mainly working in the background and can be started either automatically by an OPC Client or manually.

The OPC Server offers the possibility to load or reimport any PROPROG project and to display the current values of variables stored in the CSV file of any inserted project.

NOTE

Only variables stored in the CSV file of a project can be used by an OPC Client to visualize the running control processes. This requires that the CSV check boxes in the resource setting dialog of the programming system are set otherwise the variables are not stored in the CSV file.

The OPC Server is delivered together with a Test Client program. This Test Client can be used to check the correct function of the OPC Server.

1.2 What kind of documentation do you get for the OPC Server?

The documentation for the OPC Server consists of several parts: the OPC Server Manual and the OPC Server Help.

The OPC Server Manual is divided into the following chapters:

- Introduction - gives you general information about the OPC Server and its features.
- Getting started with the OPC Server - describes the handling and the user interface of the OPC Server.
- How to use the OPC Server - describes the required steps for the correct usage of the OPC Server.

The OPC Server Help which can be called selecting the menu item 'Contents' in the submenu '?' provides detailed information for all parts in the OPC Server. It contains a detailed description of:

- all submenus and the available menu items as well as the associated shortcut list.
- all icons displayed in the toolbar.
- the dialogs 'Resource Properties' and 'Server Status'.
- the possible error messages in the OPC Server.

1.3 Symbols and textual conventions

The following symbols are used in this manual:

∗ is used for enumeration.

• is used for an operation which has to be done.

◊ is used for an operation which is optional.

🖱 is used for a sequence of operations to be done with the mouse.

In the procedures described in this manual the instructions 'click' and 'double click' relate to the **left** mouse button. If the right mouse button is meant (e.g. to open an object context menu) this is explicitly mentioned.

⌨ is used for a sequence of operations to be done with the keyboard.

ℹ NOTE

is used to provide important information.

📖 is used to introduce references to other documents or chapters of this manual.

The following textual conventions have been set up for this manual:

' commas are used for names of icons, menu items or proper names of objects e.g. menu item 'Save'.

<ALT> brackets are used for the name of keys on your keyboard and for words you have to enter.

<ALT> + <F4> is used if you have to press two keys at the same time.

editor name Italic letters are used as place holders for proper names.

2 GETTING STARTED WITH THE OPC SERVER

2.1 System requirements

Hardware requirements

To run the OPC Server, the following hardware requirements must be at least fulfilled:

- IBM compatible PC with a 486 processor
- VGA monitor
- 16 MB of RAM
- Hard disk with at least 2 MB free memory space

Software requirements

To run the OPC Server, the following software requirements must be at least fulfilled:

- Microsoft Windows® 95 with DCOM or Microsoft Windows® NT 4.0
- A monitor SVGA resolution 800x600 is recommended

2.2 Installing the OPC Server

The installation program is a program which performs the actions necessary for installation by itself. The simplest method of installation is to use the Explorer to run the installation program 'setup.exe'. The installation includes several dialogs where you have to do the corresponding entries.

After successful installation, you will find the OPC Server program group in the Windows® program menu. During installation you will be asked whether you also want to install the OPC Server icons on your desktop.

2.3 Starting the OPC Server

NOTE

Before starting the OPC Server, ensure that the CSV check boxes in resource setting dialog of your PROPROG project are set. The OPC Server requires the sr.csv file for correct operation.

There are two possibilities to start the OPC Server:

- Start your OPC Client program and connect it to the OPC Server. The OPC Server is automatically started.
- Click the Windows menu 'Start' and choose 'Program'. Click on the program name of the OPC Server application. You can also start the OPC Server by double clicking on the corresponding icon on your desktop.

If you call the OPC Server the first time, the workspace 'Untitled' is opened. In this case you have to save the empty workspace under a new file name.

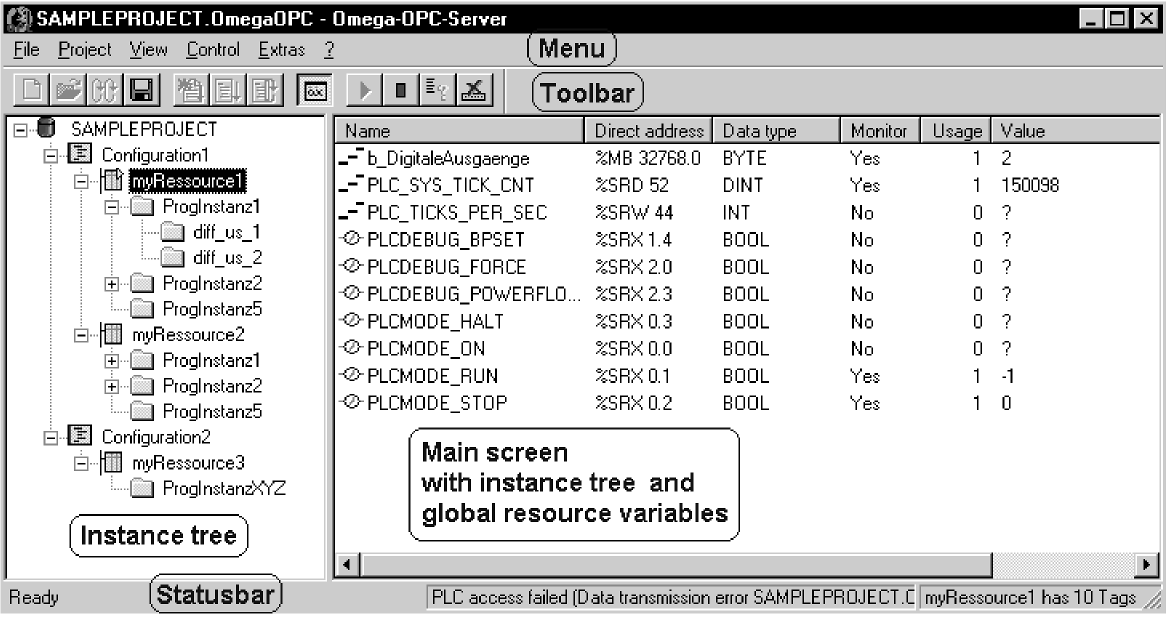
2.4 Using mouse and keyboard

The OPC Server supports full use of the mouse or the keyboard. For beginners it may be easier to start working with the mouse because it is not necessary to learn the keyboard shortcuts. In rough industrial environments the keyboard may be more appropriate.

