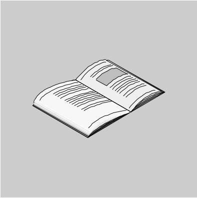
OPC Factory Server V2.5

User Manual

TLX DM OFS 2.5 Eng V2.5

Table of Contents



About the book

At a Glance

Document Scope

This manual describes the software installation of the OFS product.

Validity Note

The update of this documentation takes the latest version of OFS into account.

User Comments

We welcome your comments about this document. You can reach us by e-mail at TECHCOMM@modicon.com

Introduction to the OFS product

1

Introduction

Aim of this chapter

The aim of this chapter is to introduce to you the OFS (OPC Factory Server) product.

What's in this chapter?

This chapter contains the following topics:

Introducing the OFS Server

General

The OFS product (OPC Factory Server) is a multi-controller data server which is capable of communicating with PLCs of the TSX Compact, TSX Micro, TSX Momentum, TSX/PCX/PMX Premium, TSX Quantum, TSX Series 7 and TSX S1000 families in order to supply the OPC clients with data.

This means that it provides client applications with a group of services (methods) for access to control systems variables.

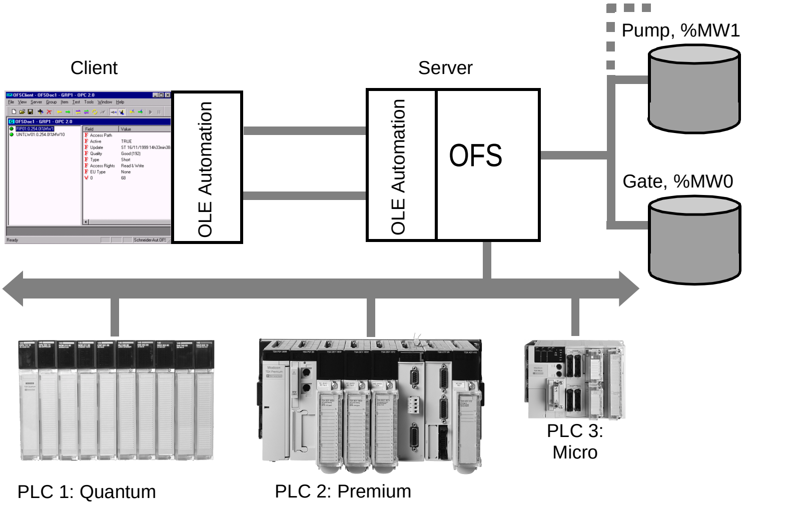
OFS is a Data Access OPC server, compliant with both OPC 1.0A and OPC 2.0, i.e., it will function with any OPC compliant client and with two types of OPC compliant software:

- supervisory software (see distributor range): the OFS server assumes the role of a driver by ensuring communication with all devices supported by Schneider Automation,
- Custom developed supervisory software, using either the OLE Automation interface or the OLE Custom interface.

Note: Knowledge of one of the following languages is required when creating a client application for the OFS, in particular for OLE Automation, OLE Custom programming and exception management:

- Microsoft Visual Basic, version 6.0 SP3 or above,
- Microsoft Visual C++, version 6.0 SP3 or above,
- Microsoft VBA in Excel, version 8.0 (Office 97) or above.

The illustration below shows an OFS interface:



The OFS server provides the interface between devices and one or more client applications in which the values of some of the data of these devices are to be viewed and/or modified.

The main characteristics are:

- multi-device,
- multiple communication protocols,
- multi-client,
- access to devices and variables via address or symbol,
- access to the server in local or remote mode,
- use of a notification mechanism, enabling values on the change of state only to be sent to the client. The server offers two modes for exchanges with the PLC: the default " classic " (polling) mode, or the " Push Data " mode where data is sent at the initiative of the PLC. This mode is extremely effective when changes of state are infrequent,
- automatic alteration of the size of network requests,
- exposes its services via both the OLE Automation and OLE Custom interfaces,
- compatible with both OPC Data Access Standard Version 1.0A and 2.0.

The OFS server offers the following services:

- reading and writing of variables in one or more PLCs present on one or more different networks. The OFS server is equipped with powerful algorithms for optimizing the network requests by compacting and grouping together the declared variables,
- a user-friendly configuration tool, giving a greater understanding of the parameters needed for the server to function efficiently, and a tool enabling parameters to be modified online, in order to maximize utilization flexibility,

- option of using the PLC application's list of symbols from the majority of existing file formats,
- a browse interface, which enables the user to discover quickly and easily which devices s/he can dialog with and their associated symbols,
- a specific list of items corresponding to the execution of particular functions: status and starting/stopping the PLC, alarm supervisory function.

Communication with PLCs

At a Glance

The OFS server operates with the Schneider Automation TSX Micro, TSX/PCX/ PMX Premium, TSX Series 7 and TSX S1000 PLC ranges (UNITE or Modbus communication protocol) as well as with the TSX Quantum, TSX Momentum and TSX Compact ranges (Modbus protocol) on the following networks:

- UNITE protocol: Ethernet network (XWAY on TCPIP, ETHWAY), FIPWAY, Unitelway, as well as ISAWAY when using a PCX coprocessor,
- Modbus protocol: Ethernet network (Modbus on TCPIP), Modbus Plus, and Modbus RTU serial link.

It also allows access to TSX S1000 PLCs via the JBUS protocol (subset of Modbus RTU) and TSX Premium via the MODBUS protocol.

The OFS server is compatible with the TSX Nano on Unitelway only, with the following restrictions:

- read operations only,
- access to a single word or x bits within 16 consecutive bits.

On TSX Micro, the use of ETZ410/ETZ510 cards in MODBUS TCP protocol is not supported.

The figure below outlines OFS 2.5 compatibility with devices in the Schneider Automation range and the different networks:

	PREMIUM	MICRO	Series 7	Series 1000	PC
Ethway	TSX ETY 110• (ETHWAY)		TSX ETH107 TSX ETH 200	ETH030•	Ethernet Card
TCP/IP (XIP)	TSX ETY110• (TCP/IP) TSX ETY410• (TCP/IP) TSX ETY510• (TCP/IP)	TSX ETZ410 TSX ETZ510			Ethernet Card
Unitelway	TER CPU port TSX SCP11•	TER CPU port TSX SCP11•	TSX SCM22•		COM1/ .. /COM4
Fipway	PCMCIA FPP20	PCMCIA TSX FPP20	TSX P•7455 TSX FPP20		TSX FPC10 (ISA bus card) TSX FPP20 (PCMCIA)
Isaway	TPCX P57203 TPCX P57353				ISA Bus
Modbus TCP	TSX ETY110• TSX ETY410• TSX ETY510•				Ethernet Card
Modbus	TSX SCP11•	TER CPU port	TSX SCM22•	JB cards•	COM1/ .. /COM4

	PREMIUM	MICRO	Series 7	Series 1000	PC
Modbus +	TSX MBP100	TSX MBP100			AM-SA85-00• (ISA bus card) 416NHM3003• (PCI bus card)

The table below shows the compatibility of OFS 2.5 with certain PLC ranges and the different standards:

	QUANTUM	MOMENTUM	COMPACT	PC
Modbus TCP/ IP.	140NOE 771••	171CCC96030 171CCC98030		Ethernet Card
Modbus	Built-in channel	171CCC760•• 171CCC780••	Built-in channel	COM1/ .. /COM4
Modbus +	Built-in channel 140NOM211••		Built-in channel	AM-SA85-00• (ISA bus card) 416NHM3003• (PCI bus card)

Installing/uninstalling the OFS product

2

At a Glance

Aim of this Chapter

The aim of this chapter is to show you how to install and uninstall the OFS product.

What's in this chapter?

This chapter contains the following Sections:

2.1 Contents of the OPC Factory Server Product

OFS Contents

Contents of the product

The OFS product comprises:

- a CD-ROM with setup instructions,
- one Product Code ID also known as the CD KEY in the text below.
- A CD-ROM including all the XWAY drivers (Modbus drivers do not need to be supplied, Modbus RTU and Modbus TCPIP are integrated into the OFS, the Modbus + driver is supplied with the network card).

The OFS CD-ROM includes:

- the OFS server,
- the OFS manager,
- the OFS Configuration Tool,
- documentation (English/French/German language),
- sample of Symbol Tables, sample Concept and PL7 applications,
- an OPC test client.

Note:

- the OFS product does not contain any cable for communication between the PC and the PLC.
- the documentation is in Acrobat Reader format (pdf file). If you do not have this software on your machine, you will find it on the CD, in the Redist\Acrobat directory.

2.2 Hardware and Software Configuration of the OPC Factory Server Product

Hardware and Software Configuration

Description of the Configuration

Minimum configuration: Pentium 166 MHz, 64 Mb of RAM.
The OFS product requires a PC "Wintel" platform: Intel x86 monoprocessor with one of the following Microsoft Windows 32-bit operating systems:

- Windows 2000 professional upgraded with Service Pack 5 (or later),
- NT 4.0 updated with Service Pack 5 (or later),
- Windows 98 (first or second edition),
- Windows 95 upgraded with Service Pack 1 (or later) with the DCOM option (1.3).

Note:

- The DCOM option is necessary under Windows 95/98 even for running the OFS server locally. The DCOM version required is version 1.3 (or later) and is included on the OFS installation CD-ROM. To install DCOM, follow the instructions given in the ReadMe.txt file on the installation CD-ROM in the Redist\DCOM\W95 (or W98) directory.

To determine which version of Windows and which service pack is installed on your PC:

- **Windows 2000 Professional:**
  Open "Parameters\Configuration Panel\System". The system version is displayed under the "general" tab, and must be at least: version 5.00.2195 (Service Pack 1). In Windows 2000, DCOM is always installed.
- **Windows NT:**
  Open Programs\Administrative tools\Windows NT\Diagnostics - your system version is displayed; it should be at least:
  - Version 4.0 (Build 1381: Service Pack 5).

  Under NT 4.0, DCOM is always installed.
- **Windows 95 / 98:**
  Open the Control Panel, then click on the "System" icon, the displayed page shows your system version :
  - 4.0.950 = no service pack installed - install service pack 1,
  - 4.0.950a = service pack 1 is installed, click OK,
  - 4.0.950b = service pack 2 is installed, click OK,
  - 4.10.1998 = Windows 98 (first edition), click OK,
  - 4.10.2222 = Windows 98 (second edition), click on OK,

In Windows 98, DCOM is generally already installed. If this is not the case, the W98 DCOM option is supplied with the OFS product. To install DCOM, follow the instructions given in the ReadMe.txt file on the installation CD-ROM, in the **Redist\Dcom\W98** directory.

In order to verify that DCOM is installed under Windows 95, Open the Control Panel, then click on "**Add/Remove programs**".

If DCOM is installed, you should find a line: "DCOM for Windows 95".

If you don't have this line and you are sure that DCOM is installed (for instance because you installed Internet Explorer 4.0) go on and ignore the installshield warning. Otherwise you should install DCOM option before installing the server (don't forget to reboot after DCOM installation).

To install DCOM, follow the instructions given in the **ReadMe.txt** file on the installation CD-ROM, under the **Redist\Dcom\W95** or **Redist\Dcom\W98** directory.

2.3 Product Installation Procedure

Introduction

Aim of this Section

The aim of this section is to show you how to install the OPC Factory Server product.

What's in this section?

This section contains the following topics:

Preparing to Install the OFS Product

Foreword

Note: In Windows 2000 or NT 4.0, it is necessary to have administrator rights.

If there is a previous version of OFS installed, it must be uninstalled beforehand.

The installation procedure is as follows: insert the CD-ROM into the CD drive. The CD is Auto-run, therefore if your PC is set up for this feature you should see the OFS main window. If Auto-run is disabled or does not work:

- click on the Start button in the task bar,
- select Settings -> Control panel,
- click on the Add/Remove Programs icon in the Control Panel,
- click on the Install/Uninstall tab then click on the Install button and follow the instructions, For Windows 2000, click on Add New Programs, then on the CD-ROM or floppy disk button and follow the instructions,
- the Install Tool will automatically find the OFS Setup.exe program on the CD, it will also display the path and file name then prompt you to perform the installation.

Two options are offered: **Full station** or **Remote station**.

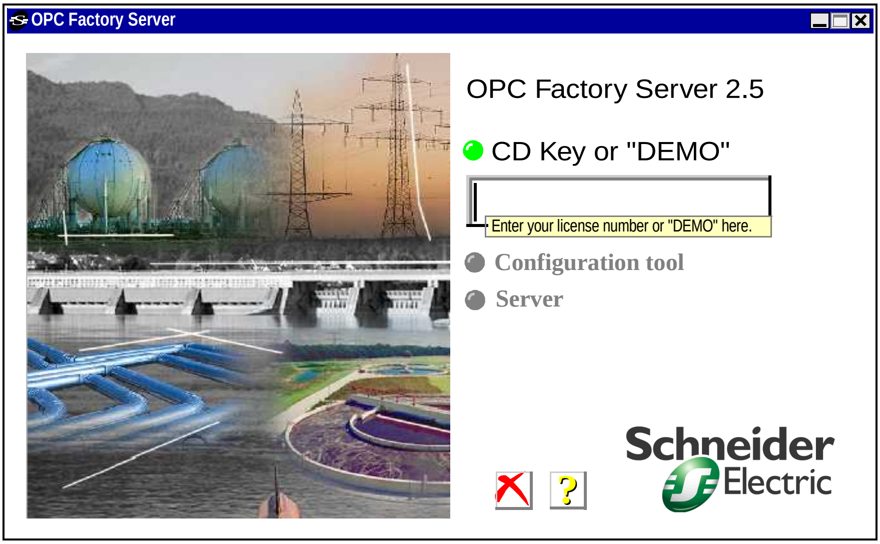
- full station is when a machine supports the OFS server and client(s). This machine can also be used as a remote server and access the OFS server by means of DCOM without requiring the installation of any additional accessories. An official OPC Factory Server license is required for each server station desired,
- remote station is a machine which only receives one or more OPS client(s) and remotely accesses the OPC server via DCOM. You do not need an OPC Factory Server license to install a remote station.

Full station and client station cannot be both installed on the same computer. These options are exclusive.

Installation of the Full Station (client + server)

Installation Procedure

The window shown below is displayed showing the OFS installation options:



OPC Factory Server

OPC Factory Server 2.5

CD Key or "DEMO"

Enter your license number or "DEMO" here.

The first thing to do is to enter either your CD-KEY or the word DEMO in uppercase into the dialog box (Your CD-KEY is your product-ID, printed on a piece of paper included with the CD-ROM). The "Configuration tool" and "Server" options become active.

When running in DEMO mode, the server has all the features enabled but some restrictions apply:

- OFS use cannot exceed 2 hours (after which the server must be shut down and restarted),
- the server cannot be run in NT service mode,
- XWAY drivers for Ethway and TCPIP are not available.

Click on **Configuration tool**:

Follow the messages which appear on the screen to perform the installation. According to the operating system, the installation of DCOM and MDAC components may be suggested.
If an old alias file is detected, the system will suggest reverting to it.

When installation of the Configuration Tool is finished, the installation start-up screen is displayed again.

Click on **Server**:
Follow the messages on the screen to perform the installation. The installation program offers the following options:

- optional installation of an OPC test client program,
- optional installation of product documentation in PDF format.

CAUTION

Use in Windows 98

When executing the installation program in Windows 98, error messages indicating problems with the version and corruption of Windows system DLL files may appear (for example: COMPOBJ.DLL, DDEML.DLL). These messages should be ignored (Windows 98 automatically corrects these problems), as they do not affect operation of the PC.
After each installation or re-installation of the server and before starting the server, it is strongly recommended that the configuration tool is executed, even if no modifications are required, in order to guarantee that all configuration parameters are taken into account.

Failure to observe this precaution can result in injury or equipment damage.

Note:

- If upgrading from a previous OFS version, first uninstall the old version and then install the new one. Performing an uninstall does not affect the parameter configuration data stored in the registry.
- In particular, avoid spaces in file names.

Remote Station

Installation Procedure

No license is required to install a remote station. Follow the messages which appear on the screen to perform the installation. The installation program offers the following options:

- **OFS remote server registration and OPC proxy DLLs** : an update of your registry base and of some system files will be performed (OPCproxy.dll & OPCcommon.dll),
- **OFS Testing Client**: this can be used to access and test any OPC server. It is not dedicated to the OFS server,
- **OFS Manager**: this is a tool allowing local or remote access to debug information from the OFS server. It also enables the user to ask the OFS server to perform certain online actions (creation of a new alias, reload tables of symbols etc.).
- **OPC Automation interface 1.0 and 2.0**: this option will install the DLL files required to be able to use the Automation Interface of the OFS server. Both dll for OPC automation 1.0 & OPC automation 2.0 will be installed.

To operate correctly, the remote station must have been the subject of a DCOM configuration on both the remote station and the server station.

Driver Installation

Description

For Modbus drivers, RTU driver and TCP-IP driver are included in the server: no additional installation is required. The Modbus Plus driver is not included with the OFS product. It should be purchased and installed separately (e.g.: with SA85 card).

For XWAY drivers, the CD-ROM is included with the product. Every available driver may be installed. There are none installed by default. It is the user's responsibility to ensure appropriate installation is carried out. The OFS server can use XWAY drivers already installed on your machine provided that they are not too old. The compatibility table below indicates the minimum version you should have installed to be sure that OFS will operate properly. Use of the OFS with older versions is neither supported nor guaranteed.

Compatibility Table:

XWAY Drivers	**Minimum version** according to the operating system			
	W95	W98	W2000	NT4.0 SP5
UNITELWAY	7.8	1.2		1.6
FIPWAY	2.4	1.0		1.3
FIPWAY PCMCIA	1.1		1.1	1.1
ISAWAY	1.2	1.0		1.4
ETHWAY	2.6		1.0	1.1
XWAY/TCP-IP	1.5			

Concept Link

Description

In order to be compatible with OFS v2.5, the Concept workshop should be upgraded either with a minimum version of the release service, or using the Concept link update included on the OFS CD (refer to the readme.txt file in the ConceptLink directory).

For Concept 2.5, the link is supplied with the Concept product.

Otherwise, an error message will be displayed in the OFS diagnostic window when the table of symbols is loaded and the server will not operate properly.

2.4 Defining the parameters for Local or Remote Operation

Defining the parameters for Local or Remote Operation

Description

The OFS server can operate in:

- Local Mode: the server and OPC client are both on the same machine => no additional parameter definition is necessary,
- Offset Mode: the OPC client and the server station are on different machines connected by DCOM, generally on an Ethernet TCP-IP network => an additional parameter definition (See DCOM Configuration, p. 165) is necessary.

2.5 Uninstalling the OPC Factory Server

Uninstall

Description

To uninstall OFS, open "Add/Delete programs" in the configuration panel and delete the installed components:

- OF Configuration tool
- OPC Factory Server
- OPC Factory Server Remote Station

Note: The parameters defined by the configuration tool and stored in the registry are not deleted.

Configuration Tool

3

Introduction

Aim of this Chapter

The aim of this chapter is to introduce the OFS product Configuration Tool.

What's in this chapter?

This chapter contains the following Sections:

3.1 Introducing the Configuration Tool

OFS Configuration Tool

Introduction

The OFS server is a Data Access OPC Server that can be used to read or write data to DEVICES (usually but not exclusively PLCs).
To do this, the server must have the following information on each device:

- the network to be used,
- the address of the device on this network,
- the table of symbols file to be used if certain symbol conversions need to be performed.

In addition, the server supports a set of configuration parameters to fine-tune the communication with the devices.
All these parameters are handled by the Configuration tool, which is thus an essential component of the OFS product. It enables the user to configure the OFS server to interface with Networks, Devices and Symbol Tables.
The OFS Configuration Tool has 2 usage levels:

- **Normal mode** which provides the device declaration services (address and dedicated parameters). This mode fulfils most end-user requirements and should be chosen if you are not very familiar with the OFS server,
- **Advanced mode** which provides, in addition to normal mode, access to all global parameters. These parameters should only be modified if the user has thorough knowledge of the OFS server.

As a general rule, most of the default values should not be modified.
To use the server, the first thing to do is create one **alias** per device you plan to access.

An **alias** is a shortcut which can be used whenever the network address is of your device is required (simple string replacement). The use of an alias is also a very practical way of dissociating your OPC application from the network addresses of the devices which can be modified if required.

Since the server has symbolic support features, you have the option to specify to the server the name and the full path of the symbols table file that should be used (one for each device). It enables the symbol browsing feature for your device.
You can then also configure other device parameters through the Device Property page. This step is NOT obligatory. In the majority of cases, it is advisable to keep most of the default values.

Note: All modifications made to the configuration parameters of the server are static: for them to be taken into account, the server should be stopped and then restarted.

3.2 Running the Configuration Tool

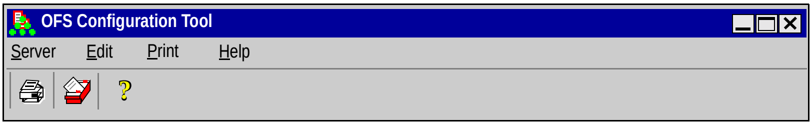
Configuration Tool Execution

Description

To launch the OFS Configuration Tool:

- click on the Start button in the task bar,
- select "Programs"\"Modicon Telemecanique"\"OFS configuration tool".

The upper part of the window offers a set of menus and a tool bar:



The upper part of the window offers a set of menus and a tool bar.

- Server menu:
  - The "**Get Archive**" option will allow you to retrieve a configuration from a backup file. See also the compatibility paragraph (See Compatibility with Previous Version of Configuration Tool, p. 62)
  - the "**Archive aliases**" option allows you to save the aliases and their properties in a file. This option is recommended if a large number of aliases has been declared. It is therefore possible to restore the list, whatever the circumstances,

Note: The archive file contains only the aliases and their properties. The remaining parameters in the other tabs are not in the back-up file (stored in registry only).

- Edit menu: access to cut, copy, paste and sort functions. This last function lists the aliases in the grid according to the sort criterion chosen, from lowest value to highest value (FALSE value then TRUE value for Boolean criteria). All the properties of the alias can be chosen here as criteria,
- Print menu: all parameters can be printed or flushed on a text file.

Note: If a previous version of the configuration tool was installed and aliases already created, a compatibility dialog box will pop up at first execution of the program, which allows existing aliases to be restored. See the Compatibility (See Compatibility with Previous Version of Configuration Tool, p. 62) paragraph for further details.