In the following sections we describe both.

2.5 User interface

After starting the OPC Server a welcome dialog is displayed followed by the user interface.



Name	Direct address	Data type	Monitor	Usage	Value
b_DigitaleAusgaenge	%MB 32768.0	BYTE	Yes	1	2
PLC_SYS_TICK_CNT	%SRD 52	DINT	Yes	1	150098
PLC_TICKS_PER_SEC	%SRW 44	INT	No	0	?
PLCDEBUG_BPSET	%SRX 1.4	BOOL	No	0	?
PLCDEBUG_FORCE	%SRX 2.0	BOOL	No	0	?
PLCDEBUG_POWERFLO...	%SRX 2.3	BOOL	No	0	?
PLCMODE_HALT	%SRX 0.3	BOOL	No	0	?
PLCMODE_ON	%SRX 0.0	BOOL	No	0	?
PLCMODE_RUN	%SRX 0.1	BOOL	Yes	1	-1
PLCMODE_STOP	%SRX 0.2	BOOL	Yes	1	0

Ready

PLC access failed (Data transmission error SAMPLEPROJECT.C

myRessource1 has 10 Tags

Figure 2-1: User interface with opened workspace and global resource variables displayed

The user interface of the OPC Server consists basically of four parts: Menu, toolbar, main screen and status bar.

The title bar at the top of the user interface displays the name of the active workspace. If you have called the OPC Server the first time or you have created a new workspace the workspace is named 'Untitled'. In this case you have to save the new workspace under the name you want to use it.

The procedure how to save the active workspace is described in the chapter 'How to use the OPC Server'.

Menu

The menu is represented in the line below the title bar. It contains the following submenus:

- The submenu 'File' can be used to create a new workspace or to open an existing workspace. There are also commands available to save the active workspace and to reimport all PROPROG projects already inserted into your workspace. It contains also the command to exit the OPC Server.
- The submenu 'Project' contains all commands to manage the PROPROG projects available in the OPC Server. The menu items can be used to insert an existing project, to import an existing project (i.e. to replace a selected project in the instance tree), to reimport or to delete a particular project. Furthermore this submenu provides the menu item 'Properties...' used to edit the resource properties.

Online help

The dialog 'Resource Properties' is described in detail in the OPC Server Help.

- The submenu 'View' can be used to design the appearance of the user interface. It provides the menu item 'Monitor' which can be used to display the current usage and value of global resource variables and instance variables. There is also a menu item available to show or hide the status bar at the bottom of the user interface.
- The submenu 'Control' contains all commands to start and stop the data transfer to your PLC. It also offers a command to monitor individual global resource variables and instance variables. Furthermore the command 'Get status' can be used to display the dialog 'Server Status' which contains information to the current server status. The menu item 'Clear error' allows to clear error messages displayed in the OPC Server.

For a detailed description of the dialog 'Server Status' refer to the OPC Server Help.

A list of all possible error messages is also included in the Help.

- The submenu '?' contains the commands to call the contents of table of Help and to get information to the current OPC Server version.

All menu items of these submenus are explained in detail in the OPC Server Help.

Example for calling a menu item with the mouse

- Click on the submenu 'File'.
  The submenu is opened and you can see the menu items.
- Click on the menu item 'Open Workspace...'.
  The dialog 'Open' appears.

Example for calling a menu item with the keyboard

- Press <CTRL> + <O>.
  The dialog 'Open' appears.

NOTE

All submenus or menu items can be performed pressing the underlined character of the corresponding word. In addition the most important menu items can be executed by using shortcut keys. Shortcuts are shown beneath the corresponding menu item in the submenu.

Toolbar

The toolbar is located below the menu and contains several icons to access often used operations more rapidly. All these operations may be also done with the menu or with shortcuts.

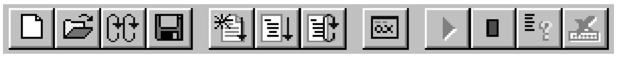


Figure 2-2: Toolbar

NOTE

If you move the mouse pointer to a particular toolbar icon and rest the mouse pointer on the icon the name of the icon and the assigned shortcut is displayed.

The individual icons and the assigned commands are explained in detail in the OPC Server Help.

Using the toolbar to insert a new project

- Click on the icon 'Insert'.
  The dialog 'Insert project' appears.

Keyboard shortcuts

The OPC Server offers the possibility to perform all commands using only the keyboard. This feature was implemented to minimize the steps to do for reaching a certain function. Keyboard shortcuts are used by pressing only one key or a key combination.

The complete shortcut list is provided in the OPC Server Help.

Using a keyboard shortcut for inserting a project

- Press <CTRL> + <I>.
  The dialog 'Insert project' appears.

Main screen

The main screen of the OPC Server is divided into two parts.

∗ The instance tree on the left side.