3.3 Standard parameters and Alias edition

Introduction

Aim of this Section

This section offers the user an introduction to standard parameters and alias edition.

What's in this section?

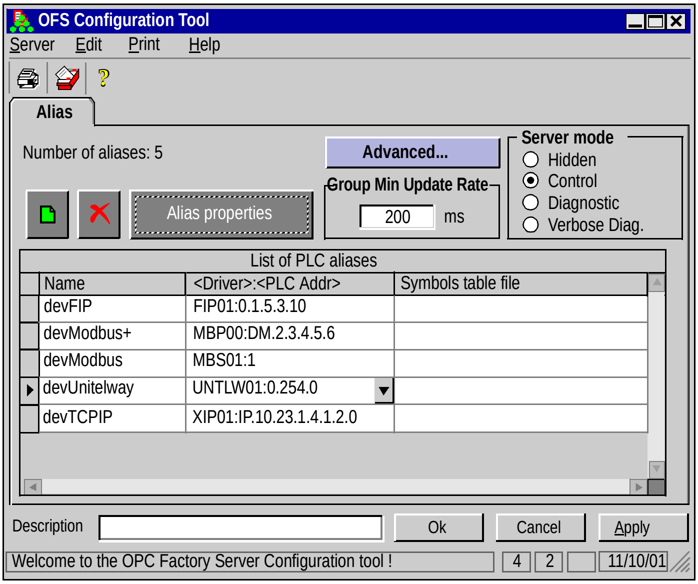
This section contains the following topics:

Introduction to Standard Parameters and Alias Edition

At a Glance

The OFS Configuration Tool main window shows the aliases listed in a grid, the "**Server Mode**" option and the "**Group Min Update Rate**". This information covers most user requirements.

Illustration:



Number of aliases: 5

Group Min Update Rate: 200 ms

Server mode:
○ Hidden
● Control
○ Diagnostic
○ Verbose Diag.

List of PLC aliases

Name	<Driver>:<PLC Addr>	Symbols table file
devFIP	FIP01:0.1.5.3.10	
devModbus+	MBP00:DM.2.3.4.5.6	
devModbus	MBS01:1	
devUnitelway	UNTLW01:0.254.0	
devTCPIP	XIP01:IP.10.23.1.4.1.2.0	

Welcome to the OPC Factory Server Configuration tool ! | 4 | 2 | | 11/10/01

The table below describes the fields of the above screen:

Server mode	**Hidden**: the server is invisible on screen. **Control**: the server is iconized but only the system menu is accessible (right mouse click). **Diagnostics**: a complete set of diagnostics windows is displayed during server execution, including a plotting window displaying warning and/or error messages. **Verbose diag.**: the plotting window displays detailed information messages in addition to errors and warnings. The rest is identical to "Diag." mode.
Group Min Update Rate	Minimum update rate allowed for groups. The update rate that will be set by the client for a given group should also be a multiple of that value. Numerical value in ms. Range: [10..32767]
"Advanced" button	Displays seven additional tab pages for advanced users (see next chapter).

Description	Here the user may enter up to 50 characters of text in order to identify the configuration of the aliases entered. This text appears at the top of the print file. If you make a back-up copy, it is saved in the chosen file. It does not figure in the configuration of the OFS server.
OK	Configuration validation and exiting Configuration tool.
Apply	Configuration validation without exiting Configuration Tool.
Cancel	Canceling modifications made after launch of the Configuration tool or enabling of the "Apply" button. A confirmation request message is displayed.

The aliases grid comprises 3 columns which contain the most important information about each device.

Creating an alias involves:

- naming,
- defining the network address of the device, which includes the network driver and the device address,
- providing a Symbols table file name (optional),
- setting the alias properties which are related to how the server will behave towards the variables created on that alias.

These operations are all accessible via click-sensitive areas in the grid and the "Alias properties" button.

Note: The information on the number of aliases declared is displayed. The "Create new alias" and "Clear current alias" buttons allow an alias to be added to the grid and the selected alias to be deleted.

	CAUTION
⚠	**Using the aliases** A device should be associated with a single and unique alias. If two aliases point to the same device and are used simultaneously, the communication will malfunction. The properties will be the same for both, and set to the properties of the alias first used to create an item. Using an alias and accessing the same device directly from the address will have similar results. **Failure to observe this precaution can result in injury or equipment damage.**

Sort function in the grid:
The aliases can be arranged in the grid according to the parameters associated with the alias:

- Name of alias, driver, symbols table file. Simply click on the upper strip of the grid in the appropriate column. Arrangement in alphabetical order.
- Other parameters : activate the "Edit ->Sort" menu and select the parameter you want. They are arranged alternately in increasing, or (by clicking again) decreasing order of values (e.g. : for a binary value parameter, increasing signifies all the aliases with the value 0, then all those with the value 1).

Backing up configuration data

At a Glance

The back-up function offered in the configuration tool menu enables you to save/ restore the complete list of aliases and their parameters, i.e. the data that is accessible from the Alias tab. However, the parameters from the advanced mode tabs are not backed up.

The print function enables you to save the ENTIRE configuration, either in printed form or in a text file. Automatic loading of the printed data is of course not possible.

Editing the Device Network Address

Description

The Configuration tool offers a guided network & address edition function, so as to avoid the tedious manual edition of alias strings, which would require the memorization of the characteristics of each individual protocol and of the various options available.

The Alias edition dialog box is obtained by clicking in the selected alias area in the grid:

Tree-view	**1st level**: possible protocols, including MODBUS and XWAY. **2nd level**: drivers.
Alias address	Displays the alias string as the result of the selections. In Read Only for XWAY or MODBUS protocols, in Read/Write for other protocols.
Reset	Erases the string.
Cancel	Exit screen without taking choices made into account.
Confirm	Exit screen taking into account the choices made.

The tabs on the right provide XWAY and MODBUS masks to facilitate data entry. They are selected automatically according to the driver chosen in the tree -view.

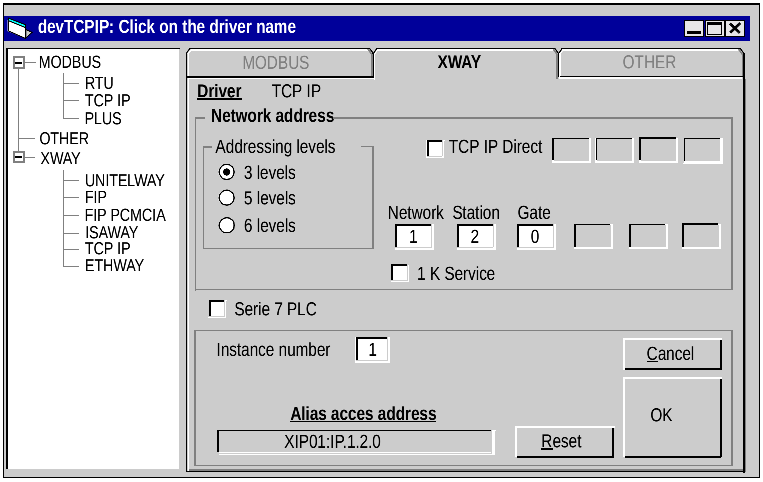
Beyond the XWAY and MODBUS families, the configuration tool is open to any protocol extension:

- it is possible to edit networks and addresses for protocols which are not in the list by selecting the "OTHER" family,
- The list of protocols may be extended in the future, as the tree-view is supplied with external data from OFS server installation.

If "unknown" drivers are used, the displayed tab OTHER, is blank. The edit box at the bottom then becomes **editable** to allow the corresponding string to be entered (rough editing mode).

XWAY Parameters:

Illustration of XWAY (See XWAY addressing modes, p. 125) addressing modes:



Driver TCP IP

Network address

Addressing levels

● 3 levels

○ 5 levels

○ 6 levels

□ TCP IP Direct

Network 1 Station 2 Gate 0

□ 1 K Service

□ Serie 7 PLC

Instance number 1

Alias acces address XIP01:IP.1.2.0

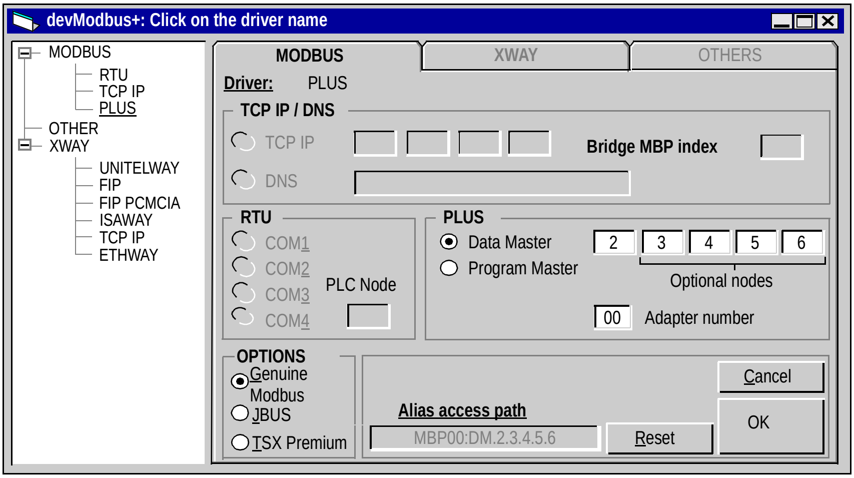
TCPIP Direct	TCPIP driver only: If selected, enter an IP address.
Addressing levels	Addressing level of XWAY address. See Communication (See XWAY addressing modes, p. 125) section.
Network/ Station/Gate	XWAY address. The 3 unlabelled text boxes are grayed out depending on the addressing level selected. For more details on XWAY addressing, see the section entitled Communication (See XWAY addressing modes, p. 125).
Instance number	One instance per driver installed. Generally set to 1. Each driver corresponds to a communication card in the PC.
1K Service	Only available with XWAY on TCPIP and ETHWAY, This option activates gate 7 (UNITE server launched on the Ethernet module). It offers enhanced performance (frame size 1024 bytes). But demands that the PLC application is periodically and not cyclically defined. Moreover, the data is no longer accessed synchronously with the PLC cycle. In certain cases this can lead to data consistency problems.

Serie 7 PLC	Check this box to access PLCs in the TSX Series 7 or TSX S1000 range. The OFS server then interacts with the device in UNITE V1 mode rather than UNITE V2 (used for TSX Premium). It is strongly advised that you check the "device verification" box in the alias properties page, particularly for 5000 and 7000 PLCs which are capable of exchanging frames of 1Kb. **Warning** do not check this box when you mean to access TSX Premium and Micro. Communication would take place, but services are a lot more restricted in Series 7 mode (only internal variables are accessible and performance levels are reduced).

Modbus parameters:

The Modbus addressing principles are explained in the section entitled Communication (See MODBUS addressing modes, p. 129):

Illustration:



devModbus+: Click on the driver name

Driver: PLUS

PLUS

● Data Master 2 3 4 5 6

○ Program Master

00 Adapter number

OPTIONS

● Genuine Modbus

○ JBUS

○ TSX Premium

Alias access path MBP00:DM.2.3.4.5.6

TCPIP/ DNS	Modbus TCPIP only: **TCPIP**: if this option is selected, enter an IP address. **DNS**: if this option is selected, enter a device name. **Bridge MBP index**: Modbus+ node for Modbus TCPIP / Modbus+ bridge.
RTU	Modbus RTU only: Select the COM port used on your computer and the address value of the target device (node). **PLC node**: address value

PLUS	Modbus PLUS only: **Data Master**: limited rights (read/write variables). **Program Master**: full rights (read/write variables, program and configuration modification). Enter the Modbus address (the first value is compulsory, the others are optional depending on routing levels)
Options	**Genuine Modbus**: Modbus device. **JBUS**: JBUS device connected to the MODBUS network. Inserts /J after the device address. **TSX Premium**: Premium PLC. Inserts /T after the device address. **Note**: when using MODBUS TCP IP, the IP address of the PLC must not be declared in the XIP driver if is active.

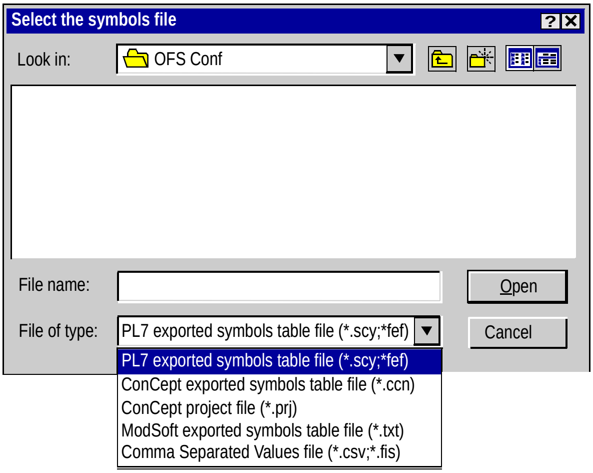
Associating a Symbols Table File

Description

A symbols table file can be associated with the alias, in order to provide access to the symbols for the variables of this device. The symbols file is generated by the PLC programming software, e.g. PL7 for Premium/Micro or Concept for Quantum.

For devices in the Series 7 and S1000 range ("Series 7 PLC" box checked in the XWAY address entry screen), the symbols file can be obtained in the same way as for a Premium, but the application must first have been converted to Premium format. The only restriction is that no consistency check will be possible with the application loaded in the PLC.

Clicking in the "Symbols table file" area for the selected alias in the grid brings up a file explorer:
Illustration:



Select the symbols file

Look in: OFS Conf

File name:

File of type: PL7 exported symbols table file (*.scy;*fef)

?
Open
Cancel
PL7 exported symbols table file (*.scy;*fef)
ConCept exported symbols table file (*.ccn)
ConCept project file (*.prj)
ModSoft exported symbols table file (*.txt)
Comma Separated Values file (*.csv;*.fis)

The file types that can be inserted are listed in the "type of files" list box. Select the appropriate file type.
Enter the file of your choice and click "Open". The file name and directory will then be displayed in the grid.

Caution: adding an FEF-type Premium symbols file to an alias with a Modbus address causes the OFS server, which thinks it is dealing with a Premium, to behave as if option /T (See Description, p. 129) were checked.

Setting the alias properties

Definition

Now that the alias has an address, it is time adjust its property settings. These parameters will allow the behavior of the server to be adapted for the associated alias.

On the one hand, all variables created on this alias, such as:

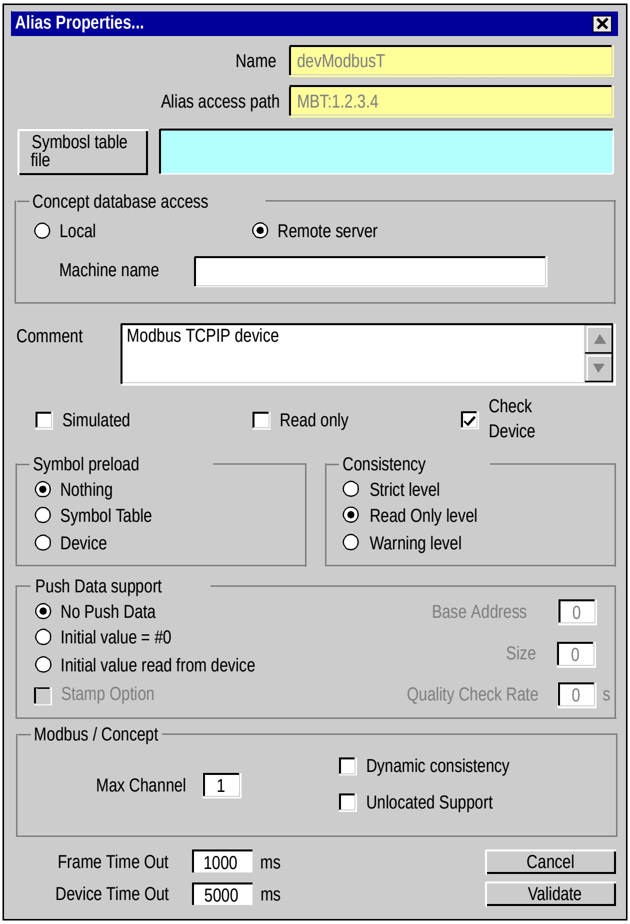
- Use of a symbols table file,
- Access rights to the variables,
- Simulation instead of real access to the device,
- Consistency check between variables and data base,
- Frame exchange timing controls,

On the other hand, parameters on the device itself, such as:

- pre-load function for better performance during alias run-time,
- number of channel reservations for Modbus,
- automatic data write operations from the device itself.

The Configuration Tool provides a properties dialog box for this purpose. Select the alias line in the grid and click on the "Alias Properties" button. The dialog box below pops up:

Illustration:



Alias Properties...

Name: devModbusT

Alias access path: MBT:1.2.3.4

Symbosl table file

Concept database access

○ Local ● Remote server

Machine name

Comment: Modbus TCPIP device

□ Simulated □ Read only ☑ Check Device

Symbol preload

● Nothing
○ Symbol Table
○ Device

Consistency

○ Strict level
● Read Only level
○ Warning level

Push Data support

● No Push Data
○ Initial value = #0
○ Initial value read from device
□ Stamp Option

Base Address: 0
Size: 0
Quality Check Rate: 0 s

Modbus / Concept

Max Channel: 1

□ Dynamic consistency
□ Unlocated Support

Frame Time Out: 1000 ms

Device Time Out: 5000 ms

Symbols table file	Name and path of the symbols table file. See dedicated section (See Associating a Symbols Table File, p. 43). It can be entered and changed here, or directly from the grid. The size of the character string is limited to 255 characters.
Concept database access	**Local**: Concept is installed locally, on the same machine as the server (or simulator) **Remote server**: Concept is installed on a remote machine connected to the server via DCOM. Its remote address must be entered in the text box which is displayed (IP address or character string). **Machine name**: the size of the character string is limited to 30 characters and the total size of the remote simulator and symbols table file character strings must not exceed 255.

Simulated	No physical connection is made to the device. The variables are simulated directly by the server. **N.b.**: the driver associated with the alias must be installed and running on the PC. See also the simulation (See Simulator Tab, p. 58) tab
Read only	All variables relating to the device are read only.
Check Device	*Not available for devices in the Series 7, S1000, Modbus on Premium and JBUS ranges.* The **Check Device** option is activated the first time an item is created for the given device, and performs the following operations: ● It verifies that the device is present. If the device is not present, item creation on this device is refused whatever the level of consistency chosen. ● For MODBUS devices, OFS determines if the Concept IEC runtime is loaded. ● Then, if a symbols file is provided, and if the device allows it (see consistency option below), OFS performs a static consistency check between the PLC application and the symbols table or the project file.
Symbol preload	Allows data to be pre-loaded at server start-up rather than during execution: **None**: by default. **Symbol table**: pre-loads the symbol table. **Device**: creates the connection to the device + loads symbol table.
Consistency	*Not available for devices in the S7 and S1000 ranges.* Definition of restrictions if the symbols consistency check referred to above is negative. **Strict level**: the creation of items is refused **Read Only level**: the items are created but are read only **Warning level**: a warning appears in the diag. window when an item is created, and the items are created in read/write.
Push data support	*Not available with devices in the S7 and S1000 ranges and UNITELWAY, ISAWAY, XWAY, TCPIP direct and Modbus+ networks.* Write orders from the device to the server. For more information on this subject, go to the push (See Push Data Support, p. 159) data section. **N.b.**: if "No Push Data" is selected, other fields are not significant.
Max Channel	*Modbus specific.* Number of channels allocated to Modbus+ and Modbus TCPIP devices. Values between [1.16] for Modbus TCPIP or [1 . 4] for Modbus+. Recommended value: 2.
Dynamic consistency	*Modbus on Quantum specific.* This function is described in detail in the dynamic consistency (See Dynamic Consistency, p. 155) chapter **N.b.**: the dynamic consistency check rate is set as a single value for all devices, in the COMM (See Communication Tab, p. 60) tab.

Unlocated support	*Concept specific.* Enables support of unlocated variables. This option must be activated when the user wishes to use the access to unlocated variables function. For a PLC programmed with a previous version of Concept 2.5, the "check device" box must be checked. For more information on this subject, go to the concept (See Concept Link, p. 143) link section.
Frame Timeout	Permissible delay between request and answer. Range: [1000..10900], maximum a third of Device Timeout.
Device Timeout	Delay for device graph transitions. Range: [3000..32767], at least three times the Frame timeout. For more information on this subject, see frame and device timeout (See Adjusting timeout item values, p. 120).

An alias has now been created which is addressed to a device on the network, and which has all the properties to be applied to all variables created on this device by the server.

Note: For S7 and S1000 PLCs, item creation is only possible if the device is present, unlike other PLCs which allow the creation of items if the PLC is absent and if the "PLC verification" box is not checked. This restriction is linked to the obligation to carry out hardware recognition in order to find out if the device is a Series 7 or S1000, request generation being slightly different on each of the devices.

3.4 Setting the advanced parameters

Introduction

Aim of this Section

The aim of this section is to describe how to set the advanced parameters.

What's in this section?

This section contains the following topics:

Introducing Advanced Parameter Configuration

At a Glance

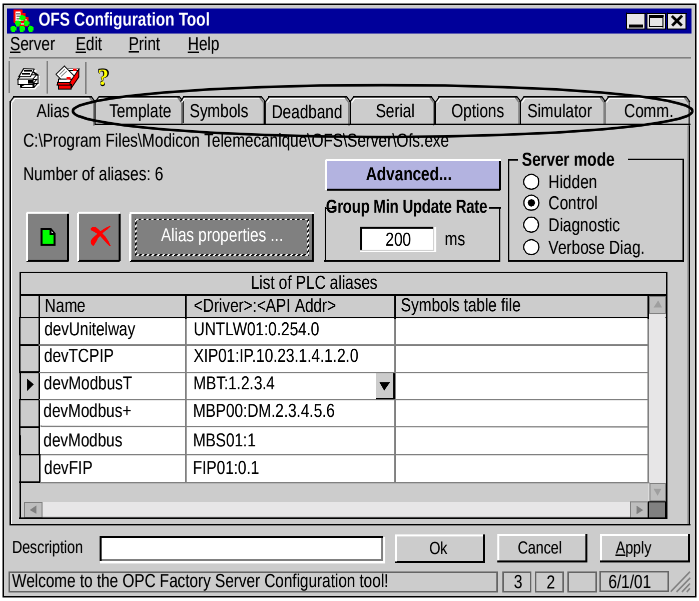
At this stage, one or more aliases are created and the associated properties are set. The client application can create variables on the devices associated to these aliases. In the majority of cases this will be sufficient.