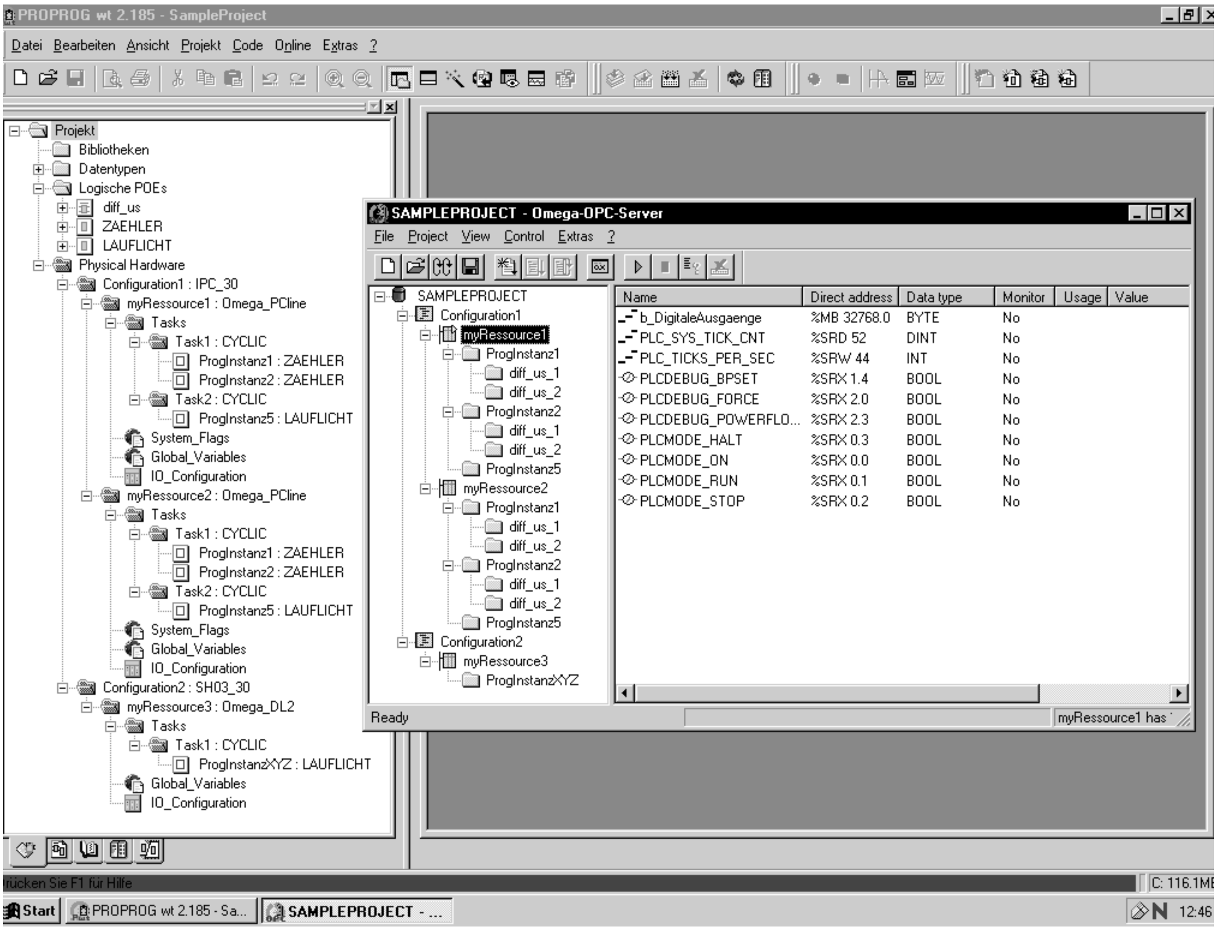
∗ Information to the global resource variables and instance variables of your project.

Instance tree

If you have inserted a PROPROG project into your workspace using the menu item 'Insert Project' in the submenu 'Project', the instance tree is displayed on the left side in the main screen.

The first time a project will be inserted only the icon for your PROPROG project is displayed with the corresponding project name. The instance tree is not visible. In this case you have to extend the instance tree (with click on the symbol „+“ or press „→“) to display the instance structure.

The following figure illustrates an example how an instance tree will be displayed in the programming system PROPROG (figure left) and in the OPC Server (figure right).



Name	Direct address	Data type	Monitor	Usage	Value
b_DigitaleAusgaenge	%MB 32768.0	BYTE	No		
PLC_SYS_TICK_CNT	%SRD 52	DINT	No		
PLC_TICKS_PER_SEC	%SRW 44	INT	No		
PLCDEBUG_BPSET	%SRX 1.4	BOOL	No		
PLCDEBUG_FORCE	%SRX 2.0	BOOL	No		
PLCDEBUG_POWERFLO...	%SRX 2.3	BOOL	No		
PLCMODE_HALT	%SRX 0.3	BOOL	No		
PLCMODE_ON	%SRX 0.0	BOOL	No		
PLCMODE_RUN	%SRX 0.1	BOOL	No		
PLCMODE_STOP	%SRX 0.2	BOOL	No		

Figure 2-3: Instance tree with sample project 'sample project'

The instance tree in the OPC Server is a graphical representation of the structure of your PROPROG project. At the top of each instance tree the inserted project name is displayed.

The OPC Server allows to insert several PROPROG projects into the active workspace with an own instance tree associated to each project. The following elements are part of the instance tree:

- Configuration - The instance tree displays all configuration elements used in the selected (CSV file) project. The configuration can be compared to a controller system, e.g. a rack. Each configuration element contains one or several resources.
  In the PROPROG project the configuration elements are represented graphically in the subtree 'Physical hardware' of the project tree. This representation reflects the structure of configuration elements determined by the PLC.
- Resource - The OPC Server shows all resources declared in the particular configuration of the PROPROG project. The resource can be compared to a CPU which can be inserted in the rack. The communication to the process is only possible via the configuration.
  The OPC Server allows to display the global variables declared in a particular resource. Note that only those global resource variables will be displayed that are written to the *.csv file of your PROPROG project. All global variables are written to the *.csv file if the CSV check box 'All global variables' in the resource setting dialog is set.
  If you select a particular resource in the instance tree, the OPC Server displays the global resource variables stored in the *.csv file on the right side in the main screen.
- Task - The assignment of the program instances to the respective tasks is not displayed.