Nonetheless, in addition to the alias creation functions, certain global parameters (which are not specific to one device but to the general operation of the server or configuration tool) may be defined.

- default properties applied to the aliases when they are created,
- variation range of variables settings for deadband customization (percentage of range variation),
- serial line configuration for Modbus,
- PLC operating mode rights,
- server shut down delay,
- diag trace displays,

Click on the "Advanced" button. Seven tabs are added to the "Alias" default tab (main window):

Illustration:



C:\Program Files\Modicon Telemecanique\OFS\Server\Ofs.exe

Number of aliases: 6

Group Min Update Rate: 200 ms

Server mode: ○ Hidden ● Control ○ Diagnostic ○ Verbose Diag.

List of PLC aliases

Name	<Driver>:<API Addr>	Symbols table file
devUnitelway	UNTLW01:0.254.0	
devTCPIP	XIP01:IP.10.23.1.4.1.2.0	
devModbusT	MBT:1.2.3.4	
devModbus+	MBP00:DM.2.3.4.5.6	
devModbus	MBS01:1	
devFIP	FIP01:0.1	

Description

Welcome to the OPC Factory Server Configuration tool! 3 2 6/1/01

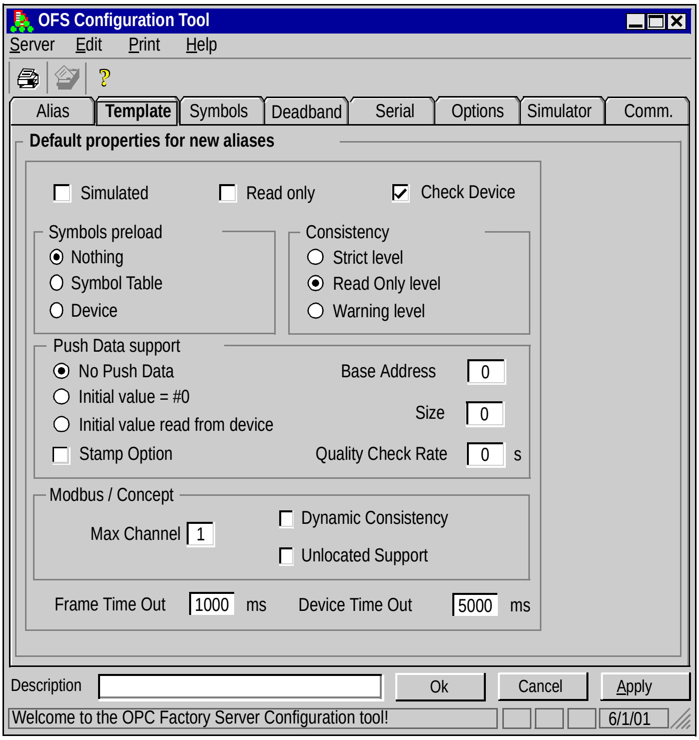
- Template Tab: allows default values of alias properties to be modified when a new alias is created,
- Symbols tab: editable list of file extensions associated with symbols table file,

- Deadband tab: edition of variable range values (deadband is a percentage of this range, set at group creation),
- Serial tab: Modbus RTU serial line configuration,
- Options tab: additional server options (execution of a command file during server shutdown, debug options, etc.),
- Simulator tab: configuration of simulated variables value variation,
- Communication tab: advanced parameters for communication

Template Tab

Description

This tab lists all of the alias properties applied by default when an alias is created. Illustration:



Default properties for new aliases

☐ Simulated ☐ Read only ☑ Check Device

Symbols preload
● Nothing
○ Symbol Table
○ Device

Consistency
○ Strict level
● Read Only level
○ Warning level

Push Data support
● No Push Data
○ Initial value = #0
○ Initial value read from device
☐ Stamp Option

Base Address 0
Size 0
Quality Check Rate 0 s

Modbus / Concept
Max Channel 1
☐ Dynamic Consistency
☐ Unlocated Support

Frame Time Out 1000 ms Device Time Out 5000 ms

Welcome to the OPC Factory Server Configuration tool! 6/1/01

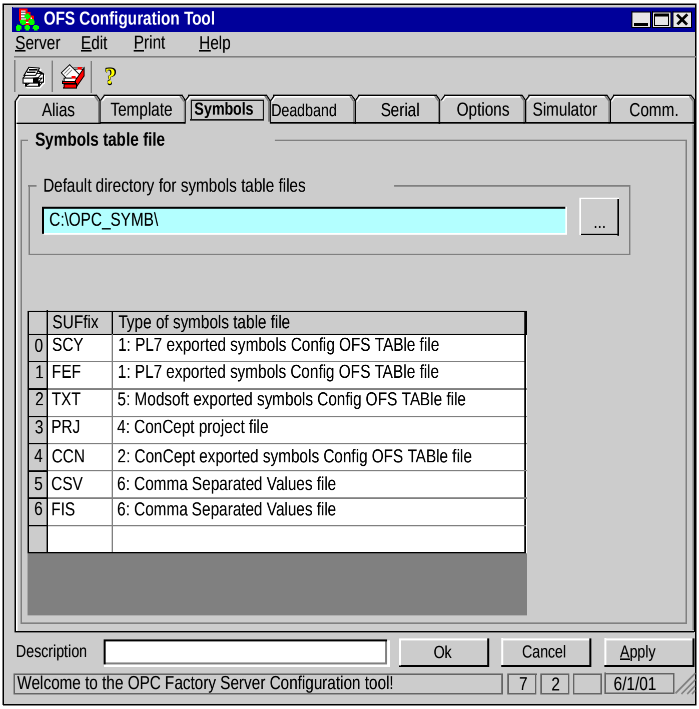
A full set of default parameters of your choice can be set up so that property adjustment for each new alias created is minimized.

Symbols Tab

Description

This tab provides a list of file name extensions associated with the symbols tables. This list can be completed with new extensions (up to a maximum of 10 suffixes). The extensions are memorized, even when the server is uninstalled and/or reinstalled. However, for this to happen, one condition must be observed: they must be entered when the grid contains extensions which have already been set by the server. If you are starting out with an empty grid (server never installed), they risk being overwritten during installation of the server.

Illustration:



Symbols table file

Default directory for symbols table files: C:\OPC_SYMB\

	SUFfix	Type of symbols table file
0	SCY	1: PL7 exported symbols Config OFS TABle file
1	FEF	1: PL7 exported symbols Config OFS TABle file
2	TXT	5: Modsoft exported symbols Config OFS TABle file
3	PRJ	4: ConCept project file
4	CCN	2: ConCept exported symbols Config OFS TABle file
5	CSV	6: Comma Separated Values file
6	FIS	6: Comma Separated Values file

Description

Welcome to the OPC Factory Server Configuration tool! | 7 | 2 | | 6/1/01

To define or edit an extension (generally three letters without a full stop):

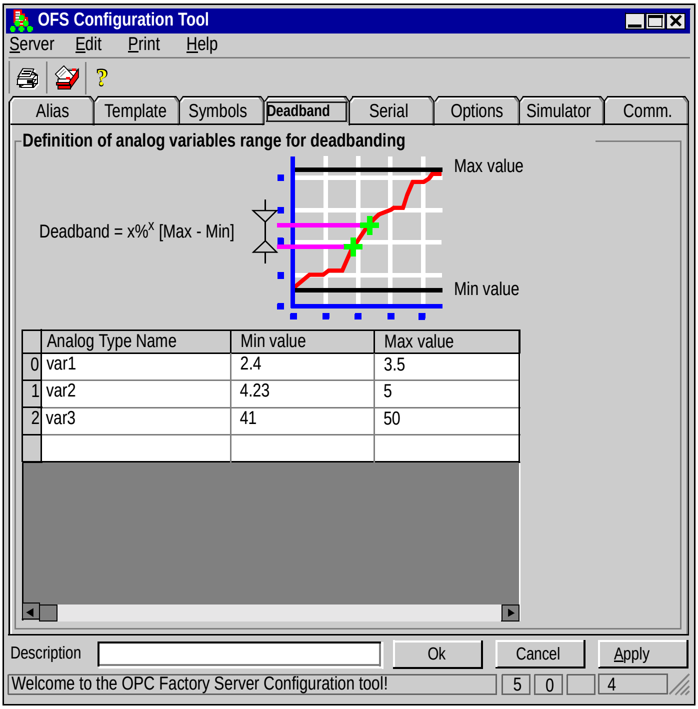
- double click on an empty or occupied field in the "SUFfix" column. Enter an extension, then press "return",
- double click on the corresponding field in the "Type of symbol file" column. A list appears. Select a file type, then click OK,
- an additional line appears at the bottom to allow another definition.

Note: Do not specify the dot separator to the left of each extension.

Dead Band Tab

Description

The Dead band is a percentage of the range of values that an analog variable can take. It is part of the Group attribute, is applied to all variables of that group, and will be considered as the criteria for notifying when the value changes.
Illustration:


OFS Configuration Tool
Server Edit Print Help
Alias Template Symbols Deadband Serial Options Simulator Comm.
Definition of analog variables range for deadbanding
Max value
Deadband = x%X [Max - Min]
Min value
Description
Ok Cancel Apply
5 0 4

	Analog Type Name	Min value	Max value
0	var1	2.4	3.5
1	var2	4.23	5
2	var3	41	50

Welcome to the OPC Factory Server Configuration tool!

The range can be adjusted here for each variable, whether floating or integer, with minimum and maximum values.

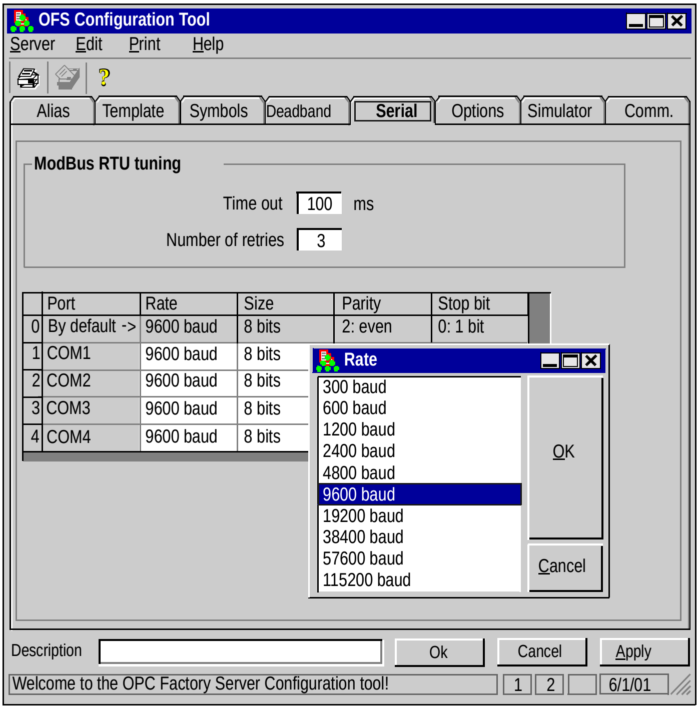
Note: The Configuration tool will not allow a value less than that entered in the "Min value" field to be entered in the "Max value" field.

Serial Tab

Description

Within this tab page, the user may define or edit all the parameters that are dedicated to the Modbus RTU protocol.

This protocol is supported for the usual 4 PC serial ports, COM1 to COM4. Illustration:



ModBus RTU tuning

Time out 100 ms

Number of retries 3

	Port	Rate	Size	Parity	Stop bit
0	By default ->	9600 baud	8 bits	2: even	0: 1 bit
1	COM1	9600 baud	8 bits		
2	COM2	9600 baud	8 bits		
3	COM3	9600 baud	8 bits		
4	COM4	9600 baud	8 bits		

300 baud
600 baud
1200 baud
2400 baud
4800 baud
9600 baud
19200 baud
38400 baud
57600 baud
115200 baud

Welcome to the OPC Factory Server Configuration tool! 1 2 6/1/01

Time out	Positive numerical value: range [100..32767] Response wait time after transmission of a frame on the network. If several transmission attempts are made (see parameter below), the wait time is applied for each of them. **Caution**: If the value provided is less than (15480000/speed), the result of this division is applied (maximum calculated time of the return journey of a frame of maximum size)
Number of retries	Numeric value: value [1..10] = number of send retries before a time out error is returned.

For each area in the grid, a double click displays a list box, shown above, which provides the list of possible values that can be entered for the serial line parameters, i.e.:

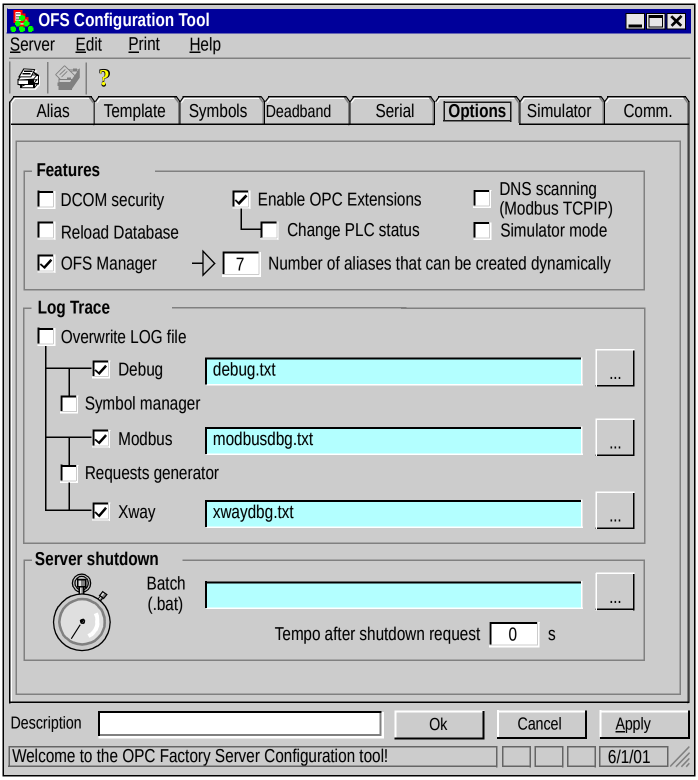
- rate (300 to 115200 baud),
- size of characters (4 to 8 bits),
- parity (no parity, odd, even, mark, space),
- stop bit (1, 1.5, 2 bits).

Caution: the default parameters proposed correspond to the configuration required for TSX Quantum PLCs.

Options tab

Description

This tab page allows the optional functions of the OFS server to be activated. Illustration:


OFS Configuration Tool
Server Edit Print Help
Alias Template Symbols Deadband Serial Options Simulator Comm.
Ok
Cancel
Apply

Features

□ DCOM security
□ Reload Database
☑ OFS Manager
☑ Enable OPC Extensions
□ Change PLC status
□ DNS scanning (Modbus TCPIP)
□ Simulator mode
7 Number of aliases that can be created dynamically

Log Trace

□ Overwrite LOG file
☑ Debug: debug.txt
□ Symbol manager
☑ Modbus: modbusdbg.txt
□ Requests generator
☑ Xway: xwaydbg.txt

Server shutdown

Batch (.bat):

Tempo after shutdown request 0 s

Description:

Welcome to the OPC Factory Server Configuration tool! 6/1/01

DCOM security	Activates/Deactivates DCOM security
Reload database	If checked and if Device Verification is activated, when an inconsistency with Concept is detected, the database will be reloaded automatically. Otherwise, the reload will not be performed. It can be performed manually using the "reload and update" service in the OFS manager (See The OFS Manager, p. 192).

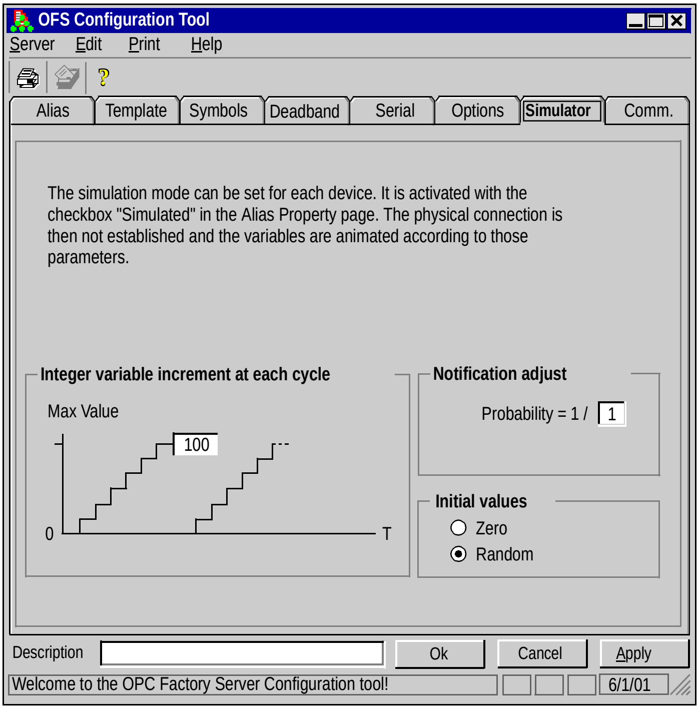
OFS Manager	Authorizes dynamic reconfiguration without restarting the server. The OFS manager allows certain additional aliases to be created dynamically. If activated, a window appears to set the number. Range: [0..99]
Enable OPC Extensions	Activates/Deactivates the specific items (See Specific Items, p. 150).
Change PLC Status	If "Enable OPC Extensions" only is checked. It allows the server to change the PLC operating mode (RUN/ STOP).
DNS scanning	Modbus TCPIP only, Authorizes the server to use DNS to identify the PLC.
Simulator mode	The server is launched in simulator mode, i.e. as if all alias parameters were set in simulated mode. A license code is not required in simulator mode. Simulator mode is shown in the diagnostics window at server startup.
Log Trace Files	**Debug, Modbus, XWAY**: activates independent trace files for each mode. When this option is selected, a text box is displayed for entering a log trace file name and path. **Symbol manager**: for Debug only. **Requests generator**: for XWAY and Modbus only. Displays all the information on requests generated by the server (compiling process, contents, etc.)
Shutdown batch file	If a .BAT file is entered here, it will be executed when program exit is requested (the tempo prior to shutdown, if different from 0, will not be counted down until afterwards)
Tempo after shutdown request	Programmable delay in seconds between server shutdown notification and actual shutdown. Range: [0..32767]

Note: If the "Change PLC Status" box is not checked, any client application attempting to write on the #PLCStatus item receives an error message "EOL_E_FEATURE_DISABLED".

Simulator Tab

Description

If the alias has been configured with the simulation (See Setting the alias properties, p. 44) property, any variable created on these devices is simulated locally by the server. This tab page defines which value variation to be applied to all simulated variables by the server.
Illustration:



The simulation mode can be set for each device. It is activated with the checkbox "Simulated" in the Alias Property page. The physical connection is then not established and the variables are animated according to those parameters.

Probability = 1 / 1

Initial values
○ Zero
● Random

Welcome to the OPC Factory Server Configuration tool!

6/1/01

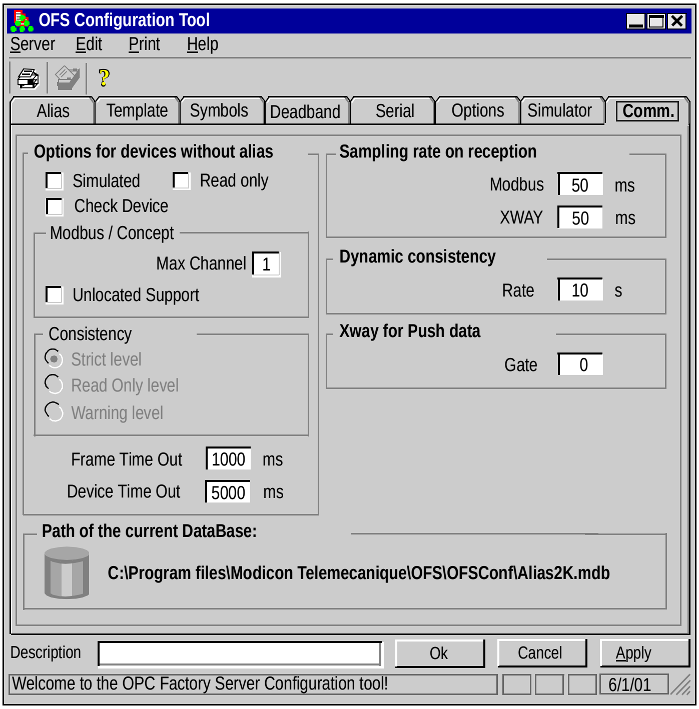
Notification adjust	N=1: simulated variables are updated at the same update rate as the group and each time a "read device" is performed (sync or async) $1 < N \leq 10$: at each period there is one chance in N that the simulated variable will be modified. There is no correlation between different declared variables; their values change individually. The decrease in the probability value (increase of N) reduces the number of notifications, and thus reduces the CPU load on the machine.

Max Value	Maximum value for the simulated variable. Range: [0..32767]. The variable is increased at each cycle, and returns to 0 when the maximum value is reached (cyclically). Boolean variables are inverted, float variables are increased by 0.3.
Initial values	**Random**: the variables are initialized to random values. **Zero**: all variables are initialized to zero.

Communication Tab

Description

The Communication tab page gives access to general synchronization parameters for data exchange with the devices and polling frequency on reception.
Illustration:



Options for devices without alias

□ Simulated □ Read only
□ Check Device

Modbus / Concept
Max Channel 1
□ Unlocated Support

Consistency
● Strict level
○ Read Only level
○ Warning level

Frame Time Out 1000 ms
Device Time Out 5000 ms

Sampling rate on reception
Modbus 50 ms
XWAY 50 ms

Dynamic consistency
Rate 10 s

Xway for Push data
Gate 0

Path of the current DataBase:
C:\Program files\Modicon Telemecanique\OFS\OFSConf\Alias2K.mdb

Description

Welcome to the OPC Factory Server Configuration tool!
6/1/01

Options for devices without alias	Same parameters as in the Properties page. The selections made here are applied only to the devices created without aliases, or aliases created dynamically with OFS manager, at server run time. The "Consistency level" option is only useful for aliases created dynamically with OFS manager.
Sampling rate on reception	Sets the period for checking data reception in milliseconds. Range: [10..32767] Values should be adjusted carefully as they affect the CPU load on your computer.

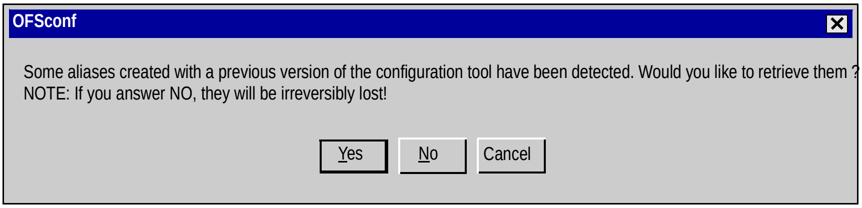
Dynamic consistency	This period sets the time in seconds in which the cyclic consistency check is performed. Range: [1..32767] This is a global value for all devices for which the Dynamic consistency option is activated.
XWAY for Push data	**Gate**: value of reception gate on which the server will receive data from the remote device. Range: [0..255]

3.5 Compatibility with Previous Versions of Configuration Tool

Compatibility with Previous Version of Configuration Tool

Description

If you have already used a previous version of the configuration tool and stored a configuration set in your PC, it will be automatically detected and you will have the option to retrieve the information.
At first execution of the Configuration tool, a dialog box pops up and asks you whether you wish to retrieve the data from the registry.
Illustration:



OFSconf

Some aliases created with a previous version of the configuration tool have been detected. Would you like to retrieve them ?
NOTE: If you answer NO, they will be irreversibly lost!

If you answer:

- YES: the configuration set will be scanned from the registry and inserted in the configuration tool database,
- NO: the aliases in the registry will be lost and replaced by those present in the database (this database is usually empty as the scan is performed at first execution of the configuration tool),
- CANCEL: exits the program, no action is performed.

⚠

CAUTION

Using aliases from an earlier version

If you answer NO, the aliases already in the registry will be permanently erased.
The INIT values in the Template tab are not stored in the registry and will thus be set with the values from the new configuration tool database (generally default values).

Failure to observe this precaution can result in injury or equipment damage.

The save files of aliases created with version 2.2 of OFS are fully compatible with version 2.5.

Product Functionalities

4

Introduction

Aim of this Chapter

This aim of this chapter is to describe the features of the OFS product.

What's in this chapter?

This chapter contains the following topics:

Definition of a group of items

Definition

All the OFS product's services are based on the concept of a group of items: a set of variables on any PLC which can be accessed either by their address or by their symbol.

- Several groups can be defined,
- a group may concern several devices: each item can have a different device address,
- a group concerns various communication devices and media: each item may refer to a different communication driver. If a device can be accessed via several communication media, it is possible to mix variables addressed via different media within one group,
- the items comprising a group can be different: it is possible to mix all types of objects managed by the OFS server,
  Example: mixing words, double words and floating points within one group.
- depending on the service called up, a group is transcribed on one or more communication requests,
- all the items in the same group have the same update rate and deadbanding percentage.

Synchronous Services

Description

- these services are used for **partial** or **complete** reading and writing of a group of items,
- the periodic scanning of the evolution of variables (read polling) must be handled by the client application,
- the term "**synchronous**" means that the client application which calls up these read or write services is blocked for the time it takes to obtain a result. The instruction which follows a synchronous read or write call in the code of the client application will only be executed when all the communication requests corresponding to that call have been processed. This term does not mean synchronization with the device. It means that, during a synchronous read operation, the OFS server **does not guarantee** that all the variables in a group **will be read in the same PLC scan** if this group is **transcribed on several** communication requests. The OFS server provides a mechanism for ascertaining the number of requests necessary to **read the whole** of a group of items (for synchronous groups only).

Note: The conditions ensuring that the items in a group are consistent with one another (read or written in the same PLC scan) are described in read consistency (See Read Consistency, p. 111) and write consistency (See Write Consistency, p. 112).

Asynchronous Services

Description

- these services are used for partial or complete reading or writing of a group of items,
- the periodic scanning of the evolution of variables (read polling) must be handled by the client application,
- the client application is not blocked during the time it takes to obtain the data,
- a notification mechanism (which must be activated) notifies the client of the results,
- For synchronization with the PLC, everything discussed for synchronous (See Synchronous Services, p. 65) services is valid.

Cyclic Read Service

Description

The periodic scanning of variables: read polling and the notification of changes in variable values are performed by the OFS server.

- the client application should program a "Wake up" function, which is called up by the OFS when values which have occurred on items of all periodically examined groups change.

This means that the "Wake Up" function is unique in the client application: it receives all the notifications from the OFS server, then it must redistribute them to the processing functions specific to each periodically scanned group.

Note: For ready-to-run supervisory software, the "Wake Up" function should be pre-programmed. If this is not the case, the notification mechanism may not be used.

The name of this "Wake Up" function is set by the OPC standard **OnDataChange**.

Caution: Processing which takes up a significant amount of CPU time (e.g.: over-complex display) should not be performed in the "Wake Up" function, as this may adversely affect the Operating System's performance.

- the OFS server notifies by group, not individually by item. That means that, for a given group, the OFS server sends the list of items whose value has changed to the client application "Wake Up" function. In the case of a table type item, the OFS server transmits the whole table even if only a subset of the elements has changed in value.

The following notions are associated with the "cyclic" read service:

- Assignment of a scanning PERIOD ("RATE") to a group: time between two read operations during "polling" performed by the OFS server. This concept reflects the need to monitor the variables at different rates.
  Example: display the PLC time every second, and display a temperature every minute.

- Allocation of a dead band to a group (dead banding): filtering of notifications when group variable values change. Notification occurs if, after the group scanning period, variables have changed by more than a certain percentage with respect to their old value, see chapter on dead band (See Installation of the dead band, p. 115)).
  Example: inform the client application only if temperatures have changed by more than 10%.

Note: Dead-banding is only applied to floating point or integer variables. The aim of these two concepts is to enable the user to control (limit) the flow of notifications sent to the client application, to avoid overloading the system.

Adjust Communication Timeout with a device

Description

Various parameters can be used to set this important communication parameter. They can be global or device (See Setting the alias properties, p. 44) specific parameters. They can also be static (configured using the Configuration Tool) or dynamic (configured using a write specific (See Specific Items, p. 150) item (See Adjusting timeout item values, p. 120) and method.

Managing the PLC Operating Mode

Description

This mechanism enables the user to view or modify the operating mode of a PLC:

- startup,
- stop,
- initialization (Premium and Micro only).

If a programming tool is already connected to the device, the request may fail. An error message will be displayed in the diagnostics window of the server (See PLC operating mode management, p. 122).

Note: The OFS server only supplies the basic service for changing the operating mode of a PLC. Since the change of state of the PLC may have repercussions on the process, all preliminary checks are the responsibility of the client application.

Example: Programming a message in the client application requesting the operator to confirm the change of the PLC operating mode.

Access to the server or simulator

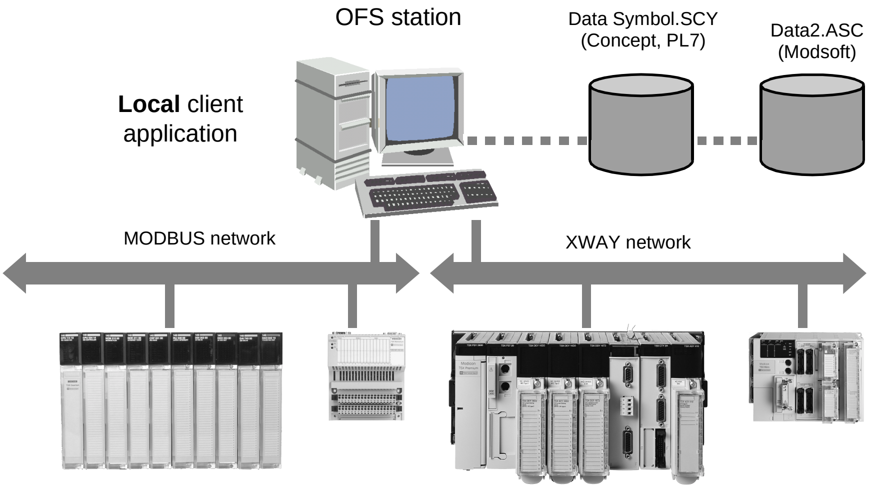
Description

There are two methods for accessing the services provided by the OFS server.

- local access:

The client application and the OFS server are on the same station.

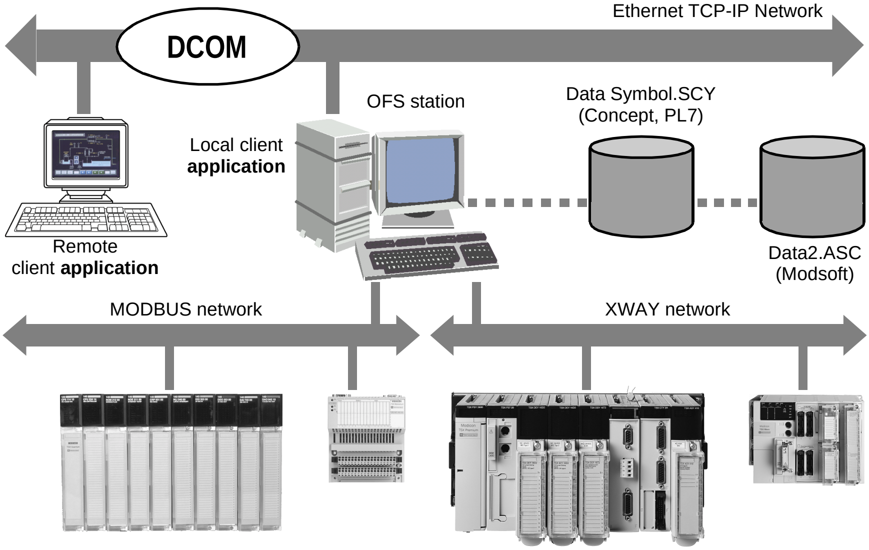
Illustration:



- remote access via DCOM:

The client application and the OFS server are on separate stations connected via the Microsoft TCP-IP network:

Illustration:



> **Note:** DCOM (See DCOM Configuration, p. 165) must be configured correctly before launching remote operation.

Support of symbols

Description

Symbol support enables the user to replace the address of any variable defined in the item by its name in the PLC application (e.g.: use of "Symbl" rather than the topological address "%MW1" or instead of State RAM location "400001"). It is similar to a string substitution and has no effect on Read/Write operations.

The supported symbol (See Symbol management, p. 131) table formats are:

- PL7 symbol table file or exported project file,
- Concept exported symbol table file,
- Concept project file (direct link with Concept database),
- Modsoft exported symbol table file,
- CSV symbol table file (Excel exported format),
- Taylor exported symbol table file (identical to Excel format).

Note: For old ranges: XTEL files must be converted to PL7 format in order to use symbols on the series 7 (using the "PL7-3 converter" function of the PL7 PRO). The series 1000 does not allow the use of symbols.

Symbol consultation

Description

The OPC Browse Interface is supported by the OFS product. This enables users to browse symbols available for a given PLC, provided that the OPC Client used supports the Browse interface. This is an easy way to determine which variables can be created for a given device. Browsing structures and tables are available when the programming language includes these object types (e.g.: Concept programming tool).

Note: Only devices declared with the Configuration Tool and associated with a Symbol Table can be browsed.

Concept link

Description

The direct link with the Concept database enables:

- symbol support,
- symbol browse support,
- access to unlocated variables and to structured data.

Concept and OFS can run simultaneously on the same Concept project. Several Concept (See Concept Link, p. 143) projects can be opened at the same time provided that they are all the same version.

OFS 2.5 supports the following versions of Concept:

- Concept 2.1,
- Concept 2.2,
- Concept 2.5 SR2 (and above).

Concept 2.5 is only partially supported by OFS 2.2 (there is no database consistency verification and no access to unlocated data).

Remote Concept link

Description

The remote link with the Concept database has exactly the same features as the normal Concept link. The only difference is that the server does not need to be located on the same machine. The Concept machine and the server machine should be linked via DCOM (usually over TCP/IP).

This feature should be enabled within the device property page.

A running server is required on the machine on which the Concept programming tool and the Concept project are located. This server acts as a proxy for the "real" server. It can either be a server which is already used as such, that is, connected to certain devices and OPC clients, or it can be a server which is dedicated to this proxy function.

Simulator mode

Description

Simulator mode enables the user to test the application without the actual devices. It provides a simple animation of all created variables. Except for device access, it is identical to the actual Server.

One single restriction: the network drivers which correspond to the aliases used must be installed on the machine.
The server can be started in simulator mode in two ways:

- by selecting the "OFS Factory Server Simulator" shortcut created during installation. It launches OFS.exe with the "-simu" parameter,
- by checking the "simulator mode" option in the "options" tab of the configuration tool.

When the server is started in simulator mode, no license code is required.
The animation of simulated variables can be set in the configuration tool, under the "Simulator" tab.

Note:

- As all the variables are simulated, there is no link between an item which is actually related to a table of elements (bits, words), and the items related to the individual components of this table,
- in simulation mode, there is no way to determine the maximum supported frame length for a given device. This means that the access right for any table item is set to READ_WRITE,
- It is possible that when using the real device an item that was READ_WRITE in simulation mode becomes READ_ONLY in real mode,
- Concept BOOLEAN variables that are located in State Ram in the register areas (3x or 4x) are actually simulated as bytes and not as Boolean values.

Individual simulation of a device

Description

This feature allows the server to simulate a missing device.

The choice between accessing a real device and device simulation is made, device by device, in the properties page of the device (See Setting the alias properties, p. 44).

The conditions for use are the same as for simulator mode (network driver installed, setting the parameters for animation of the variables from the "simulator" tab of the configuration tool).

NT Service

Description

The NT service feature (only available under Windows 2000, NT 4.0) allows the OFS server to start automatically each time the machine boots, and to stop automatically before each shutdown.
In this case, the OFS server icon is not visible; the server runs in the background.
The user may always start and stop the server manually using the usual NT service commands (Windows 2000 control panel, administration tools, Services or NT control panel icon, Services icon).
The OFS server operating in this way may be used locally or remotely by any user (administrator or not).
For further details, see the section on NT (See OFS as an NT service, p. 157).

Dynamic consistency check.

Description

This feature enables the server to verify regularly the consistency between the application loaded into the PLC and the Concept symbols database currently open. When a difference is detected, the Concept database is automatically closed and reloaded. The new symbols and new locations are then taken into account.
For more details, see the section on consistency (See Dynamic Consistency, p. 155).

Push data support

Description

As a general rule, to update OPC items automatically, the server sends some network requests to the device, then waits for the network responses to update its internal data tables. This is called "polling" the device.

In contrast, the "Push data" feature corresponds to the spontaneous transmission of data by the device to the active server, without any request having come from the server.

The data is regarded as being "pushed" by the PLC. This feature is particularly worthwhile when the values of the data being monitored do not change very frequently. Considerable savings can be made on network load and on the server's CPU consumption. This does, however, mean that specific processes have to be included in the PLC application for sending data.

The feature is supported for Modbus TCPIP, XWAY TCPIP (direct excepted), FIPWAY and ETHWAY devices.

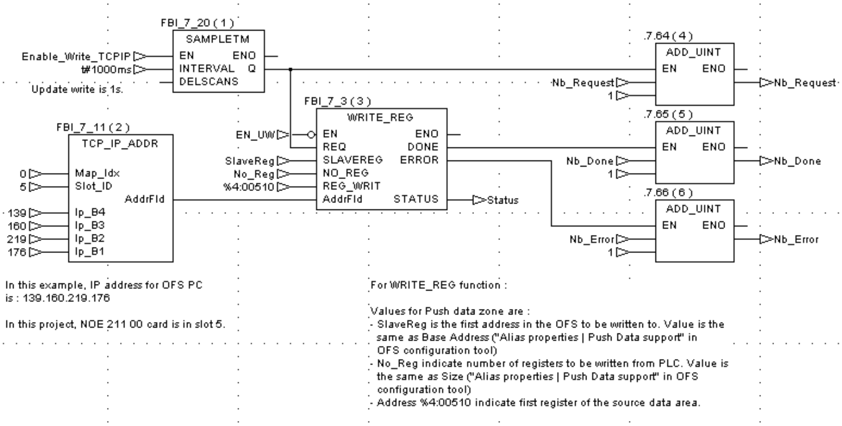
This feature can be enabled and configured device by device using the device (See Setting the alias properties, p. 44) property page.

Data sent by the device to the server should fit within the Push data range that has been defined for this device. Only one range can be defined for each device (using the device (See Setting the alias properties, p. 44) properties page).

Any number of OPC items can be defined within this range. They are seen as ordinary OPC items.

The device has the option (Device properties (See Setting the alias properties, p. 44)) of sending a timestamp with the data, used by the server to update the timestamp property of all items associated with Push data.

Concept example of use of WRITE_REG to test the PUSH DATA feature:



In this example, IP address for OFS PC is : 139.160.219.176

In this project, NOE 211 00 card is in slot 5.

For WRITE_REG function :

Values for Push data zone are :
- SlaveReg is the first address in the OFS to be written to. Value is the same as Base Address ("Alias properties | Push Data support" in OFS configuration tool)
- No_Reg indicate number of registers to be written from PLC. Value is the same as Size ("Alias properties | Push Data support" in OFS configuration tool)
- Address %4:00510 indicate first register of the source data area.

Some PL7 and Concept sample applications are provided on the CD to show how a PLC application can send Push Data to the server.

Definition of the Diag Buffer

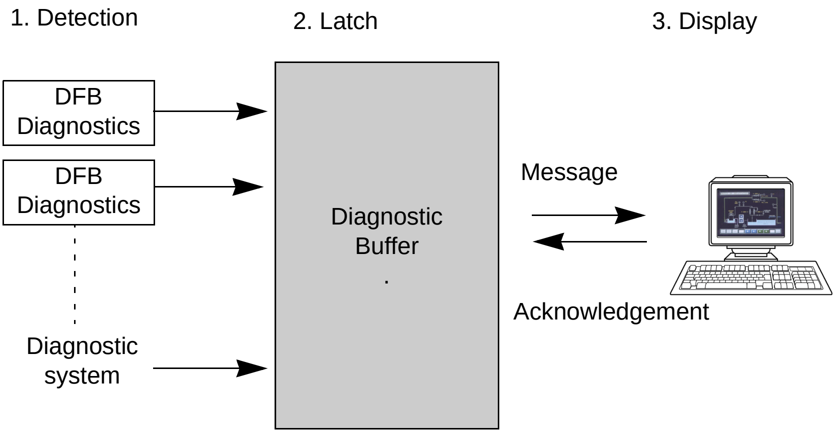
General

The Diag Buffer detects faults on monitored elements and transmits error messages to the display systems.
Diagnostics DFBs only allow the installation of this feature on Premium TSX57/PCX57/PMX57 PLCs which have a minimum software version (see PL7 documentation for more information).

It enables the visualization of an alarm being triggered in real time and gives all the characteristics of the alarm triggered in a bits table:
- type of error,
- start date and time,
- end date and time,
- trigger zone between 0 and 15 (in case several modules are declared on the same PLC),
- alarm comments...

Illustration

The diagram below shows the functioning of the Diag Buffer:



Operation

The table below describes the different functioning phases:

Phase	Description
1	The diagnostics DFB integrated in the applications program or system detect the process fault conditions.
2	A buffer memory called diagnostics Buffer memorizes all faults in time recorded message form.
3	One or several multi-station Viewers (15max) allow: ● visualization of one or several zones of a PLC, ● visualization of one or several zones of several PLCs, ● acknowledgment of error messages, ● visualization of the evolution of the status of an item.

For more information on the Diag buffer, see the section on Diag buffer installation (See Diag Buffer, p. 169).

Variable types managed by the server

5

At a Glance

Aim of this chapter

This chapter gives an exhaustive list of the various types of objects managed by the OFS server, and indicates the access permitted for each type: read only (R) or read/write (R/W).

What's in this chapter?

This chapter contains the following Sections:

5.1 OPC item general syntax

General information on OPC items

General

Before reading or writing values, an OPC item should be created for each device variable.
The general syntax for an OPC item is:
<**item**>::=<**driver name**>:<**device address**>!<**variable definition**>[:<**table length**>|<number of extracted bit>][;<**postfix**>]
The <**driver name**>:<**device address**> part can be replaced by an alias (See Standard parameters and Alias edition, p. 34) created with the configuration tool.

If an alias is not used, the **driver name** must be one of the names given in the following list and the **device address** is the address of the device on the communication medium:

Driver name	Example of device address	Communication medium
[s7]UNTLW01 *	0.254.0	UNI-TELWAY
[s7]FIP01 *, [s7]FIP02 *	0.31.0	FIPWAY adapter 01 or 02
[s7]FPP2001 *	0.31.0	FIPWAY PCMCIA adapter 01
ISAWAY01, ISAWAY02	0.5	ISAWAY adapter 01 or 02
[s7]ETHWAY01 *, [s7]ETHWAY02 *	0.5	ETHWAY adapter 01 or 02
[s7]XIP01 -> [s7]XIP09	0.5	XWAY TCP-IP adapter 01 to 09
MBP00,MBP01,MBP02,MBP03	PM.12 or DM.15.3	MODBUS+ adapter 0 or 1 or 2 or 3
MBT	139.160.218.102	MODBUS TCP-IP
MBS01,MBS02,MBS03,MBS04	6	MODBUS RTU on COM1,COM2,COM3,COM4

*"[s7] means that "s7" should be added to the beginning of the driver name to access series 7 and series 1000 PLCs.

The variable definition part can either be a variable address (see the "Syntax" column in the other tables in this chapter) or a symbol (See Support of symbols, p. 73).
Modbus+ users planning to use Concept and OFS simultaneously, or the multi-channel feature, should use DM mode. Otherwise you may not be able either to connect to the PLC with Concept or download your application.

For variables that support this feature, the table length enables the user to create items that are actually tables and gives the number of elements making up this table.

The Postfix part can be R: R means READ ONLY and is a way of creating an item that will always be considered as READ ONLY.
For Modsoft (See Modsoft variables, p. 101) variables, F and D postfix values are also supported.

Note: The **driver name**, **device address** and **variable definition** parameters are mandatory.
The **table length** and **postfix** parameters are optional.
Examples:

- UNTLW01:0.254.0!%MW3
- MBS01:12!400003
- FIP01:0.31.0!%MW5
- MBP00:DM.5!400005
- XIP01:0.5!%MW100
- MBT:1.2.3.4!400100
- TSX1!%MW100
- QTM1!400100
- TSX2!toto
- QTM2!toto

The MBT **device address** field uses the postfix ";xx" to designate the destination index used in the Modbus+ mapping table defined in the Ethernet to Modbus+ bridge. "xx" is the decimal value of the index. For example, "139.160.218.103;50".