∗ Program instance - Each program instance of a project can contain one or several function block instances. In the example the program ‚ZAEHLER' contents two instances of the FB ‚diff_us' each. The instance tree in the OPC Server displays those program instances that contain function block instances with variables marked as CSV variables. The instance variables must be explicitly marked as CSV variables in the variable declaration and the CSV check box 'All marked variables' in the resource setting dialog must be set otherwise the variables are not written to the *.csv file of your project.
If you select a particular instance name in the instance tree, the OPC Server displays all instance variables marked as CSV variables on the right side in the main screen.

Detailed information about configurations, resources and instances can be found in the PROPROG manual.

Using the mouse to zoom in the instance tree

- Click on the desired project.
- Click on the sign '+' associated to the project name.
  The configuration subtree is displayed.

Using the keyboard to zoom in the instance tree

- Press <↓> or <↑> to mark the desired project.
- Press <→> to display the configuration subtree.

Global resource variables

The OPC Server allows to display the global resource variables of a particular project resource. These variables are declared in PROPROG as global variables in the selected resource of your project, i.e. the variables can be used in all programs and function blocks of your project. Global variables are declared as VAR_GLOBAL in the global declaration of the project.

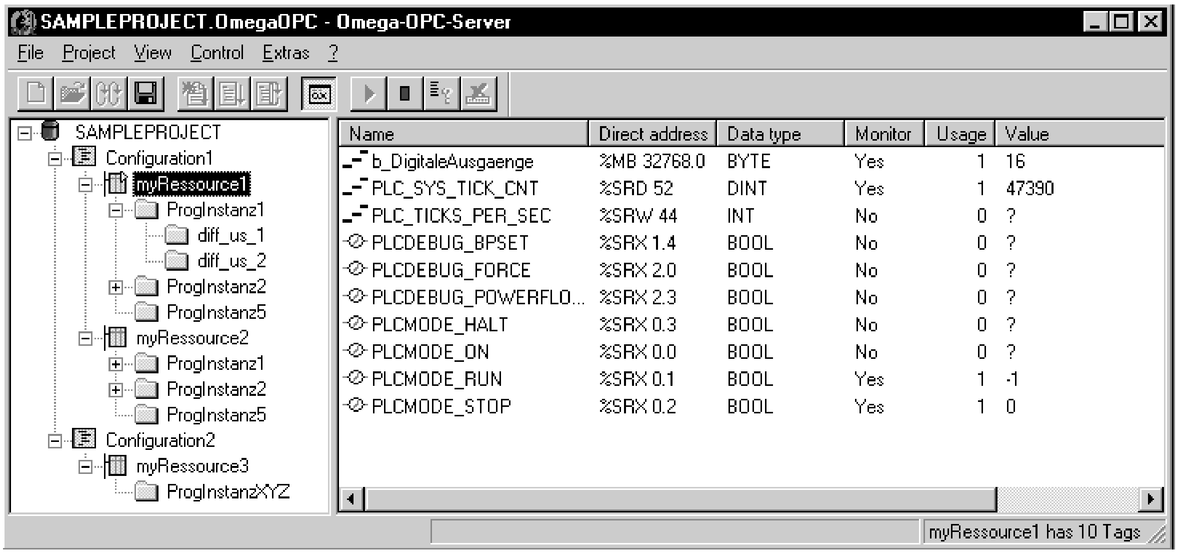
To display the global resource variables in the OPC Server you have to select a particular resource in the instance tree.

The global resource variables are displayed on the right side in the main screen.

NOTE

There are only those global resource variables visible in the OPC Server that are written to the *.csv file of your PROPROG project. To ensure that all global variables are stored in the *.csv file the CSV check box 'All global variables' in the resource setting dialog must be set.

The global resource variables are displayed as follows:



Name	Direct address	Data type	Monitor	Usage	Value
b_DigitaleAusgaenge	%MB 32768.0	BYTE	Yes	1	16
PLC_SYS_TICK_CNT	%SRD 52	DINT	Yes	1	47390
PLC_TICKS_PER_SEC	%SRW 44	INT	No	0	?
PLCDEBUG_BPSET	%SRX 1.4	BOOL	No	0	?
PLCDEBUG_FORCE	%SRX 2.0	BOOL	No	0	?
PLCDEBUG_POWERFLO...	%SRX 2.3	BOOL	No	0	?
PLCMODE_HALT	%SRX 0.3	BOOL	No	0	?
PLCMODE_ON	%SRX 0.0	BOOL	No	0	?
PLCMODE_RUN	%SRX 0.1	BOOL	Yes	1	-1
PLCMODE_STOP	%SRX 0.2	BOOL	Yes	1	0

myRessource1 has 10 Tags

Figure 2-4: Global resource variables displayed in the OPC Server

Online help

The meaning of the several columns on the right side in the main screen is described in detail in the OPC Server Help.

The procedure how to display the global resource variables is explained in the chapter 'How to use the OPC Server'.

Instance variables

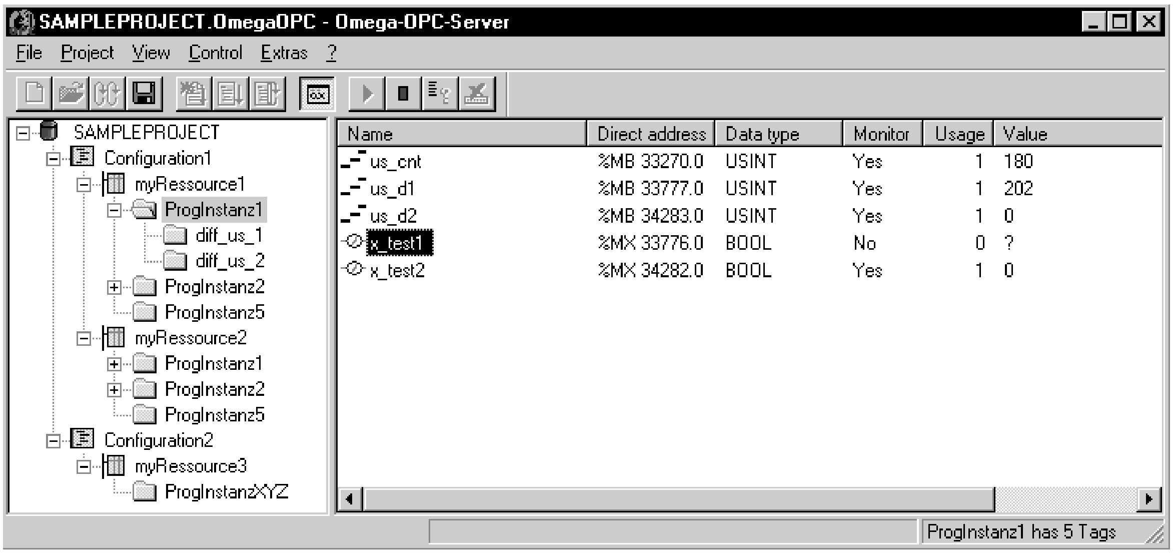
The OPC Server allows to display the instance variables declared for a particular instance (both program instance and function block instance) of your PROPROG project. Instance variables are local variables declared as VAR, VAR_INPUT and VAR_OUTPUT in PROPROG. The instance variables can be used only within the particular instance.

To display the instance variables in the OPC Server you have to select a particular instance in the instance tree.

NOTE

There are only those instance variables visible in the OPC Server that are written to the *.csv file of your project. To ensure that instance variables are stored in the *.csv file the CSV check box 'Marked variable' in the resource setting dialog of the programming system must be set and the instance variables must be marked with CSV.

The instance variables are displayed as follows:



SAMPLEPROJECT.OmegaOPC - Omega-OPC-Server
File Project View Control Extras ?

SAMPLEPROJECT
- Configuration1
  - myRessource1
    - ProgInstanz1
      - diff_us_1
      - diff_us_2
    - ProgInstanz2
    - ProgInstanz5
  - myRessource2
    - ProgInstanz1
    - ProgInstanz2
    - ProgInstanz5
- Configuration2
  - myRessource3
    - ProgInstanzXYZ

Name	Direct address	Data type	Monitor	Usage	Value
us_cnt	%MB 33270.0	USINT	Yes	1	180
us_d1	%MB 33777.0	USINT	Yes	1	202
us_d2	%MB 34283.0	USINT	Yes	1	0
x_test1	%MX 33776.0	BOOL	No	0	?
x_test2	%MX 34282.0	BOOL	Yes	1	0

ProgInstanz1 has 5 Tags

Figure 2-5: Instance variables displayed in the OPC Server

The meaning of the several columns on the right side in the main screen is described in detail in the OPC Server Help.

The procedure how to display the instance variables is explained in the chapter 'How to use the OPC Server'.

Status bar

The status bar is displayed at the bottom of the user interface. It contains several fields where different kinds of messages are displayed while you are working with the OPC Server.

The left area of the status bar describes actions of menu items as you use the arrow keys to navigate through menus. If you choose a menu item a short information on this menu item is given in the message field of the status bar.

The second field is used to display error messages detected by the OPC Server. If several error messages exist, only one error message is displayed in this field. To display further error messages you have to clear the current message using the menu item 'Clear error' in the submenu 'Control'.

The third field contains the number of tags associated to the selected item in the instance tree.

2.6 Using help for the OPC Server

The OPC Server offers a full Help which contains topics for all parts of the application. The table of contents of help contains a list of all main topics. It is called choosing the menu item 'Contents' in the submenu '?'.

🖱 **Calling the table of contents with the mouse**

- Choose the menu item 'Contents' in the submenu '?'.
  The table of contents of the OPC Server Help appears.

⌨ **Calling the table of contents with the keyboard**

- Press <ALT> + <?>.
  The submenu '?' is opened.
- Press <C>.
  The table of contents of the OPC Server Help appears.

3 HOW TO USE THE OPC SERVER

3.1 Correct usage of the OPC Server

NOTE

Before starting the OPC Server ensure that the CSV check boxes in the resource setting dialog of the PROPROG project are set. The OPC Server requires the sr.csv file for correct operation.

The following steps must be performed for correct usage of the OPC Server:

- Start the OPC Server.

For a detailed description how to start the OPC Server refer to the chapter 'Getting started with the OPC Server'.

- Insert an existing PROPROG project into the active workspace.
- Save the active workspace.
- Exit the OPC Server.
- Start your OPC Client program and connect to the OPC Server.

3.2 Inserting a project

The first step you have to do after calling the OPC Server is inserting a PROPROG project in the active workspace. The OPC Server needs the project file (*filename*.mwt) to enable the communication between your PLC and an OPC Client.

If a new workspace is created the new workspace is named 'Untitled'. The name of the active workspace is displayed in the title bar at the top of the user interface. You can insert and edit the workspace 'Untitled' and save it afterwards under the name you want to use it.

Inserting a project with the mouse

- Choose the menu item 'Insert Project' in the submenu 'Project'.
  The dialog 'Insert project' appears.
- Choose the name of the PROPROG project you want to insert.
- Confirm the dialog.
  The project is inserted in the active workspace and the project name appears on the left side in the main screen.

Inserting a project with the keyboard

- Press <CTRL> + <I>.
  The dialog 'Insert project' appears.
- Choose the name of the PROPROG project you want to insert.
- Press <↵> to confirm the dialog.
  The project is inserted in the active workspace and the project name appears on the left side in the main screen.

3.3 Communication with PLC

The OPC Server uses for the communication with the target resource (PLC) the same connection as the programming system PROPROG wt II.
The settings of the communication parameter done in the dialog „Resource settings for ...“ are transferred.

For a detailed description of the dialog „Resource settings for ...“ refer to the PROPROG wt II Manual.

If the target PLC is attainable with the PROPROG control dialog for resources, then although for the OPC Server.

Mind the following note for serial ports:

NOTE

The OPC server does not share the serial port with any other application running on the same PC. This means, if the OPC server is accessing a PLC via the serial port, it is not possible to enable online services in PROPROG wt II for the same PLC. See “Notes for client developers” on page 23.