5.2 OPC Data Types

Different OPC data types

Description

The OPC data type handled by the OFS client (called "expected") can be different from the native data type of the variable within the device (called "canonical").
By default, at item creation, they are identical. However, the user can choose another type.
More specifically, conversions between canonical type arrays or 16 bit words and expected type character strings are supported, giving the user easy handling of character strings with PLCs (these do not have canonical type character strings):

- the conversion byte array -> produces an ASCII string,
- the conversion 16 bits word array -> produces a UNICODE string.

5.3 PL7, XTEL and ORPHEE variables

At a Glance

Aim of this Section

The aim of this section is to introduce you to the different PL7 variables on OFS. Only memory objects of standard objects are accessible for series 7 (XTEL) and series 1000 (ORPHEE) PLCs. The syntax used on these PLC ranges have been recovered and are highlighted in Italics. They are only accessible with these types of PLCs.

Meanings of the terms used in the tables:

- R: read
- W: write

What's in this section?

This section contains the following topics:

Standard objects

Definition

- System objects:

Object	Syntax	TSX 37	PCX/TSX 57
System bit	%Si	R/W	R/W
System word	%SWi	R/W	R/W
System Double word	%SDi	R/W	R/W

Note: Some of these objects can only be modified by the system: the PL7 workshop installation manual specifies the type of access (R or R/W) permitted for each of the system objects defined.

- Memory objects (variables and constants):

Object	Accepted syntax	TSX 37	PCX/TSX 57	TSX Series 7	TSX S1000
Internal bit	%Mi *%Bi* *%MXi*	R/W	R/W	R/W	R/W
Memorized internal bit (specific S1000)	%Rxi	-	-	-	R/W
Internal byte	%MBi	R	R	-	-
Internal word	%MWi *%Wi*	R/W	R/W	R/W	R/W
Internal double word	%MDi *%DWi*	R/W	R/W	R/W	R/W
Floating (32 bits)	%MFi *%FDi*	R/W	R/W	R/W	R/W
Constant word	%KWi *%CWi*	R	R	R	-
Constant double word	%KDi *%CDi*	R	R	R	-
Floating constant (32 bits)	%KFi *%CFi*	R	R	R	-
Common word on network 0	%NW{j}k j=station nb k=word nb	R/W	R/W	-	-

Object	Accepted syntax	TSX 37	PCX/TSX 57	TSX Series 7	TSX S1000
Common word on other networks	%NW{i.j}k i= network nb j=station nb k=word nb	R/W	R/W	-	-

Grafcet objects

Description

Object	Syntax	TSX 37	PCX/TSX 57
Step state	%Xi	R	R
Activity time of a step	%Xi.T	R	R
Status of step in a macro step	%Xj.i	-	R
Activity time of a step in a macro-step	%Xj.i.T	-	R
State of the IN step of a macro-step	%Xj.IN	-	R
Activity time of the IN step of a macro-step	%Xj.IN.T	-	R
Status of the OUT step of a macro-step	%Xj.OUT	-	R
Activity time of the OUT step of a macro-step	%Xj.OUT.T	-	R

Note: The macro-steps are only available on PCX/TSX 57 version 3.0 or above

Standard function blocks

Definition

See also PL7 (See Managing PL7 standard function blocks, p. 142) blocks for R/W property modification.
PL7_3 timer: %Ti

Object	Syntax	TSX 37	PCX/TSX 57
Current value	%Ti.V	R	R
Preset	%Ti.P	R/W	R/W
Done Output	%Ti.D	R	R
Running Output	%Ti.R	R	R

IEC 1131-3 timer: %Tmi

Object	Syntax	TSX 37	PCX/TSX 57
Current value	%TMi.V	R	R
Preset	%TMi.P	R/W	R/W
Working Output	%TMi.Q	R	R

Monostable: %Mni

Object	Syntax	TSX 37	PCX/TSX 57
Current value	%MNi.V	R	R
Preset	%MNi.P	R/W	R/W
Running Output	%MNi.R	R	R

Up/Down Counter : %Ci

Object	Syntax	TSX 37	PCX/TSX 57
Current value	%Ci.V	R	R
Preset	%Ci.P	R/W	R/W
Empty Output	%Ci.E	R	R
Done Output	%Ci.D	R	R
Full Output	%Ci.F	R	R

Register: %Ri

Object	Syntax	TSX 37	PCX/TSX 57
Input word	%Ri.I	R/W	R/W
Output word	%Ri.O	R	R
Full Output	%Ri.F	R	R

Object	Syntax	TSX 37	PCX/TSX 57
Empty Output	%Ri.E	R	R

Drum: %Dri

Object	Syntax	TSX 37	PCX/TSX 57
Full Output	%DRi.F	R	R
Number of current step	%DRi.S	R	R
Activity time	%DRi.V	R	R

Table objects

Definition

Reminders:

The size of the tables is unlimited, excepting (system and memory) bit tables, which are limited to 450 elements.

- Table of system objects:

Element type	Syntax	TSX 37	PCX/TSX 57
System bit	%Si:L	R	R
System word	%SWi:L	R/W	R/W
System Double word	%SDi:L	R/W	R/W

Note: Access to system objects via table syntax is an extension as far as PL7 language is concerned. The system objects defined in the TSX Micro and PCX/TSX Premium ranges are not all consecutive. This can limit access via table syntax in certain cases.

- Table of memory objects:

Element type	Accepted syntax	TSX 37	PCX/TSX 57	TSX Series 7	TSX S1000
Internal bit	%Mi:L *%Bi:L* *%Mxi:L*	R/W	R/W	R W if length modulo 8	R W if length modulo 8
Internal word	%MWi:L *%Wi:L*	R/W	R/W	R/W	R/W
Double word	%MDi:L *%DWi:L*	R/W	R/W	R/W	R/W
Floating (32 bits)	%MFi:L *%FDi:L*	R/W	R/W	R/W	R/W
Constant word	%KWi:L	R	R	R	-
Constant double word	%KDi:L	R	R	R	-
Floating constant (32 bits)	%KFi:L	R	R	R	-
Common word on network 0	%NW{j}k:L j = station no. k = word no.	R/W	R/W	-	-

Element type	**Accepted syntax**	**TSX 37**	**PCX/TSX 57**	**TSX Series 7**	**TSX S1000**
Common word on other networks	%NW{i.j}k:L i = network no. j = station no. k = word no.	R/W	R/W	-	-
Character string	%MBi:L *%CHi:L*	R/W*	R/W*	-	R/W**

(*) %MBi :L are R/W ONLY if the address and length are even. If not, they are read-only.
(**) ** The size must be between 2 and the maximum size permitted by ORPHEE.

Note:
- access to common words via table syntax is an extension as far as PL7 language is concerned,

- Table of Grafcet objects:

Element type	**Syntax**	**TSX 37**	**PCX/TSX 57**
Step status	%Xi:L	R	R
Activity time of a step	%Xi.T:L	R	R
Status of step in a macro step	%Xj.i:L	-	R
Activity time of a step in a macro-step	%Xj.i.T:L	-	R
Status of the IN step of a macro-step	%Xj.IN:L	-	R
Activity time of the IN step of a macro-step	%Xj.IN.T:L	-	R
Status of the OUT step of a macro-step	%Xj.OUT:L	-	R
Activity time of the OUT step of a macro-step	%Xj.OUT.T:L	-	R

Note: Beyond the "steps status", access to other Grafcet objects via table syntax is an extension as far as PL7 language is concerned.
Reminder:
Macro-steps are only available on TSX/PCX Premium version 3.0 or higher
Additional information on macro-step tables:

- %Xj.i:L syntax reads several consecutive steps (number L) of the macro-step (j).

Example:
%X1.0:3 corresponds to %X1.0, %X1.1 and %X1.2.

- The syntax of a particular step (IN or OUT) in a macro-step (j) reads this step for several consecutive macro-steps (number L).

Example:
%X1.IN:3 corresponds to %X1.IN, %X2.IN and %X3.IN.
%X1.OUT.T:3 corresponds to %X1.OUT.T, %X2.OUT.T and %X3.OUT.T.

5.4 Concept variables on OFS

Variables concept

Definition

State Ram Objects	Range	Access
Coils	0x	R/W
Input status	1x	R
Input reg as UINT	3x	R
Holding Reg. as UINT	4x	R/W
Holding Reg. as UDINT	4x	R/W
Holding reg. as REAL	4x	R/W

Symbols are supported throughout and all variables are represented by symbols, as there is no address syntax in the Concept language.
Relation between Concept basic data types and OPC data types:

Concept data type	OPC Data Type	Variant type	Returned format
BOOLEAN	BOOL	VT_BOOL	True/False
BYTE	BYTE	VT_UI1	8 bits
WORD	INT	VT_I2	16 bits
INT	INT	VT_I2	16 bits
UINT	UINT	VT_UI2	16 bits
DINT	DINT	VT_I4	32 bits
UDINT	UDINT	VT_UI4	32 bits
FLOAT	FLOAT	VT_R4	Floating IEEE
TIME	DINT	VT_I4	32 bits

Structures are supported. It is possible to access a structure either by a bytes table (the user needs to know the internal fields and their type) or field by field with the following syntax:
<**structure name**>.<**field name>**
In this case, the server finds out the data type directly from the Concept database.

Note:

- Structure access may only take place with a device associated with a Concept project file (*.prj) as a Symbol Table file. Both located and unlocated devices may be accessed if the above is true.
- For easy manipulation of a structure, the user may create a group, and then one item for each field of the structure within this group.
- Access to unlocated variables and structures is ONLY possible if the IEC runtime has been enabled in the PLC configuration. IEC Runtime availability on the PLC is only detected if the Device Verification option is active. If not, the unlocated support may be defined device by device, see Concept (See Concept Link, p. 143) link.
- In addition, unlocated variables and structures must actually be used in the PLC application to be readable/writeable with OFS. In fact, with Concept, any unused, unlocated variable is not recognized by the PLC. This is why OFS accepts the creation of an item linked to an unused, unlocated variable, but immediately defines its quality attribute as BAD to show it can no longer be read or written. It is possible to obtain automatic updates using the Concept programming tool and the DCC feature.
- A table item or an unlocated structure has "read only" access if the entire size of the table or of the structure exceeds 200 bytes.
- When an item represents an entire structure, it is considered a table.

5.5 Link between Modbus and IEC 1131 variables

Link between Modbus and IEC 1131 variables

At a Glance

It is possible to access certain Modbus variables using IEC 1131 syntax and vice-versa. That does not apply to located variables.
Access to Premium data via Modbus: the only variables that are accessible are those indicated below.
The link is as follows:
Modbus to IEC 1131:

0x	%Mi
1x	%Mi
3x	%Mwi
4x	%Mwi

IEC 1131 to Modbus:

%Mi	0x
%MWi	4x
%MFi	4x (access to 2 registers)
%MDi	4x (access to 2 registers)

Tables are also accepted.
Example:
The variable "Toto", located on register 400023, can also be accessed with %MW23 (UINT), %MF23 (Real) or %MD23 (UDINT). For %MF23 and %MD23, registers 23 and 24 are actually read. The syntax Toto:5 or %MW23:5 represents an array of five registers starting with Toto (=400023).
In fact, access to data on a Premium PLC via Modbus is strictly limited to the variables: %M, %MW, %MD and %MF.

5.6 Modsoft variables on OFS

Modsoft variables

Definition

The Modsoft supported syntax is limited to long addresses only (6 digits).
Example: 400001.
The following syntax ARE NOT supported (not to be confused with table syntax):

- 4:00001,
- 40001,
- 4x00001.

Any register located in the 6x range-id is not accessible.
The table syntax <reg. number>:<length> is supported for range-id 0,1,3,4.
It allows one or more registers to be read at once (actually <length> registers).
For Holding registers, it is possible to create a floating item or a Long integer using the postfix F or D. Two consecutive registers will be used. The usual R postfix can be used at the same time.
Example:
400001;F Floating point for registers 1 and 2
400012;D Long integer (32 bits) for registers 12 and 13
400120;FR floating point considered as read-only for registers 120 and 121

Modsoft syntax

Object	**Range**	**Item syntax**	**Access**	**Table**	**Max. size write**
Coils	0	00000i	R/W	00000i:L	800
Input status	1	10000i	R	10000i:L	-
Input register	3	30000i	R	30000i:L	-
Holding register	4	40000i	R/W	40000i:L	100

Reminder: In read mode, the size of the tables is unlimited, except for bit tables (system and memory), which are limited to 2000 elements.

5.7 Extracted Bits Support

Support of extracted bits

At a Glance

Generally speaking, the reading of extracted bits is supported for any variable of simple integer data type (including Concept unlocated variables):
The syntax is: <variable definition>: Xn or <variable definition>, n for XTEL.
The bits are numbered from 0 to 7 for 8 bits integers, from 0 to 15 (for 16 bits integers) and from 0 to 31 for 32 bits integers.

Element types, access:

Element type	**Accepted syntax**	**Quantum**	**TSX 37**	**PCX/TSX 57**	**TSX Series-40, 5000, 7000**
Extracted byte bit	%MBi:Xj	R	R	R	-
Word extracted bit	%MWi:Xj *%Wi,j*	R	R/W	R/W	R
Double word extracted bit	%MDi:Xj *%DWi:Xj*	R	R	R	R
System word extracted bit	%SWi:Xj	R	R	R	-
Constant extracted bit	%KWi:Xj	R	R	R	R
Symbol extracted bit (single or double word)	Symbol:Xj	R	R	R	R (series 7 only)
Structure field extracted bit	Struct.member: Xj	R	-	-	-

Examples:

PL7	**CONCEPT**	**XTEL**	**ORPHEE**	**MODSOFT**
%MB100:X6 %MW100:X3 %MD200:X25 %SW6:X7 %KW100:X0 pump :X4	pump:X5 struct1.member:X8 tab1[1000]:X4	W100,3 DW200,25 CW100,0 Pump,4	%MW100:X3 %MD200:X25	300500:X11 400100:X12

Writing extracted bits is only possible for %MW variables on PCX/PMX/TSX Premium and TSX Micro, version 3.0 or later (Device verification option must be activated) and is not supported for variables used in Push data.

5.8 Local variables on OFS

Local variables

Definition

There is a pseudo-protocol (driver name: "LOCAL") which allows for the creation of variables which are only local in relation to the server (unrelated to any hardware device). These local variables are always WORD (VT_12) type variables, created using a name.
Syntax: "LOCAL": ! <name>

Example: "LOCAL:!Bridge"

If two or more clients create the same local variable (the same name), its value is shared. This means that if a client modifies the value, the other client(s) will be notified (if notification has been activated). As long as the value remains unwritten (once it has been created), the quality is reported as bad. This function is generally used to exchange data from one client to the other.

OFS Server installation

6

At a Glance

Aim of this Chapter

The aim of this chapter is to introduce you to OFS server installation.

What's in this chapter?

This chapter contains the following Sections:

6.1 Introduction to server installation

Introduction to Server Installation

Introduction

Before using the OFS server, it must be installed (See Product Installation Procedure, p. 21) and configured (See Configuration Tool, p. 31). Once these two phases are complete, the OFS server is ready for use.

> **Note:** When using ready-to run supervisory software, you may not have the ability to use all the features that are listed in the following chapter (see the documentation of the OPC interface of your supervisory software to verify that point).

- Configuration:
  The Configuration Tool (See Configuration Tool, p. 31) allows the following operations to be carried out:
  - configuration of symbols tables,
  - configuration of aliases and addresses,
  - configuration of the device's options with its properties page,
  - configuration of general server options.
- Operation:
  The client must launch the server and initialize communication. The user can then:
  - create groups,
  - create items,
  - execute synchronous read,
  - execute synchronous write,
  - activate group notification,
  - activate group.

At the same time, the server automatically sends notification of value changes.

6.2 Using the OFS Server

Programming

Description

The main phases of application client programming (using either VB and Automation Interface 2.0 or C++ and Custom Interface) are as follows:

- creation of a CONNECTION with the OFS server (in local or offset mode):
  OPC-AUTOMATION: Connect()
  OPC-CUSTOM: CoCreateInstance() + QueryInterface() for IOPCServer + Get GroupCollection(),
- creation of one or more GROUPS:
  OPC-AUTOMATION: GroupCollection \ Add() + Get ItemsCollection()
  OPC-CUSTOM: IOPCServer \ AddGroup()+ QueryInterface() for IOPCItemMgt,
- creation of ITEMS in an existing group:
  OPC-AUTOMATION: ItemsCollection \ AddItem() or AddItems()
  OPC-CUSTOM: IOPCItemMgt \ AddItems(),
- READ or WRITE of group ITEMS:
  OPC-AUTOMATION: ptr \ ASyncRead() group or ptr \ ASyncWrite() group
  OPC-CUSTOM: IOPCASyncIO2 \ Read() or IOPCASyncIO2 \ Write(),
- destruction of existing GROUPS (this may include the destruction of all items contained within these groups):
  OPC-AUTOMATION: GroupCollection \ Remove() or RemoveAll()
  OPC-CUSTOM: IOPCServer \ RemoveGroup(),
- closing the CONNECTION with the OFS server:
  OPC-AUTOMATION: Disconnect()
  OPC-CUSTOM: IOPCServer \ Release().

Cyclic read of a group of items

Installing periodic read for a group of items requires the use of a notification mechanism, set up by carrying out the following additional operations:

1	Activation of a group and at least one of its items
2	Subscription to notification service
3	Regular receipt of notifications ("waken" function)
4	Cancellation of subscription to notification service
5	Deactivating groups and items

Note: The above information is only needed for the creation of new personalized applications.

6.3 Different Groups of Items

The Different Groups of Items

Description

The OFS product has 2 types of groups:

- **User Group:**
  - an item may be localized on any device,
  - it is not possible to find out the number of requests needed to read the whole group,
  - Any one part of a group may be read,
  - the group is notifiable,
  - the name of a group may be any string of characters.
- **Synchronous Group:**
  - all items must be localized on the same device,
  - it is possible to find out the number of requests needed to read the whole group (specific items #NbrRequest),
  - even if the user executes the read function on a part of the group, all items are read,
  - the group is notifiable,
  - the declaration of items in the Push Data box is prohibited in synchronous groups (it is impossible to guarantee that items in Push Data and items in polling will be updated synchronously)
  - the group name must start with $ or $$,
  - the device timeout for the devices in use in synchronous groups must be set to 0 (this function should NOT be used).

$: number of requests limited to 1. The creation of items is prohibited when the maximum size of a read request is reached. A write request is refused when the items in the group targeted for reading exceeds the maximum authorized size (it should be noted that a write request, because it contains both the description of the items and the values to be written, is more restrictive as far as number of items is concerned).

$$: any number of requests, all linked to the same device.

A synchronous group may include the specific item "#NbrRequest" which enables the user to find out the number of communication requests needed to read all of the items in the group.

This item is read-only and may be read at any time, without needing to physically read the group (no time used up on network communication).

This item can only be used in a synchronous group.

Note: The system group (See OFS server compatibility, p. 216) function is used for compatibility reasons only. Avoid this as much as possible (it is of no use for an ordinary group).

6.4 Managing Tables

Management of tables of variables

Description

- the OFS server manages tables of variables. This provides easy access to a group of contiguous variables of the same type,
- the OFS server accepts several kinds of syntax, according to the target PLC: <**Origin Element**>:<**Length**>
  The <Origin Element> field represents either the address or the symbol of the first element in the table. The <Length> field represents the number of elements (of the same type as the origin variable) in the table.

Example: either of a %MW10 address variable and of a MYARRAY symbol.
A table of 20 elements starting with this variable may be referenced in the following two (equivalent) ways:

- %MW10:20
- MYARRAY:20

Note: This is the only syntax to allow the referencing of a table as a symbol for **PL7** objects, as the tables cannot be symbolized in PL7 language. The **Concept** language accepts symbolic references to a table. This syntax can always be used with **Concept** and **Modsoft** variables.

- There are no limits to the size of the tables. However, they must not exceed the zones configured through the workshop,
- a table of variables corresponds to a single item of a group.

Reminder:
This means that during the "cyclic" read of a group containing a table item, the OFS server sends the whole table to the client application, regardless of the number of elements in this table whose values have been changed.

6.5 Read Consistency

Read Consistency

Definition

- Consistency of a group of items:
  The OFS server guarantees that all the items in a group are consistent with each other (i.e. read in the same PLC cycle) if and only if the group is transcribed on a single request. This means that the client application can be sure of the consistency of the items accessed in read mode when the #**NbrRequest** specific item associated with the group or the device equals 1 (synchronous group only)
  For more details refer to the section on performance (See Number of Lodgeable Items in a Single Request, p. 200).
  When the **prefix** '$' is used before the name of a group, the OFS server checks each time an item is added that the request number does not exceed the unit. These are known as **single request** user groups.
  During a write request, if the number of items from a synchronous group exceeds the size of a request, it will be refused in its entirety.

Note: The maximum quantity of lodgeable items in a write frame is generally less than the lodgeable quantity in a read frame. That is the reason why writing all of the items in a synchronous group can fail

The OFS server (AddItems primitive) refuses to add the item and reports an error if a single request group cannot be transcribed on a single request.

6.6 Write Consistency

Write Consistency

Definition

The write primitive displayed by the OFS server allows one or several items to be written in a group at the same time. Obviously the items must be modifiable.

Note: During a write request, the OFS server overwrites the old values present in the PLC. The client application must take charge of the preliminary overwrite confirmation, if this is necessary.

If, during a write request concerning several items, there is some overlapping between items, it is the value associated with the last of these items which prevails.

Example: If the write concerns items "%MW0:5" and "%MW0", the value given for the 1st element of item "%MW0:5" will be written but will be overwritten as soon as the write request of the 2nd item ("%MW0") is taken into account.

Consistency of the variables with each other during a write operation:
Write consistency is guaranteed when the data to write is lodged in the same network request, i.e. either table type variables or same type variables, whose addresses are contiguous and whose total size does not exceed the maximum (See Number of Lodgeable Items in a Single Request, p. 200) size for a request.

6.7 Asynchronous Functioning of the Server

Asynchronous Operation

Description

In asynchronous operation mode, a request for any asynchronous operation receives an immediate response. That does not mean that the operation requested has been completed, but that it has either been refused (incorrect code response), or that it is underway (correct code response).

The completion and the outcome of the operation will be announced using the notification mechanism. In order for this to occur, this mechanism must be activated before using asynchronous operation.