3.4 Saving the active workspace

Every time you have made any changes to the active workspace you should save it. If you close a workspace without having saved it a dialog appears which asks you to save the current changes. Confirm this dialog. This ensures that the next time you will open this workspace the current settings are available.

The first time the OPC Server is called a new workspace with the name 'Untitled' is created automatically. To use this workspace for further operations you have to save it under the name you want to use it.

If you save this workspace the dialog 'Save As' appears. In this case enter a new name for the workspace (*filename*.OmegaOpc) and choose the folder where your want to save it.

We recommend to save the active workspace under the name of your inserted PROPROG project.

Saving the active workspace with the mouse

- Choose the menu item 'Save Workspace' in the submenu 'File'.
  The active workspace is saved.

Saving the active workspace with the keyboard

- Press <CTRL> + <S>.
  The active workspace is saved.

3.5 Displaying the global resource variables

If you have connected an OPC Client to the OPC Server and you have inserted at least one PROPROG project into the active workspace, the global resource variables of this project can be displayed and monitored on the right side in the main screen.

The global resource variables are only visible if the menu item 'Monitor' in the submenu 'View' is activated.

NOTE

The OPC Server displays only the global resource variables declared as CSV variables in your PROPROG project. Therefore ensure that the CSV check boxes in the resource setting dialog of the PROPROG project are set.

Using the mouse to display global resource variables

- On the left side in the main screen click on the desired project name.
- Click on the sign '+' associated to the project name.
  The Configuration icon is displayed.
- Click on the sign '+' associated to the Configuration icon.
  The Resource icon is displayed.
- Click on the resource name.
  The global resource variables appear on the right side in the main screen.

Using the keyboard to display global resource variables

- Press <↓> or <↑> to mark the desired project.
- Press <→> to display the Configuration icon.
- Press <↓> to mark the Configuration icon.
- Press <→> to display the Resource icon.
- Press <↓> to mark the Resource icon.
  The global resource variables appear on the right side in the main screen.

3.6 Displaying the instance variables

The OPC Server allows to display the instance variables of a particular program instance in your PROPROG project. If you have connected an OPC Client to the OPC Server and you have inserted at least one PROPROG project into the active workspace, the instance variables declared in the program instances of this project can be displayed and monitored on the right side in the main screen.

The instance variables are only visible if the menu item 'Monitor' in the submenu 'View' is activated.

NOTE

The OPC Server displays only the instance variables marked as CSV variables in your PROPROG project. Therefore ensure that the CSV check boxes in the resource setting dialog of the PROPROG project are set.

Using the mouse to display instance variables

- On the left side in the main screen click on the desired project name.
- Click on the sign '+' associated to the project name.
  The Configuration icon is displayed.
- Click on the sign '+' associated to the Configuration icon.
  The Resource icon is displayed.
- Click on the sign '+' associated to the Resource icon.
  The program instance icon is displayed.
- Click on the program instance icon.
  The instance names are displayed.
- Click on the instance name.
  The instance variables are displayed on the right side in the main screen.

Using the keyboard to display instance variables

- Press <↓> or <↑> to mark the desired project.
- Use the <→> key to move through the instance tree until the desired instance name is marked.
  The instance variables are displayed on the right side in the main screen.

3.7 Manually reimporting a changed project

NOTE

If you have made any changes to your PROPROG project and you compiled it, you can manually reimport this project into the OPC Server in order to display the edited project structure and to provide this project to the OPC Client.

In addition to a manual reimport of a changed PROPROG project the OPC Server allows an automatic update of the current project opened in the OPC Server. For further information to this feature refer to the next section.

If you have for example inserted new resources in your PROPROG project, these changes are only visible in the OPC Server when you reimport the particular project.

Reimporting of a changed project can be done using the menu items 'Reimport all' in the submenu 'File' or 'Reimport Project' in the submenu 'Project'.

Using the mouse to manually reimport a changed project

- In the instance tree click on the project you want to reimport.
  The desired project is marked.
- Choose the menu item 'Reimport Project' in the submenu 'Project'.
  The selected project is automatically reimported. The project structure now represents the current structure of your project.

Using the keyboard to manually reimport a changed project

- Press <↓> or <↑> to select the project you want to reimport.
- Press <CTRL> + <R>.
  The selected project is automatically reimported. The project structure now represents the current structure of your project.

3.8 Automatic update of the current project

The OPC Server allows an automatic update of the current PROPROG project. Therefore it is no longer necessary to reimport a PROPROG project, that you have changed and built.

To allow this automatic process you have to mark the checkbox 'Include OPC data' in the PROPROG dialog 'Download'. In this case the CSV file, which contains all variables to be monitored using the OPC Server, is included in the download process.

After downloading the CSV file to the PLC the OPC Server updates automatically the loaded project so that the edited project structure can be seen immediately and the OPC Client can get the current values from the connected PLC.

NOTE

An automatic update of the actual project is **impossible** at Ωmega Drive-Line II.

3.9 Exiting the OPC Server

You can exit the OPC Server whenever you want. It doesn't matter if an OPC Client is connected to the OPC Server. If you have not saved the changes you have done in your active workspace a dialog appears and you can either save the changes or close the active workspace without saving it.

Exiting the OPC Server with the mouse

- Choose the menu item 'Exit' in the submenu 'File'.
  The OPC Server is closed.

Exiting the OPC Server with the keyboard

- Press <ALT> + <F>.
  The submenu 'File' is opened.
- Press <x>.
  The OPC Server is closed.

4 NOTES FOR CLIENT DEVELOPERS

4.1 General

This chapter provides specific information about developing OPC clients for the OmegaOS OPC server. It does not include support for creating OPC clients in general e.g. how to use the OPC interfaces or example client code. A good approach in creating OPC clients is to use a client-toolkit like it is provided by Technosoftware (http://www.technosoftware.ch/).

4.2 Related documents

#	Title	Path to file
1	OPC Data Access Specification 1.0A	http://www.opcfoundation.org/
2	CALL-Runtime Specification 2.0	http://www.open-control.com/

4.3 OPC Interface

4.3.1 Custom interface

ProgID

The ProgID of the custom interface is: OmegaOPC-Server.1

Implementation

All required OPC Interfaces (according #1) are implemented, but some interface functions are not implemented (returns E_NOTIMPL):

- IOPCAsyncIO :: Cancel(DWORD dwTransactionID)
- IOPCServer :: GetErrorString(HRESULT dwError,LCID dwLocale, LPWSTR * ppString)

The optional OPC interface IOPCBrowseServerAddressSpace is implemented, except one function (returns E_NOTIMPL):

- IOPCBrowseServerAddressSpace :: BrowseAccessPaths(LPCWSTR szItemID, LPENUMSTRING* ppIEnumString)
  → .....returns E_NOTIMPL

The optional public groups are not supported, but the interface IOPCPublicGroupStateMgt is implemented:

- IOPCPublicGroupStateMgt :: GetState(BOOL * pPublic)
  → .....sets always pPublic to FALSE, indicating that the group is private.
- IOPCPublicGroupStateMgt :: MoveToPublic(void)
  → .....always returns E_NOTIMPL

4.3.2 Automation interface

ProgID

The ProgID of the automation interface is: PcosOPCAutomation.1

Implementation

IOPCBrowseServerAddressSpaceDisp is not implemented in an OPC conform way. To browses the available items in the server, SetEnumeratorType(0, 0) must be called before "For Each ... In ... " works properly.

- Only OPC_FLAT browsing is supported by the automation interface.

Visual Basic Example:

```
Private Sub BrowseAddressSpace_Click()
Dim Svr As IOPCBrowseServerAddressSpaceDisp

' non OPC conform handling:
' SetEnumeratorType(0, 0) browses the available items in the server
Call OpcServer.SetEnumeratorType(0, 0)

' show the number of available items
MsgBox (OpcServer.Count)

Set Svr = OpcServer

Dim Val As Variant
For Each Val In Svr
    MsgBox (Val)
Next Val

End Sub

'-------------------------------------------------------------------
Private Sub EnumPrivateGroup_Click()

' SetEnumeratorType(0, 1) browses the privat groups (the default)
Call OpcServer.SetEnumeratorType(0, 1)

' show the number of private groups
```

```
MsgBox (OpcServer.Count)

Dim Val As Variant
For Each Val In OpcServer
    MsgBox (Val.Name)
Next Val

End Sub
```

The following automation interface function are not implemented (returns E_NOTIMPL):

- IOPCGroupStateMgtDisp :: CloneGroup (VARIANT Name, IDispatch **ppDisp)
- IOPCItemDisp :: put_ActiveStatus(VARIANT_BOOL ActiveStatus)
- IOPCItemDisp :: get_Blob(VARIANT *pBlob)
- IOPCItemDisp :: put_ClientHandle(long Client)
- IOPCItemDisp :: get_RequestedDataType(short *pRequestedDataType)
- IOPCItemDisp :: put_RequestedDataType(short RequestedDataType)
- IOPCItemDisp :: put_Value(VARIANT NewValue)
- IOPCItemDisp :: get_EUType(short *pError)
- IOPCItemDisp :: get_EUInfo(VARIANT *pError)
- IOPCServerDisp :: GetErrorString(long Error, long Locale, BSTR * ErrorString)
- IOPCServerDisp :: SaveConfig(BSTR FileName)
- IOPCServerDisp :: LoadConfig(BSTR FileName)
- IOPCServerDisp :: GetItemIDString(BSTR ItemDataID, BSTR *ItemID)
- IOPCServerDisp :: SetAccessPathEnumerator(BSTR ItemID)

4.4 Datatypes

4.4.1 Mapping IEC- to VARIANT types

The following table describes the mapping of the IEC-Datatypes to the VARIANT types:

IEC Datatype	IEC Size/Byte	Variant Datatype	Supported
BOOL	1	VT_BOOL	yes
BOOL8	1	VT_BOOL	yes
SINT	1	VT_I2	yes
INT	2	VT_I2	yes
DINT	4	VT_I4	yes
LINT	8	VT_R8	no

IEC Datatype	IEC Size/Byte	Variant Datatype	Supported
USINT	1	VT_UI1	yes
UINT	2	VT_I4	yes
UDINT	4	VT_R8	yes
ULINT	8	VT_R8	no
REAL	4	VT_R4	yes
LREAL	8	VT_R8	no
STRING	85	VT_BSTR	yes
BYTE	1	VT_UI1	yes
WORD	2	VT_I4	yes
DWORD	4	VT_R8	yes
LWORD	8	VT_R8	no
TIME	4	VT_R8	yes

4.4.2 Support for userdefined datatypes

String

Userdefined strings are supported but the length is limited to 80 characters. If a string exceeds this length it is truncated to 80 characters.

Array

One dimensional arrays of elementary datatypes are supported. The whole array is provided through a safearray of the type VT_ARRAY | VT_*. Array components are handled like elementary datatypes but the corresponding item name is the name of the array extended with the index in brackets
e.g. MYPROJ.MYCONF.MYRES.MYARR[5]

The single elements of arrays are not accessible by the IOPCBrowseServerAddressSpace interface. Arrays greater than 64KB cannot be provided through the OPC server.

Struct

Structs cannot be supported through a common way with released versions of the OPC Data Access specification.

4.5 PLC access (IEC 61131 resource)

Communication

The possible access paths to the PLC that are provided in PROPROG wt II are also provided by the OPC Server. These are the serial port (COM1...4) and TCP/IP socket communication at the moment. The communication parameters of the PROPROG project are also used in the OPC Server.

Note for serial communication:

NOTE

The OPC server does not share the serial port with any other application running on the same PC. This means, if the OPC server is accessing a PLC via the serial port, it is not possible to enable online services in PROPROG wt II for the same PLC.