The following are the four operations:

- Read,
- Write,
- Refresh,
- Cancel.

Read/Write:

Similar to asynchronous operation with the same name (same functions, same restrictions).

Refresh:

Requests notification of all the values in progress of all the group's active items. The group must be active.

Cancel:

Stops the progress of a current read, write or refresh operation. It is not possible to know if the operation has actually been stopped.

6.8 Installation of the Periodic Read Utility

Periodic Read Utility Installation

Description

Installation of the periodic read utility of a group's items consists of 4 stages:

- Subscription of the group to the notification service set up by the OFS server.
- Programming of the OnDataChange "waken" function, called by the OFS server to notify of changes in values that have occurred in the groups.
- Activation of all the items to be examined, if this has not already been done.
- Activation of the group to trigger regular examination of the group's items, which the OFS server is responsible for: ActiveStatus property set to TRUE. In terms of performance, it is advisable to activate the elements within a non-active group first and THEN activate the group. By doing this, you will avoid having too long a start up time due to numerous network requests.

Note: Synchronous read and write functions are possible during the periodic read. They are not recommended however, as they may bring about anomalies (notification is not sent, as the value was read before the required sending of notification.)

Reminder:

The OnDataChange primitive receives notifications for **all groups** whose servers ensure read polling.

- **Notification** is done by group, and not individually for each item of a group. So the OnDataChange primitive receives the **list of the group's items** having changed value from one of the read polling function's iterations to another.

Stopping a group's periodic read utility is carried out in 2 phases:

- deactivation of the group: ActiveStatus property set to FALSE,
- stopping the group's subscription to notification service.

Note: For user groups: it is possible to activate/deactivate the group's item at any time. For synchronous groups: (name starting with $ or $$) all items are still regarded as active, in other words no partial activation/deactivation is possible.

6.9 Installation of the dead band

At a Glance

Aim of this Section

The aim of this section is to introduce you to dead band installation.

What's in this section?

This section contains the following topics:

Definition of dead band

Definition

Dead-banding is associated with the cyclic reading of a user group, and is a method of filtering notifications of changes in the value of items: it avoids waking up the client application when the variable changes within a dead range around the last value received.

Note:

The dead-banding mechanism does not reduce the flow of requests between the server and the PLC. It is used to reduce the number of notifications sent by the server, and thus processed by the client application: this saves CPU time.

Note: The dead banding mechanism has no effect when the client requests a synchronous or asynchronous read or refresh.
The dead-banding feature is not available for items that are actually arrays.

Description of the dead band mechanism

Description

The OFS server uses dead-banding as specified by OPC standards:
In general, dead-banding only concerns real variables: "%MF", called "ANALOG" variables by the OPC standard. As an extension of this standard, it is possible to use this feature for integer variables provided that the configuration steps described below have been followed.

Note:
The OFS server uses this OPC term to refer to "floating point" type PLC variables, even though this term does not correspond to the idea of analog variables usually used in the control system field.

Dead-banding is based on the following notions:

- The analog type, defined with min. and max. limits which represent the range of values (interval) of the variables being handled. This notion has been introduced because the OFS server has no way to obtain these maximum and minimum values directly from the programming tool (PL7, Concept, XTEL or ORPHEE).

Example:
AnalogType = [-1.0, 1.0]
The max limit of an analog type (1.0 in the above example) is called Engineering Unit high bound (Eng. high bound unit). The min limit (-1.0) is called EU low bound (Eng. Low bound unit)

- notion of a usual notification range, which corresponds to the difference between the max limit and the min limit defined for an analog type.
  In the previous example:
  The usual range of variation of the "AnalogType" is: 2 = (1.0 - (-1.0)),

- notion of notification threshold, which conditions the transmission of a notification to the client application: notification is transmitted if, and only if, the difference (in absolute value) between the value read and the last value sent is above this threshold.
  The threshold value of an analog type is calculated by applying the dead-banding value defined for the group to the usual variation range of this type.
  Dead banding is a percentage variation between 0 (i.e. 0%) and 1.0 (i.e. 100%).
  To summarize, for an analog type, the notification condition is as follows:
  ABS (Value read - Last value sent) > Dead-banding * (Max limit - Min limit).

Note:
All notifications are sent if the dead-banding is 0% (default value). In the previous example:

If the dead-banding value assigned to the group is 10%, the notification threshold for the "AnalogType" is:
0.2 = 0.10 (dead-banding) * 2 (usual variation range).
This means that only those variables in the group whose value varies by a difference of more than 0.2 (in absolute value) will be notified to the client application.

Installation of the dead band in a client application

Description

- declaration of analog types: use of the Configuration tool (See Configuration Tool, p. 31).
  Note:
  "AnalogType" is the name given to the analog type by the user.

> **Note:**
> 1. A maximum of 100 analog types can be defined.
> 2. The limits for an analog type cannot be modified by the user once the server has started. This is due to the fact that when it is started, the OFS server learns the analog types defined in the registry. The OFS server must be stopped and then restarted for an alteration of the limits of an analog type to be taken into account.

- Defining the value of the dead band:
  The dead-banding Percentage associated with a user group can be set when the group is created (AddGroup primitive) or dynamically adjusted during the server session (PercentDeadBand property).
- attaching an item to an analog type:
  The general syntax of an item (See General information on OPC items, p. 86) has an optional parameter which states the analog type to which it belongs, and so informs the OFS server of its notification threshold.
  The syntax of an item with analog type is as follows:
  <item> ::= <driver name>:<API address>!<variable name>[@<analog type name>]
  Note:
  The space before the @ character is optional.
  Example of an item definition: "FIP01:0.31.0!%MF330 @AnalogType".

Comments:

- it is possible to have the same item twice in the same group (e.g. "%MF330") with and without the analog type suffix (" @AnalogType"), so as to be able to compare the effect of dead-banding for notification filtering.
- it is possible to have items of different analog types in the same group (i.e. several analog types referenced in the same group).

6.10 Adjusting the Device Time-Out values

Adjusting timeout item values

Description

The **frame time out**:
It shows the maximum length of time the Server will wait for an answer from a device after sending a request. This time-out can be defined device per device within the device property page. If the value in the device property page is set to 0 (see alias (See Setting the alias properties, p. 44) properties), the default value is 1000 ms (device value).
The frame time-out may be configured dynamically, device by device, using the specific (See Specific Items, p. 150) #TimeOut item.

The **device time out**:
To avoid item quality flickering and over-long OPC application startup time due to missing devices, a device time-out feature has been introduced.
When active, this option has two effects:

- if the device detection process fails (check device option is active and device is physically missing), then the device will be considered 'missing' for a time equal to the Device Time-out. During this time, the creation of any items for this device is immediately rejected.
- during normal device operation, two kinds of requests are sent to the device:
  - polling requests (for active items within active groups)
  - one-shot requests (synchronous or asynchronous read or write)

In case of a failure, one-shot requests report failure immediately (usually though items quality).
Malfunctions linked to polling requests are not shown immediately. They are only shown when the malfunction persists during the entire device time out. In this case, the malfunction is shown for all active items in the device (all items quality set to BAD, as device considered in BAD state). Any successful polling request resets all parameters to GOOD (items quality and device state).
This time out can be defined device by device in the properties page.

If the value in the device property page is set to 0, then the feature is disabled.
Its default value is 5000 ms.
This feature is **incompatible** with **synchronous groups**.

On **Modbus+**, you can set two links (DM or PM), which are considered as modes for the different devices. Therefore, the values of one device do not influence the values of the other.

For **Modbus** devices, if the multi-channel (See MODBUS Multi-Channel Feature, p. 154) feature is enabled, the frame time-out value is the same for all channels open with a given device.

6.11 Managing the PLC Operating Mode

PLC operating mode management

Description

The PLC operating mode can be controlled using the specific item #PLCStatus. The modification of any PLC operating mode by the server can be enabled/disabled using theConfiguration tool (See Options tab, p. 56).
The current operating mode of the PLC is viewed by reading the #PLCStatus specific item. Since this item can be activated, it is possible to monitor the PLC operating mode using it.
The current operating mode of the PLC is modified by writing the #PLCStatus specific item.
The following values are associated with the different operating modes of the PLC:
STOP: 0* RUN: 1* INIT: 2**
(*) Non operational for S1000 type PLCs,
(**) Non operational for Quantum, S7 and S1000 type PLCs.

Note: If the programming tool is connected to the device, modification of the operating mode of the PLC may fail due to the exclusive reservation performed, for example, by PL7 or Concept.
Modbus+ devices have Data Master (DM) or Program Master (PM) modes. To modify the PLC operating mode of a device, the path of its alias must be defined with the PM option.

6.12 Communication

At a Glance

Aim of this Section

The aim of this section is to describe the type of communication used by the OFS server.

What's in this section?

This section contains the following topics:

Introduction

General

- the OFS server allows the use of several different communication media simultaneously: a client application can, for example, gain access to a PLC using FIPWAY and to another using ISAWAY.
- the OFS server provides XWAY and MODBUS network transparency: A client application can access PLCs in a PLC network architecture including bridges for switching between communication media.

The OFS server allocates for:

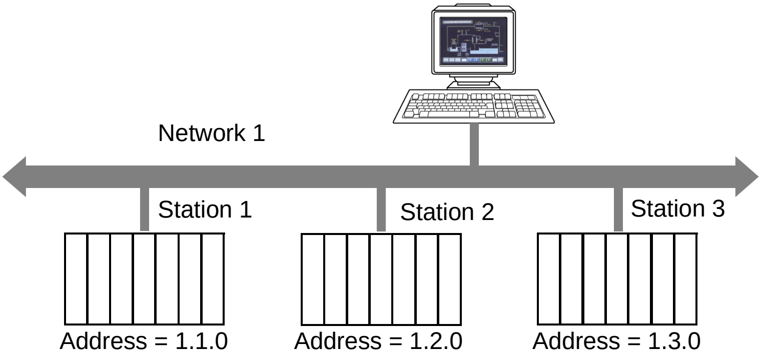
- XWAY: a socket (communication channel) for each medium
- MODBUS PLUS: a path for each device (PM) or up to 4 paths for each device (DM), see Modbus (See MODBUS Multi-Channel Feature, p. 154) multi-channel section.
- MODBUS/TCP-IP: up to 4 sockets for each device, see Modbus (See MODBUS Multi-Channel Feature, p. 154) multi-channel section.

Note: MODBUS PLUS paths are opened and closed dynamically according to requirements. Therefore, even with only one SA85 card (8 DM paths), it is possible to dialog with more than 8 devices.

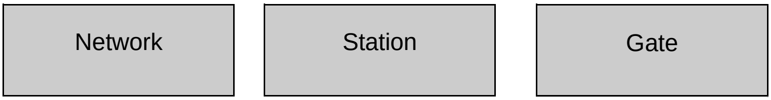
XWAY addressing modes

Description

Example of access through a network:



Addressing to 3 levels:
Allows a station connected to the network at any point of the XWAY communication architecture to be reached.
Illustration:



The Network and Station values make up the station address.

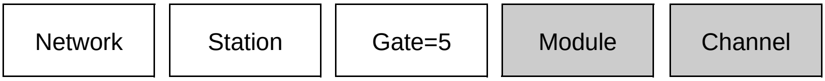
- Network: value between [1.127] or 0 = my network.
- Station: value between [1.63] or 254 = my station or 255 = diffusion.

The value "Gate" refers to the communication entity within the station: system server (Gate 0, the most recent), the terminal port (Gates 1,2,3), 1K asynchronous server (Gate 7), etc.
In the case of multiprocessor stations such as PLCs, each processor module built into the system can support communication entities, frame routing requiring supplementary addressing levels (inter-station routing capabilities). PLC "processor modules" are situated in the PLC's racks or offset on field buses.

Addressing to 5 levels:

It is generally used for devices connected on a UNI-TELWAY bus.

Illustration:

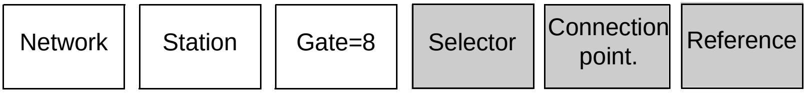


- Module: physical location of the communication module in the rack. Its value must be defined as follows: (Master rack number * 16) + Number of master module.
- Channel: address of the device connected to the communication module. Its value must be defined as follows: (Master channel number * 100) + slave Ad0 number.

Addressing to 6 levels:

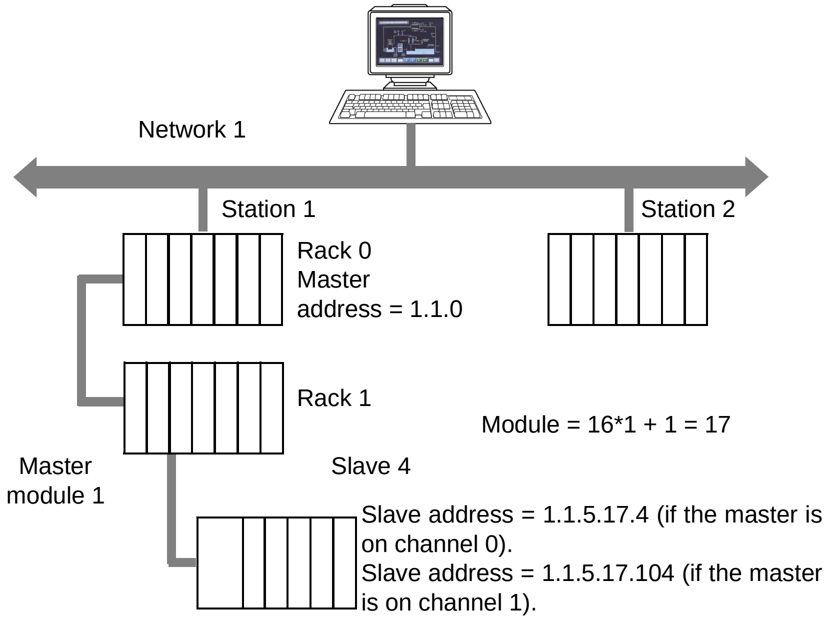
This is similar to addressing to 5 levels. It was created for extended services (FIPIO, communication module integrated in rack).

Illustration:

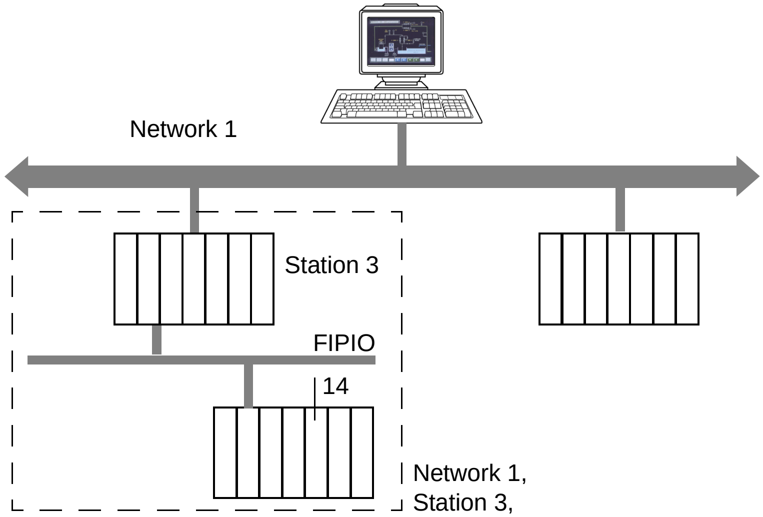


- **Selector**: designates a communication module on the CPU (2) or in a separate module (1).
- **Connection point**: device address, if the destination module is FIPIO. Physical positioning in the PLC rack, if the destination module is a PLC card.
- **Reference**: communication entity in the device (similar to the Gate number).

Examples:
5 level addressing:


Network 1
Station 1
Station 2
Rack 0
Master
address = 1.1.0
Rack 1
Module = 16*1 + 1 = 17
Master
module 1
Slave 4
Slave address = 1.1.5.17.4 (if the master is on channel 0).
Slave address = 1.1.5.17.104 (if the master is on channel 1).

6 level addressing:



Network 1,
Station 3,
Gate 8 (FIPIO),
Module 2 (CPU communication module,
Address 14,
Gate 0 (UNITE target PLC server),
→ 1.3.8.2.14.0

For more information on the XWAY address, see "XWAY Communication" documentation, ref. TSX DR NET.

Note:
In point to point connections (UNI-TELWAY, ISAWAY), the default address 0.254.0 can be used to reference the PLC.
With ETHWAY and XIP, it is possible to use gate 7, which accepts large frames (up to 1024 bytes). In order to do this, the PL7 application must be configured in periodic mode (MAST task). The "1K service" option must be checked in the alias definition page.
Example: normal address: XIP01:1.2, to use gate 7: XIP01:1.2.7

MODBUS addressing modes

Description

- over TCP/IP, the only information required is the IP address. This can be given as four groups of numbers separated by dots or by a DNS name e.g. "My Station". In this case, the DNS scanning feature should be enabled (See Options tab, p. 56).
- over Modbus+, the syntax is:
  <access level>.<node1>.<node2>.<node3>. <node4>.<node5>

The access level can be:

- PM = Program Master,
- DM = Data Master.

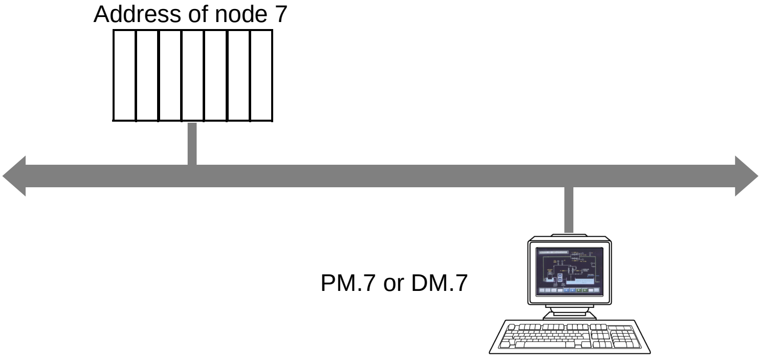
The node number should be used to specify the full path. To access a device without a bridge, only access mode and node are required.

For TCP/IP - MODBUS+ bridges, the syntax is:

MBT:<IP bridge address>;<Modbus+ device node number>

The IP bridge address corresponds to the number entered into the configuration tool's "MBP bridge index" box. This configuration is detailed in the network section of the device (See Editing the Device Network Address, p. 39).

Example:



- JBUS devices connected to MODBUS networks:
  - use the option /J after the device address (e.g. MBS01:6/J, see modification of the device network address (See Editing the Device Network Address, p. 39). This option tells the server that items should start with index 0 (%MW0 should be read as register 0 and NOT as usual with Modbus where %MW1 is read as register 0) and that double words and floats should be processed with JBUS word ordering and not with MODBUS word ordering,
  - any memory area of a JBUS device may be accessed as long as the absolute address is given in decimal form (6 digits (See Variables concept, p. 98)) and not in hexadecimal form. E.g.: read the bit %RX0 on a JBUS device: offset is $E000 so the item 057344 should be created.

  - It is possible to create a CSV file with one line for each necessary address / symbol link to avoid typing and to link it with the JBUS device (See Setting the alias properties, p. 44). Example (See CSV symbol table file, p. 138): 057347 RX_3.
- TSX devices connected to MODBUS networks:
  - use the option /T after the device address (Example (See Editing the Device Network Address, p. 39): MBT:1.2.3.4/T). This option tells the server that items should start with index 0 (%MW0 should be read as register 0 and NOT as usual with Modbus where %MW1 is read as register 0).
  - It is possible to link and use any PL7 symbol table file with a TSX device connected to a MODBUS network (PL7-3 symbols files are not managed).
- OFS server behavior in the event of a communication problem with the PLC (PLC missing, disconnected, etc.) :
  - whatever the type of communication problem, all the requests corresponding to one group will be transmitted, both for reading items and for writing items.
  - from a performance point of view, this means that the execution period for the read or write primitive may rise to n times the time-out period (where n is the number of requests associated with the group).

  Note:
  There are no request retries on time out.

Note: For networks with logical connections, if the connection is broken, the server automatically tries to re-establish it.
E.g: Modbus TCP-IP.
When the XIP driver is used more than one device connected and one of these is absent, communication with devices connected by XIP is blocked for a few seconds as XIP uses Winsock and awaits the end of the TCPIP timeout. After this timeout, everything should come back normal except, of course, communication with the missing PLC.

The OFS server will indicate communication errors to the client application in the following way: each item belonging to a failed request will be marked "invalid" *, whether it is for a group's synchronous or cyclic read request.
* Whatever the method used to perform the read, "invalid" means that the Quality attribute is Bad. "Valid" means that the Quality attribute is Good.
Notes:

- the client application can determine whether the PLC has been reconnected by re-addressing a synchronous read request to the group concerned,
- during the cyclic read of a group, the quality of items (Quality attribute) will change from Bad to Good when the PLC is reconnected. The section on managing errors (See Error management, p. 145) describes the Quality attribute associated with an item.

6.13 Symbol management

At a Glance

Aim of this Section

The aim of this section is to describe symbol management.

What's in this section?

This section contains the following topics:

Introduction to symbol management

Introduction

OFS server establishes symbol/address correspondence using a symbol file. This symbol file may have been created by a programming workshop (Concept, Modsoft, PL7) or with an external tool such as a Text Editor (CSV format).

For devices in the S7 range, access to symbols is only possible by firstly converting corresponding applications to Premium applications.

Supported symbol file formats are the following:
- PL7 exported symbol table (default extension SCY),
- PL7 exported application file (default extension FEF),
- Concept exported symbol table file (default extension CCN),
- Concept project file (default extension PRJ),
- Modsoft exported symbol table file (default extension TXT),
- CSV exported symbol table file (default extension CSV),
- Taylor exported symbol table file (default extension FIS).

For each format, only symbols that are associated with enough information for accessing variables are loaded and can be used (see below for details).
Symbol/address correspondence can also use a Concept (See Concept Link, p. 143) project file.
Several devices or groups can share the same symbol table file.

The link between the symbol file and a group of items is established either:
- by creating a link between a device and a symbol table. The Configuration tool is used for this:
  - creating an extension for the format you intend to use (e.g. .txt for Modsoft format),
  - creating an alias for the device with the Configuration tool,
  - linking the symbol table and this device.
- when the group is created, by entering the name and path of the symbol table. E.g.: creating group 1= C:\test.csv

> **Note:** In all cases, the extension should have been configured (See Symbols Tab, p. 52) beforehand.