4.6 CALL-Runtime

To change the PLC mode, download a bootproject or control the PLC status, several interfaces of the CALL-R specification 2.0 are supported:

ICallRServer

Only the function MakeDefaultCallrProgramGroupObject() is implemented. All other functions return E_NOTIMPL.

ICallrProgramGroup

The following functions are not implemented:

- AddCallrProgramGroupObject
- CreateCallrProgramGroupEnumerator
- GetProgramGroupState
- HaltProgram

AddCallrProgramObject

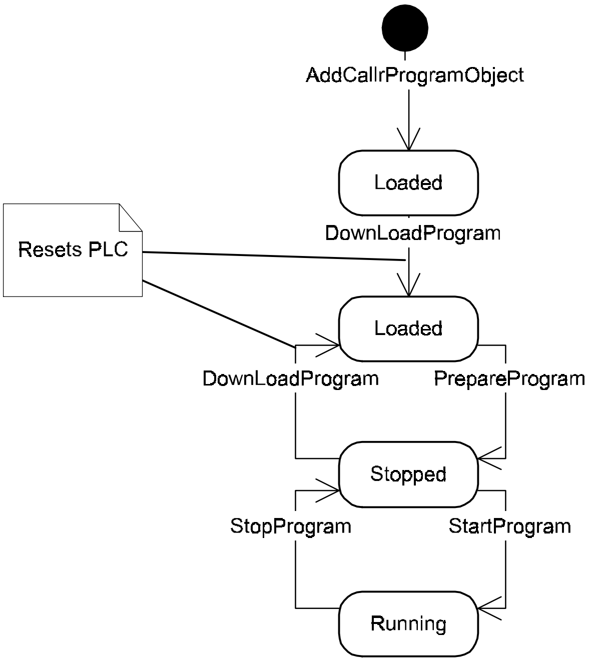
The first parameter ‚szProgramName' must be given in the form PROJECT.CONFIGURATION.RESOURCE and not as CONFIGURATION.RESOURCE.PROGRAM like it is specified in #2. The second parameter ‚nProgramType' must be set to 3 (IEC 61131 program).

ICallrProgram

The following functions are not implemented:

- UpLoadProgram
- HaltProgram

The OmegaOS Callr server acts like the following statechart:



DownLoadProgram

Only OmegaOS created bootprojects which are available as file (bootfile.pro) in the resource directory of PROPROG projects can be downloaded. Therefore the first parameter ‚vProgramCode' should be set to VT_EMPTY and the second parameter ‚nCodeAttribute' must be set to 1 (Automatically started when restarting the system).

NOTE

To get the bootproject created at PROPROG wt II compile time, enable the checkbox "Generate bootproject during compile" in the resource settings dialog of PROPROG.

GetProgramState

The following table describes the mapping of the OmegaOS PLC states to the Callr program states:

OmegaOS PLC state	Callr program state
On	CALLR_PROGRAM_LOADED
Download	CALLR_PROGRAM_LOADED
Stop	CALLR_PROGRAM_STOPPED
Run	CALLR_PROGRAM_RUNNING
Halt	CALLR_PROGRAM_HALTED

NOTE

⇒ It is not possible to distinguish between the modes ‚On' and ‚Download'.

⇒ The function returns E_FAIL if the PLC is not reachable (timeout).

⇒ Every call of this function results into PLC access. Calling this function very frequently decreases the OPC server performance.

4.7 Performance

The following hints should help to increase the performance of the client applications.

Prefer static groups to dynamic groups

Every call of AddItems() and RemoveItems() leads to additional traffic with the PLC. This issue gets really important if the PLC is connected via a slow bus e.g. a serial line. An example: A visualization with some pictures could create a group for every picture and add all items of a picture at the time the picture is shown. The performance decreases if the application adds and removes items every time the user presses a button on a page.

Prefer IDataObject to IOPCSyncIO for reading

The IDataObject enables the OPC server to inform the client in the case that the value of one or more items has changed and releases the client from polling the values in a cyclic interval. This increases the overall performance significantly and decreases the amount of processor time the OPC server needs.

Call functions with arrays of pointers instead of single pointers

If possible, call an interface-function only once with an array of pointers to data instead of calling the function for every single pointer to data. This is in particular true for the AddItems() and RemoveItems() functions.

Select the group update rate only as high as really needed

The requested group update rate (third parameter of the AddGroup() function) is also used internally for getting data from the PLC. Setting this value very low increases the amount of processor time the OPC server needs.

Choose the custom interface for large applications

The additional overhead of the automation interface can decrease the performance of large client applications.

4.8 Client debugging

Test client

A simple test client named "OpcClient.exe" can be found in the installation directory of the OPC Server. This client is provided for testing and debugging purposes. It is available only as binaries and not as source files.

Debug trace

The OPC server is able to generate a trace file with all accesses of the custom interface. This file could be useful during development of OPC clients. The tracefile is enabled by creating the following Registry entry:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Baumueller Nuernberg\Omega OPC-Server v10\Dump]
"Level"=dword:00000001
```

Every time the OPC server is started, it generates the tracefile "c:\PcosOpcR.csv" which can be opened with a text editor.

4.9 Limits

Number of items per workspace

The total amount of items which are accessible through a single OPC Server is limited to 12000 per workspace. This is the static configuration limit for the namespace of the OPC Server.

Number of items per resource

The total amount of items of the same resource which can be added to groups is limited to 3200 although the total amount of configured items per resource is allowed to exceed this limit. This is a runtime limit of the OPC server. An example: Imagine a configuration with a single resource with 5000 exposed CSV-variables. All the variables are accessible in the namespace e.g. for browsing but it is not possible to add more than 3200 of these variables to groups because the OPC server cannot access more than 3200 different variables on the same PLC.

Number of items per group

The total amount of items per group is limited to 3200. This is a runtime limit of the OPC server.

Group update rate

The group update rate must not fall short of 10ms.

5 INDEX