Symbol management is intended for a user group. Syntax of a group name: <group name>[=<symbol table file path>].
OFS server will report an error to the client application if, while establishing this link, it detects that the neutral file does not exist or that it is invalid (its content is syntactically incorrect).

If a symbol file contains "collisions" (multiple declarations of the same symbol or address) the OFS server only retains the 1st occurrence of this identifier, and does not take following occurrences into account:
for example, if a symbol file contains the following associations:

- "PUMP" associated with "%MW0",
- "PUMP" associated with "%MW1",
  so the OFS server considers that the "PUMP" symbol corresponds only to %MWO. This is an example of an incorrect file.

Note: The use of symbols has no effect on the performance of the read write utilities of variables displayed by the OFS server. The only difference in performance concerns the group creation phase: The creation of a group of symbols is in fact longer, as it includes the translation of the symbols into addresses when creating the items in the group (AddItems primitive).

PL7 exported symbol table file

Description

To create such a file using the PL7 workshop:

- open the application using PL7,
- open the application browser,
- open the data editor,
- open any window in this editor (e.g. memory objects),
- use the File->Export menu to create the file.

This exported file authorizes the consistency check (name and application version) between the symbol table file and the application in the PLC (see Setting the alias properties, p. 44 and Communication Tab, p. 60.

PL7 exported applications file

Description

To create such a file using the PL7 workshop:

- open the application using PL7,
- use the File->Export Application menu to create the file.

exported file authorizes the consistency check (name and application version) between the symbols table file and the application running in the PLC (see Setting the alias properties, p. 44 and Communication Tab, p. 60). The OFS server does not use this file's configuration data.

CONCEPT exported symbol table file

Description

To create such a file using the Concept workshop:

- open the application with Concept,
- use the File->Export menu (Export-> File),
- select Variables : delimited text (Variables: or Program : IEC text ,
- do not choose a section,
- create the file with the appropriate (See Associating a Symbols Table File, p. 43) extension.

The two other choices in the File-Export menu (Factory Link: and Modlink variables) should be avoided.

The exported file authorizes the consistency check (application name and version) between the symbol table file and the application in the PLC (see Setting the alias properties, p. 44 and Communication Tab, p. 60).

Only access to located variables is possible using this kind of file since it does not contain the necessary information to access unlocated variables. For the same reason, access to structured variables is not possible. All these types of symbols are ignored.

MODSOFT exported symbol table file

Description

To create such a file with the Modsoft workshop:

- open the application with Modsoft,
- from the main menu, select "Utility"->"Symbol Table" > to open the symbol table editor,
- use the "File I/O"->"Export" menu to create the file.>

This exported file does not authorize the consistency check (application name and version) between the symbols table file and the application in the PLC (See Standard parameters and Alias edition, p. 34).

Modsoft applications may receive comments, using the comments section of the file. However, the OFS server only uses reference symbols.

Only symbols compliant with IEC format are supported. Symbols defined for the bits extracted from registers ARE NOT supported; all symbols of that kind are ignored as well as all symbols which are not IEC compliant.

CSV symbol table file

Description

This type of file can be used with tools such as text editors (e.g. Notepad) or other tools (e.g. Excel 97 or later).
The format of each line is very simple:
<Address><Separator><Symbol><Separator><Comments>

- the <Address> should be a valid address for the device associated with that symbol file,
- the <Separator> can be a comma, a space or a tab character,
- the <Symbol> can be any string of characters without a comma/space/tab/special character.

In cases where certain special features are used (table length, special postfix such as R), add them to the address.
Example: table with 10 Read Only registers,
400001:10;R Table_Status
This file does not authorize the consistency check (application name and version) between the symbols table and the application in the PLC (see Setting the alias properties, p. 44 and Communication Tab, p. 60). With Excel 97, use commas as separators.

Note: Maximum lengths are 50 characters for the address, 33 characters for symbols and 510 characters for comments.

TAYLOR exported symbol table file

Description

To create the symbols file using the Taylor workshop:

- open the application with the Taylor ProWORX Nxt tool,
- in the main menu, select "File"->"Utilities" -> "Import/Export">>,
- select "Export",
- select the file type "Symbol .FIS file",
- click on the Start button.

This exported file does not authorize the consistency check (application name and version) between the symbols table file and the application in the PLC (see Setting the alias properties, p. 44 and Communication Tab, p. 60).

6.14 Symbols browse support

Browsing symbols

At a Glance

Symbols browsing is supported by the OPC-Browse interface. It has a multi-level hierarchy:
A node for each device declared in the registry (Alias, Path, Symbols table) whether it is actually connected or not.

For each node:

- a sub-directory named "#Specific" for all specific items that can be created for this device,
- a sub-directory for each structured variable or array (Concept project file only) which, in turn has a sub-directory if the structure contains arrays or sub-structures.
- the complete list of application symbols declared in the symbols table file (See Associating a Symbols Table File, p. 43) associated with the device (See Associating a Symbols Table File, p. 43) or nothing (no symbol) if no symbols table has been declared for the device.

Devices which are connected but which have not been configured in the alias table can not be browsed.
Filtering possibilities exist to allow the user to select by type (for example, to ask for all Boolean variables), by name (wildcard '*' accepted), by rights of access, by located or unlocated character (Concept project file only), by structured or non structured character.
The associated address and comments can also be obtained with each symbol ("&A" filter for the address and "&C" for the comments, or both "&A&C").
It is also possible to filter the variables using criteria based on their addresses.

Summary of the filter syntax (BNF syntax):
<Symbol filter>[=<Address filter>][&A][&C][&E][&S].
<Symbol filter> any symbol string, including the wildcard '*',
<Address filter> any address string, including the wildcard '*',
&A: requests the display of the address,
&C: requests the display of the comments,
&E: only displays simple elements and not structures or arrays (for Concept project only),
&S: only displays structures and arrays (for Concept project only).
Examples of filters:

T*	Requests all symbols beginning with T
B* &C	Requests all symbols beginning with B as well as any possible associated comments

* =%UL	Concept project file : requests only unlocated variables
* =%MW1*	Requests all variables with addresses beginning with %MW1
T* =%MX* &A&C	Requests all symbols beginning with T, with addresses beginning with %MX and requests the display of the address and comments

So that the browse interface can go faster (certain types of software require that all the symbols tables are opened when the browse interface is opened), a symbols table can be preloaded when the server is started. This option is selected with the configuration tool when an alias is created in the properties page.

6.15 Managing Standard PL7 Function Blocks

Managing PL7 standard function blocks

Reminder

It is possible to modify the R/W fields of a Standard Function Block (e.g. "Preset" field of a %MNi.P Monostable), only if the Function Block has the "adjustable" property. "Adjustable" or "non adjustable" properties are assigned in the Configuration editor of the PL7 workshop.

During a write request from the R/W field of a standard Function Block, the OFS server does not carry out preliminary checking to make sure that the object has the "adjustable" property.

This means that if the Function Block does not have this property, the OFS server returns the generic error code corresponding to write request failure.

6.16 Concept link

Concept Link

Description

Installation of the Concept link is only possible with Concept 2.1 or above.
To install the Concept link, you must simply select the prj project file (see Associating a Symbols Table File, p. 43 and see Symbol management, p. 131) as symbols file for any device or group.
This prj file authorizes the consistency check (application name and version) between the symbol table file and the application in the PLC (See Dynamic Consistency, p. 155).

The Concept workshop and the prj files should always be located on the same machine. The OFS server can be located either on the Concept machine (usual case) or on another machine (Remote Concept Link feature).
The same project can be used simultaneously with the Concept and OFS Workshop in Windows 2000 & NT 4.0 as long as Concept is running in its own memory space (16 bit program).

In order to do this:

- edit the usual Concept shortcut properties,
- in the Shortcut tab, check the "Run in Separate Memory Space" box.

With OFS, more than one Concept project can be used at once, as long as they are from the same version of Concept. In order to do this, simply create the aliases required, and for each of these select a different project file.
OFS software, when used with the "stripped" Quantum executable file will not read non-located variables.

If you envisage using non-located variables:

- the Quantum exec MUST NOT be a "stripped" version,
- IEC runtime must be activated on the PLC,
- the "unlocated support" option MUST be checked in the properties page. Otherwise, no access to any unlocated variable will be performed.
- the "device verification" option must be checked if you are using a version of Concept project prior to 2.5.

> **Note:** If non-located variables are accessed without consistency between PC/PLC applications, they will be rejected. So there is no risk of obtaining erroneous data.

For experienced users, there is supplementary information in the ConceptLink\readme.txt which is located on the product's CD.

6.17 Remote Concept link

Remote Concept Link

Description

The remote link offers exactly the same features as the normal Concept link. The only difference is that the Concept machine (where the Concept programming tool and the Concept project files are situated) is not the one on which the OFS server or the simulator is launched.

These machines must be linked by DCOM (usually on TCP/IP). An OFS server (with a license) or an OFS simulator (DEMO mode) must be installed on the Concept machine. An appropriate DCOM configuration must be carried out in order to enable access to this server which is called "proxy server".

On the OFS machine, when specifying a Concept project, open the device properties page in order to check the appropriate remote Concept option (the proxy server is either an OFS server or an OFS simulator) and give the complete access path of the Concept machine.

The path of the Concept project must be the same as that seen by the proxy server on the Concept machine (it must begin with the letter of a drive, followed by the complete path).

6.18 Error management

At a Glance

Aim of this Section

The aim of this section is to introduce you to error management.

What's in this section?

This section contains the following topics:

Error Feedback Mechanism

Description

- **the error feedback mechanism consists of three parts:**
  - description of the result of the call (execution) of a primitive,
  - the description of the validity of an item: Quality flag,
  - the availability of a GetErrorString primitive used to call up the description label of an error from its code (See Exhaustive list of error codes, p. 217).

- **description of the result of calling up a primitive:**
  - All the methods offered return an error code. The programming language used to carry out the OPC client can use it as an error code or trigger an exception (usually languages using OLE Automation, e.g. Visual Basic).

In particular, this means that an error detected by a "function" type primitive is not indicated to the caller by means of the value that it returns.

- **errors that can be returned are the following:**
  - E_xxx errors: standard errors defined by OLE and Win 32,
  - OPC_E_xxx errors: errors specific to OPC,
  - EOL_E_xxx errors: errors specific to the EOLES server,
  - OFS_E_xxx errors: errors specific to the OFS server,
  - in addition to the action described above, some of the exposed primitives contain a pErrors parameter in their call interface (output parameter).

This pErrors parameter is defined for the primitives that can manage several items during the same call (e.g.: AddItems).

- **pErrors allows you:**
  - to log a report for each item (an element in the pErrors table),
  - to indicate an error to the caller through a channel other than that for triggering the exceptions. Typically, when S_FALSE is returned, there is no triggering of exceptions, as the result from the primitive is of type success with a warning. In order to know on which item the error occurred, the pErrors parameter must be consulted.

For example, the pErrors parameter allows notification for the AddItems primitive that some of the items mentioned have an invalid syntax.

- **description of the validity of an item:**
  - primitives with "synchronous" and "cyclic" read contain the parameter pQualities, which describes the validity of the items concerned. They give a Quality attribute for each item.

For these primitives, this parameter comes in addition to the pErrors parameter. The Quality attribute of an item is a value on 8 bits composed of 3 fields : Quality, Substatus and Limit.

B7	B6	B5	B4	B3	B2	B1	B0
Quality		Substatus				Limit	

To obtain the error code which corresponds to the field concerned, it is advisable to apply the appropriate extraction mask and to study the value thus obtained.

- the Limit field (2 bits) is not managed,
- the Quality field (2 bits) which designates the validity of an item's value:

B7	B6	Quality	Meaning
0	0	Bad	The value of an item is incorrect for the reasons shown in the Substatus field
1	1	Good	The value of an item is correct
0	1	Uncertain	An error has been detected on the item, but it is still "too early" to set it to Bad. Transitional status.

- the Substatus field (4 bits) which provides details on the Quality field, and whose significance varies according to the value (Bad, Good) of the Quality field.

The **Substatus** field for the **Bad** value of the Quality field:

B5	B4	B3	B2	Substatus	Meaning	Validity value
0	0	0	0	Non-specific	Incorrect value with no specific reason: various causes	0
0	1	1	0	Communication error	Incorrect value due to a communication problem with the PLC	24

The **Substatus** field for the **Good** value of the Quality field:

B5	B4	B3	B2	Substatus	Meaning	Validity value
0	0	0	0	Non-specific	Correct value. No particular conditions	192

The **Substatus** field for the **Uncertain** value of the Quality field:

B5	B4	B3	B2	Substatus	Meaning	Validity value
0	0	0	0	Non-specific	A risk of an error has been detected.	64

Note: For all the other values not mentioned in the above tables, please contact technical support.

Objects outside Software Configuration

Description

The OFS server does not have access to the software configuration of the applications it accesses.

If a group contains items that are outside the software configuration of the application in the PLC, the reading of the group may fail on other items that are compatible with the configuration, due to the fact that optimization algorithms are used in read requests.

Where the erroneous item is a table, the OFS server sets the whole table to error, even if only one sub-element of its elements is outside configuration.

Example 1: Application in which 522 words have been configured: from %MW0 to %MW521. The read or write of a group containing table item %MW520:10 will send back an error for the whole of this item, even though the words %MW520 and %MW521 are in the configuration.

Note: The words %MW520 and %MW521 in this example can be accessed individually.

Example 2: Application in which 522 words have been configured: from %MW0 to %MW521.

An active group with the active items %MW0 (quality GOOD) and %MW500 (quality GOOD).

If the item %MW530 is added, %MW500 becomes BAD and %MW530 is BAD but %MW0 remains GOOD.

Explanation: reading the whole active group requires 2 requests: one for %MW0 and another for %MW500 and %MW530.

The first request is still OK: %MW0 remains GOOD.

However, the second request is reported as erroneous: %MW500 and %MW530 are reported BAD.

If the item %MW530 is deleted, %MW500 becomes GOOD again.

6.19 Specific Items

Specific Items

Description

A specific item is an OPC item which is not related to any PLC variable but is a way to view/modify some internal parameters (internal to the OPC server or internal to the PLC). These items can be used with the test client given with the product, thus avoiding any modifications to your OPC application that may not be reusable with any another OPC server (See OFS client, p. 195).

- A specific item has a path like any other item.
- the definition of a specific item always starts with a '#' character.
- specific item can be created in any group
- Certains specific items (See OPC Item Properties, p. 164) can have an active status in an active group. Thus, the server can automatically detect changes,
- specific items can be read/written within any subset of the group (including both ordinary and specific items).
- to read or write a specific item either synchronous or asynchronous functions can be used

Specific items available for a device can be browsed in the "#Specific" sub-folder attached to any device. The "Diag Buffer" function has been implemented in the form of a set of specific items. They are introduced in a separate section, in addition to the list provided below.

> **Note:** All specific items are disabled if the OPC Extensions is not checked in the "Options" tab of the configuration tool (See Options tab, p. 56).

Name	Type	Access	Can be activated	Limitation
#AppliName	VT_BSTR	R	No	Not available on S7 and S1000

It gives you the name (if any) read from the device.

Name	Type	Access	Can be activated	Limitation
#AppliVersion	VT_BSTR	R	No	Not available on S7 and S1000

It gives you the application version (if any) read from the device.

Name	Type	Access	Can be activated	Limitation
#DisableDevice	VT_I2	R/W	Yes	

If the communication with the device is enabled, the value read is 0, otherwise the value read is 1.
To modify the state, write either 0 or 1.
This item can be used to temporarily disable communication with a device (for instance before modifying the device that will cut communication) to avoid time-outs or any other communication problems.
If the value written is 1, the items related to that device will become "BAD" immediately since the server will stop sending requests to that device. If the value written is 0, the server will once again send all the requests to the device, the items should become "GOOD" again within a few seconds.

Name	Type	Access	Can be activated	Limitation
#MaxChannel	VT_I2	R/W	Yes	

This item is related to the multi-channel (See MODBUS Multi-Channel Feature, p. 154) feature.
Even if it is possible to create it for any device, it only makes sense with Modbus Plus & Modbus TCP-IP devices (Premium excepted).
You can ascertain the maximum number of channels currently configured for this device by carrying out a read. Its value may be the result of the off-line configuration (See Setting the alias properties, p. 44).
You can set the maximum number of channels that can be used to communicate with the device. by carrying out a write. You can either decrease or increase the value and it will be taken into account within a few seconds ; this allows the results of the tuning of the number of channels to be viewed immediately.

Name	Type	Access	Can be activated	Limitation
#NbrMaxPendingReq	VT_I2	R/W	Yes	

Read/Write of the NbrMaxPendingReq parameter for a given device. This parameter is the number of requests that can be queued up to this device before setting overflow state. Usually, this parameter is automatically adjusted by the server.

Name	Type	Access	Can be activated	Limitation
#NbrRequest	VT_I2	R	No	

Its value (a number of requests) is only related to one device (defined by its path). This indicates the number of requests sent to that device, by the server, to refresh its internal cache. This includes all the frequencies that may exist in the server.
This item (See The Different Groups of Items, p. 109) can be created with no path within a synchronous group (name starting with $ or $$). In this case its value is the number of requests necessary to read the whole group.
If this item is created with no path within an ordinary group, its value is always 0. It is only possible for matters of compatibility.

Name	Type	Access	Can be activated	Limitation
#PlcStatus	VT_I2	R/W or R	Yes	This functionality is only available for TSX Quantum and TSX Compact on Modbus, and for TSX Micro, TSX Premium on XWAY and for TSX S7.

The value returned is the PLC mode (1 if the PLC is running, 0 if it is stopped).
It is possible to write the value to force the operating mode of the PLC. To do this, the "API operating mode (See PLC operating mode management, p. 122)" option must have been checked in the "options" tab of the configurator.
Caution: The use of the #PlcStatus is very costly. You are strongly advised to insert this item within a group with a large period.

Name	Type	Access	Can be activated	Limitation
#RefreshDevice	VT_I2	R/W	No	Not available on S7 and S1000

This item is designed to manage the consistency between the symbol table file and the application in the PLC.
If the value 1 is written to this item, the server will read the application name and version from the Device.
If this item is read, the server will perform a consistency check between the application name and version already read from the device and the same information from the symbol table file open for that device. The value returned can be:

- 0 : no check was performed (no symbol table information or device information),
- 1 : everything is OK and consistent,

- 2 : application names are not consistent,
- 3 : application versions are not consistent.

Name	**Type**	**Access**	**Can be activated**	**Limitation**
#TimeOut	VT_I2	R/W	Yes	

Its value (expressed in ms) is only related to one device (defined by its path). This value represents the Frame Time-out which is the time the server will wait for an answer from the device after it has sent a request. Any write will modify the server internal parameter for this device.

6.20 MODBUS Multi-Channel Fearure

MODBUS Multi-Channel Feature

Description

Modbus networks are half-duplex networks which means that after sending one request, the server should wait for the answer before sending the next request. The only way to speed up the communication is to open more than one channel between the sender and the receiver.

With OFS it is possible to do that but only for the Modbus Plus and Modbus TCP-IP networks (Premium excepted).

You can open between 1 and 4 channels for Modbus+ or between 1 and 16 channels for Modbus TCPIP for each device and you can configure that number either statically with the OFS configuration tool (See Setting the alias properties, p. 44) or dynamically with the specific (See Specific Items, p. 150) item #MaxChannel.

The default value is obviously 1. However, the most adequate value for Modbus TCPIP (excepted Premium) and Modbus+ is 2.

For a given device, each time a request is put into the waiting queue because all channels that are already open are busy, OFS will check to see if it is allowed to open one more channel with this device.

For Modbus Plus, since the Modbus Path allocation is dynamic all channels are closed when there is no more request to send (thus it makes sense to configure 2 channels for 6 devices even if there is only one SA85 with 8 paths).

For Modbus TCP-IP, as long as the device is in used (at least one variable related to that device exists), all the open sockets remain open. When the device is no longer used, all sockets but one are closed.

6.21 Dynamic Consistency

Dynamic Consistency

Description

This feature can be used only if:

- The PLC is connected to a Modbus network,
- The PLC is running an application written with the Concept workshop (version 2.1 or later),
- This device been configured (See Associating a Symbols Table File, p. 43) to use the Concept direct link feature (symbol table file of prj type).

This feature enables the server to verify regularly the consistency between the application loaded into the PLC and the Concept symbols database currently open. So if you are running under Windows NT both Concept 2.1 and OFS, if you download some changes into the PLC with Concept, after a few seconds, OFS will close and reload the Concept database (this feature is available under both Windows 2000/98 and NT but under Windows 95/98 it is not always possible to run OFS & Concept at the same time).

The automatic reload of the database can be deactivated for a given device (see Options tab, p. 56). In that case, the reload can be operated manually using the OFS manager (See The OFS Manager, p. 192) "reload and update" service.

OFS will automatically update his networks requests in case some variables locations have changed and if you close and reopen the OPC browse interface, an updated list of symbols will show up.
With Concept, it is possible to use unlocated variables but it is not possible to read their values as long as they are not used. With OFS all unused unlocated variables will be displayed with a "BAD Quality attribute ".

If after an automatic update of the Concept database, OFS detects that some unlocated variables that were unused turn out to be used the "BAD Quality attribute" will be replaced by "GOOD Quality attribute" and an up-to-date data value will be displayed.

To use this feature:

- configure the device with a Concept project file (See Associating a Symbols Table File, p. 43),
- Check the Dynamic consistency option within the Device Property Page
- Start OFS and create at least one variable for the device: if the consistency check is OK between the PLC and the project file, a message should be displayed into the server debug trace "MODBUS : Started Application Dynamic Consistency Check OK"

- Start Concept and open the project file (may be done after or before starting OFS),
- Make and download a modification to the current application,
- The following message should appear in the server debug trace "Application Modification Detected for..."

Note: To work properly with Concept, the automatic saving option should be enabled (under Concept menu Options->Preference->Common, verify that the "save after download" option is checked). If you don’t want to use this option, you should use the save command manually.

6.22 OFS as NT Service

OFS as an NT service

Description

> **Note:** Monitoring an NT service is carried out by starting the service tool in **Parameters/Control Panel/Administration tools/Services** (Windows 2000) or **Parameters/Control Panel/Services** (NT).

To use the NT service feature, you need to make some modifications to the configuration of your machine (this feature can NOT be used in DEMO mode):

- Configure your server (alias, time-out...) preferably using the "hidden" option in the alias tab.
- Check that the OFS server is not already in use and start the Microsoft dcomcnfg tool. This tool's executable file is : Winnt\system32\Dcomcnfg.exe
  - Using this tool, Select the "Schneider-Aut OPC Factory Server" application
  - Select "Properties"
  - Select "Identity"
  - Check "This User" and enter the name and password of a local Administrator-User.
  - Click on "OK"
  - Click on "OK"
  - Close Dcomcnfg
- Run the OFSService.bat batch file which can be found in the directory which contains the server's executable file.
- Start the Services tool.
  OFservice should appear in the list:
  - select OFservice,
  - The default value is "Manual"

  You can then start OFservice and OFS simply by using "Start".
  The same applies for stopping it using "Stop".
  It can be started automatically using "Startup" set to "automatic".
  Close the Services tool.
- Reboot your machine and OFS should run (Use the Windows NT Task manager to verify). Before rebooting you can test that everything is OK by starting OFservice manually.

> **Note:** If the user configured to run the OFS server is not logged on, they will only be able to connect to the server.

Note: To uninstall the OFS product while OFservice is running, proceed as follows:

- stop OFservice,
- to cancel OFservice registration on the NTservice, run the OFSNoService.bat batch file which can be found in the directory which contains the server's executable file,
- uninstall the product.

Note: If you want to delete OFS as an NT service, but want to keep OFS installed, proceed as follows:

- stop OFservice,
- to cancel OFservice registration on the NTservice, run the OFSNoService.bat batch file which can be found in the directory which contains the server's executable file,
- start the DCOMcnfg tool. Select the "Schneider-Aut OPC Factory Server" application, then Properties, then Identity and check "The Interactive user" box. Confirm, close DCOMcnfg and restart the machine.

6.23 PUSH Data Support

Push Data Support

Description

The OFS server can accept Push Data from one device at a time via Modbus and Xway. The data received by the server should fit within the range that has been defined for the Push Data coming from this device.
This feature is supported for Modbus TCPIP and XWAY devices, with the exception of the S7 and S1000 ranges and UNITELWAY, ISAWAY and XWAY TCPIP direct networks.
This data must be sent to the server using a 37h request code for Xway (usually via the WRITE_VAR PL7 function) and the function code 16 for Modbus (usually via Concept EFB WRITE_REG function).
In both cases the features and the behavior of the server are exactly the same.

To use these features, proceed as follows:

1	Create an alias for the device using the configuration tool.
2	Open the device properties page.
3	Define the Push Data range for your device (Base and Size). Example: range %MW1000..%MW1500: base = 1000, size = 500 Example: range 401000..401200 : base = 1000, size = 200
4	Define how this Push Data area should be initialized: values set to 0 or values read from the device.
5	Close the properties page and the configuration tool by saving the parameters.
6	Create an application or use a sample application provided on the CD which is able to send Push Data to the server (check the consistency with the Push Data range for the device mentioned above). Load it into the PLC.
7	Start the OPC test client and connect it to the OFS server.
8	Create an item linked to the device so that the device connection is established and the Push Data area is initialized.
9	A message should appear in the server debug window, indicating that the device is receiving Push Data.
10	Create an item within the Push Data range using the OPC Test client.
11	Start a write from the application.
12	The value of the item should have been updated.
13	You can verify using the server debug interface (Xway or Modbus window), then read the counters from the transaction zone. Slave Request and Slave Answers.

The number of OPC items that can be created within the Push Data area is unlimited (single variables or tables) but it is not possible to create a variable that straddles the area limits.

Besides its value, each OPC item must include the following important attributes:

- quality,
- timestamp.

For items within the Push Data area, the Value attribute is identical for all items and can be either:

- always set to GOOD (if the « Quality Check Rate » defined in the device properties page is 0),
- set according to the device communication status and the device running mode (if the « Quality Check Rate » value defined in the device properties page is set to NN not to 0). Every NN seconds, the server tries to read the running mode of the device:
  - if communication fails, quality is set to BAD,
  - if communication is established and if the running mode is set to RUN, the quality is set to GOOD,
  - if communication is established but the running mode is not RUN (generally speaking it means STOP), the quality is set to UNCERTAIN,

  The « Quality Check Rate » option is not available with Modbus on Premium (TSX or JBUS option).

For items within the Push Data area, when the Timestamp option is set, the date/time is set as follows:

- the current server date and time when a write operation is requested by the OPC client for this item,
- time and date information from the PLC when the server receives new values from the latter,

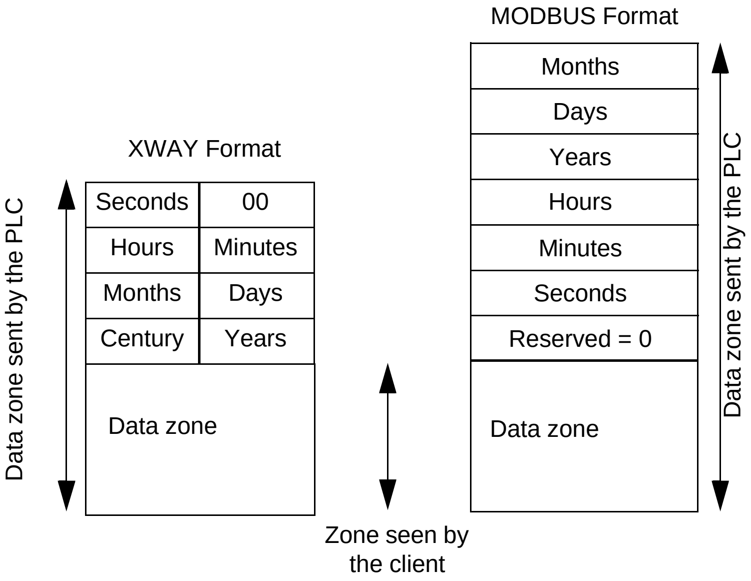
The timestamp option can be activated individually for each alias, from within the Properties page.

In order to transmit the date/time to the server, the PLC must include it as a header with the data transmitted.

It is recommended that you set the PLC time to GMT in order to meet OPC standard.

Request header format is different depending on whether it is an Xway request or a Modbus request.

Illustration of the 2 formats:



Note: With PL7, the date/time can easily be inserted using the RRTC function. With Modbus on Premium, if you are using RRTC, a BCD to INT conversion is required to match the Modbus format.

Some sample Concept and PL7 applications are supplied on the CD (see Appli folder).

To enable the creation of the Push Data area and the reception of related data before any item for this device is actually created, it is possible to configure the device (in the device properties page) so that is preloaded when starting the server.
Any OPC write is performed directly on the device, the Push Data area is not affected in any eventuality.
Any OPC read of the device is performed directly, the Push Data area is updated simultaneously.
For XWAY devices, only %MW and %MD variables can be associated with the Push Data area. The others (%MB, %MF) are handled as if the area were not defined.
For Modbus devices, the Push Data area is always located in the 4x range. Only type INT, DINT or FLOAT variables can be created in the Push Data area.

Note:

- If you use the Push function on a Premium via Modbus TCPIP, and the XIP driver is also active, the IP address of the Premium must not be declared in the latter (the same 502 TCP/IP port is shared).
- a single Push Data area can be created per device. However, if the device can be reached by several network addresses, it is possible to define one Push Data area per address,
- if the server is not shut down normally, it may be necessary to stop/restart the XWAY driver used, because the gate used to receive the Push Data might not have been released.

6.24 Tuning the OFS networking

Tuning the OFS Network interface

Description

The main parameters for tuning the OFS network interface are as follows:

- **Group Min Update Rate** (see Standard parameters and Alias edition, p. 34): the value of this parameter should refer to the update rates you plan to use for your OPC groups. -{}-If you set the value of this parameter to X, the update rates you can use are X, 2X, 3X .. nX. The rule of the thumb is to set it to X/2 if you plan to use rates X, 2X, 3X, nX
  Example:
  If you plan to use update rates that are 1, 2, 5 seconds, you can set this parameter to 500ms.
  If you give this parameter too low a value, you waste the PC's processing time.
- **XWAY** or **Modbus sampling rate** (seer Communication Tab, p. 60) : these parameters refer to the XWAY and MODBUS networks respectively. In both cases, the meaning is exactly the same. The parameter defines the frequency at which the internal entity in charge of RECEIVING the responses will check the network drivers to see if any messages have been received. The rule of thumb is to set it to X/2 if you know that your best device responds within X ms.
  Example:
  If the best device responds within 100ms, set the parameter relating to the protocol (MODBUS or XWAY) to 50 ms.
  If you give this parameter too low a value, you waste the PC's processing time.
- **MaxChannel** for Modbus+ and Modbus TCP-IP devices (see Setting the alias properties, p. 44 and see Specific Items, p. 150): In general, the rule of thumb is that using 2 channels instead of one often has a very positive effect. Using more than 2 channels is not always worthwhile (this depends on the network load and on the type of device used).

For Modbus Plus, it is possible to communicate with, for example, 15 PLCs using only one SA85. But as the SA85 only supplies 8 paths, it can be worthwhile adding another SA85 and configuring certain devices to communicate through this new Modbus adapter (in other words, by modifying the alias address).

6.25 OPC Item Properties

OPC Item Properties

Properties

The IOPCItemProperties interface (see § 4.4-6 of the OPC custom interface specification document) is supported by the OFS server.

The following properties are supported:

- canonical data type,
- value,
- quality,
- timestamp,
- access rights,
- description (only if a comment has been given in the Workshop).

For **Concept variables** only:

- InitialValue (the initial value of a variable),
- VariableKind (the type of variable: elementary, structured, function block, section),
- VariableTypeId (The Type id as known by the Concept tool),
- MemoryArea (areas: 0x,1x,3x,4x, unlocated, unused, etc.),
- AreaIndex (the index inside of the memory area),
- VariableSize (the size, useful for non elementary variables),
- RelativeOffset (the offset inside of a structured variable).

For a given variable, some of them may not be supported if it doesn't make sense (eg : no Description if the variable does not have a comment, no InitialValue if the variable has no initial value, etc.)

To test the use of OPC item properties, you can use OFS client (see OFS client, p. 195).

Example of use:

You want to know when the link between the PC and the PLC is broken. When this is the case, you want to display something special in your OPC client application:

The quality of an item is the specification to use : in general, it is not possible to use the item quality to display something, only its value can be used.

The solution is to create an item, which has a value directly linked to the quality of another item.

When everything is OK the quality value is always 192 (QUALITY_GOOD). When something is wrong the quality value can be something like 24 or 28 (QUALITY_BAD + some reason).

Using OFS client, create a group and an item. When you have done this, reopen the browse interface, reselect the same symbol and click on the Properties button. Select ID 3 (Item Quality) then OK and OK. The value of the new item is the quality of the previous item.

6.26 DCOM Configuration

DCOM Configuration

Description

The OFS server can operate in local mode (the server and the OPC client are located on the same machine) or in remote mode (the OPC client and the server are on different machines connected by DCOM generally via Ethernet TCP-IP).
The remote execution mode requires an additional adjustment using the DCOMCnfg.exe tool provided with the DCOM package.
This package is included with Windows 2000, NT 4.0 or 98, the version needed for Windows 95 is provided with the OFS product (see redist directory).
The server and the client station should be configured appropriately.

Server (Windows 2000 & NT4.0):
These configuration parameters must be defined whilst logged on to the machine with an account having the necessary administration rights to access and start up the server.

1	Start DCOMCnfg.exe located in the Winnt\System32 directory.
2	In the "Default Properties" tab, verify that: ● the option "Enable Distributed COM on this computer" is checked, ● the field "Default Authentication level" is set to "Connect", ● the field "Default Impersonation level" is set to "Identify" or "Impersonate".

3	In the Application folder, select "Schneider-Aut OPC Factory Server" in the list, then click on "Properties". The dialog box "Schneider-Aut OPC Factory Server Properties" will appear. ● click on the "Identity" tab. The "Interactive User" option should be selected. The "This User" option with an appropriate password should be selected if nobody is logged on to the server or if the session is modified or interrupted for no reason, ● click on the "General" tab, the authentication level should be set to "Default" ● click on the "Location" tab, the option "Run application on this computer" should be selected. ● in the "Security" tab, select "Use Custom Access Permission". ● click on "Edit", the "Access Permission" dialog box appears. ● Click on "Add", add the users, then authorize their access (the SYSTEM, INTERACTIVE and NETWORK users should be in this list, others such as Everyone may be added), ● click on "OK" to close the dialog boxes. ● in the "Security" tab, select "Use Custom Launch Permission". ● click on "Edit", the "Launch Permission" dialog box appears, ● click on "Add", add the users, then authorize them to start up the server (the SYSTEM, INTERACTIVE and NETWORK users should be in this list, others such as Everyone can be added), ● click on "OK" to close the dialog boxes.

Note: If the client and the server do not belong to the same NT domain or if there is no NT domain, remember that identical users with identical passwords must be created on both machines (note that it is case sensitive).

Client (Windows 2000 & NT4.0)

These configuration settings must be made while logged on to the machine with an account having the necessary administration rights to access and start the client.

1	Start DCOMCnfg.exe located in the Winnt\System32 directory.
2	In the "Default Properties" tab, verify that: ● the option "Enable Distributed COM on this computer" is checked, ● the field "Default Authentication level" is set to "Connect", ● the field "Default Impersonation level" is set to "Identify" or "Impersonate".
3	In the "**Default security**" tab, modify the "**Default Access Permissions**" list in order to make sure that the users SYSTEM, INTERACTIVE, NETWORK and EVERYONE are present. This last setting is only necessary to allow the server to send back notifications to the client machine. If you know that the client you are using takes care of this, you shouldn’t need this setting.

Server (Windows 95/98):

It is impossible for a client to launch a server running Windows 95/98 remotely. There is also no configuration which relates to "Launch Permissions" in Windows 95/98.

Before going any further, use the Control panel Network icon, make sure that the "Client for Microsoft Network" is installed and configured correctly.

These configuration parameters must be defined whilst logged on to the machine when you have an account with the necessary administration rights to access and start up the client.

1	Using the Control panel Network icon, make sure that the "User Level Access Control" option is selected (in the "Access Control" tab) and that it indicates the name of the security server (either the name of the NT server, or the name of the NT domain) enabling the management of groups and users.
2	Start the DCOMCnfg.exe from the Windows\System directory.
3	In the "Default Properties" tab, verify that: ● the option "Enable Distributed COM on this computer" is checked, ● the field "Default Authentication level" is set to "Connect", ● the field "Default Impersonation level" is set to "Identify" or "Impersonate"
4	In the "Default Security" tab, make sure that the option "Enable Remote Connection" is selected.
5	In the Application folder, select "Schneider-Aut OPC Factory Server" in the list, then click on "Properties". The dialog box "Schneider-Aut OPC Factory Server Properties" will appear. ● in the "Location" tab, the option "Run application on this Computer" should be selected ● in the "Security" tab, select "Use Custom Access Permission", ● click on "Edit", the "Access Permission" dialog box appears, ● click on "Add", add the users, then authorize their access (the SYSTEM, INTERACTIVE and NETWORK users should be in this list, others such as Everyone can be added), ● click on "OK" to close the dialog boxes.

Client Station (Windows 95/98):

Before going any further, use the Control panel Network icon, make sure that the "Client for Microsoft Network" is installed and configured correctly.

1	Using the Control panel Network icon, make sure that the "User Level Access Control" option is selected (in the "Access Control" tab) and that it indicates the name of the security server (either the name of the NT server, or the name of the NT domain) enabling the management of groups and users.
2	Start the DCOMCnfg.exe from the Windows\System directory.

3	In the "Default Properties" tab, verify that: ● the option "Enable Distributed COM on this computer" is selected ● the field "Default Authentication level" is set to "Connect" ● the field "Default Impersonation level" is set to "Identify" or "Impersonate"
4	In the "Default security" tab, modify the "Default Access Permissions" list in order to make sure that the SYSTEM, INTERACTIVE, NETWORK and EVERYONE users are present. This last setting is only necessary to allow the server to send back notifications to the client machine. If you know that the client you are using takes care of this, you shouldn’t need this setting.

Note: In the case of connection problems (Windows 2000, NT 4.0 or 95/98), it may be necessary to modify the check box that relates to DCOM Security in the "Options" tab of the configuration tool. This option is only useful if you are planing to launch the server (manually or as an NT service) before establishing any remote connection. When the machines belong to the same NT domain, only the authenticated users can have access, even via the pseudo-user "Everyone". In other words, even if "Everyone" has access and startup authorization, calls can fail if they are sent by a user connected to a local account on one of the two machines.

6.27 Diag Buffer

At a Glance

Aim of this Section

This section deals with the installation of the Diag buffer and its main features.

What's in this section?

This section contains the following topics:

Functioning from an OPC client

Reminder about the Diag Buffer

The Diag buffer (See Definition of the Diag Buffer, p. 82) is a feature which detects faults on monitored elements and transmits error messages to the visualization system (known as the viewer).

Note: The implementation of the diagnostics DFBs in the PLC is necessary for Diag Buffer functioning.

Description of the client interface

Diag Buffer functions authorize access to PLCs using specific items.

The table below shows the specific items:

Utility	Item	Type	Access	Value read	Value to write
Open connection	#DiagLogon	VT_UI2	READ/WRITE	Viewer or 0xFFFF identifier	zone number
Close connection	#DiagLogout	VT_UI2	READ/WRITE	Viewer or 0xFFFF identifier	not important
Read next error	#DiagReadNextError	VT_UI1+VT_ARRAY	READ	Error	
Error acknowledgement	#DiagAckError	VT_UI2	WRITE		error ID number see Information retrieved by the Diag buffer at the top of the table, p. 182
Evolution Status	#DiagReadStatus	VT_UI4	READ/WRITE	Status0 + Status1	Status handle

Type corresponds to OPC standards:

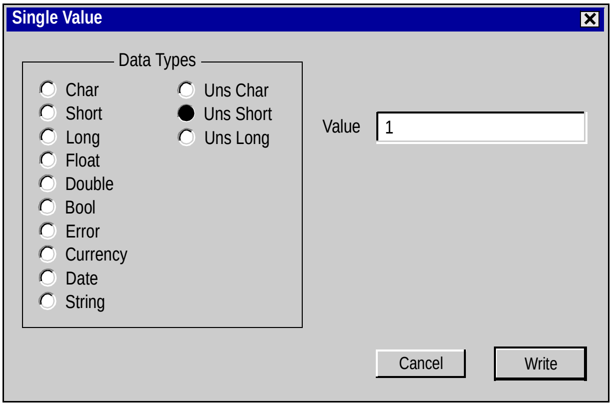
- VT = variant type,
- UI1 = unsigned integer on 1 byte,
- UI2 = unsigned integer on 2 bytes,
- UI4 = unsigned integer on 4 bytes,
- ARRAY = table of bytes

#DiagLogon specific item

Type	Access	Can be activated	Limitation
VT_UI2	R/W	no	

This item enables connection to the PLC. Firstly, the number of the zone that you wish to monitor must be indicated on the PLC (between 0 and 15) by carrying out a WRITE function.

Example of a write on #DiagLogon:



● Uns Short

Value 1

Value to write:

- bit i = 1: the zone is displayed,
- bit i = 0: the zone cannot be displayed.

  Bit 0 corresponds to zone 0, bit 15 corresponds to zone 15.

  Examples:
  - to monitor zone 6: write the value 0040h
  - to monitor zones 2 and 15: write the value 8004h

Value returned after read:

- the viewer number is displayed if the connection is open, if not the connection is not established and 0xFFFF is returned.

Value returned by the item:

HRESULT	Comment
OFS_E_DIAG_OK	OK
OFS_E_DIAG_NO_BUFFER	Diag buffer not activated
OFS_E_DIAG_MMI_ALREADY_CONNECTED	The viewer is already connected

HRESULT	Comment
OFS_E_DIAG_BUFFER_FULL	Diag buffer is full
OFS_E_DIAG_TOO_MUCH_MMI	All possible viewers (15) are connected

Note: To monitor all zones, write the value FFFFh or 0 in #DiagLogon.

#DiagLogout specific item

Type	Access	Can be activated	Limitation
VT_UI2	R/W	no	

This item allows PLC disconnection.

Value to write:

- not important,

Value returned after read:

- if the disconnection is successful, the OxFFFF value is returned, if not the viewer number is returned again.

Value returned by the item:

HRESULT	Comment
OFS_E_DIAG_OK	OK
OFS_E_DIAG_NO_BUFFER	Diag buffer not activated
OFS_E_DIAG_BUFFER_FULL	Diag buffer is full
OFS_E_DIAG_WRONG_MMI_ID	The viewer identifier is not valid (outside range 1 to 15)
OFS_E_DIAG_MMI_NOT_CONNECTED	OPC client not connected

Note: Destruction of the #DiagLogon item will automatically disconnect you from the viewer, without using the #DiagLogout item.

#DiagReadNextError specific item

Type	Access	Can be activated	Limitation
VT_UI1 + VT_ARRAY	R	yes	

This items enables you to read errors in the diag buffer memory.

Value to write:
- nothing,

Value returned after read:
- errors saved in the form of a 120 byte table (See Diag buffer table formats, p. 181).

Value returned by the item:

HRESULT	Comment
OFS_E_DIAG_OK_READ_NOTHING	Read successful, no modification is recorded in the 120 byte table
OFS_E_DIAG_OK_READ_MODIFIED	Read successful, modifications recorded in the 120 byte table (the error has been acknowledged or has disappeared)
OFS_E_DIAG_OK_READ_NEW	Read successful, a new table has been created (a new error has appeared)
OFS_E_DIAG_NO_BUFFER	Diag buffer not activated
OFS_E_DIAG_BUFFER_FULL	Diag buffer is full
OFS_E_DIAG_WRONG_MMI_ID	The viewer identifier is not valid (outside range 1 to 15)
OFS_E_DIAG_MMI_NOT_CONNECTED	OPC client not connected

#DiagAckError specific item

Type	Access	Can be activated	Limitation
VT_UI2	W	no	

This item allows alarm acknowledgement.

Value to write:

- the value on 2 bytes corresponding to the "N identifying the error" zone starting by a read of the highest ranking bit (the first two bytes in the table).
  E.g. the value returned in the "N identifying the error" zone of the #DiagReadNextError item table is such that: Var[0] = 04h, Var[1] = 05h. The value to write in the item #DiagAckError is 0504h.

Value returned after read:

- nothing.

Value returned by the item:

HRESULT	Comment
OFS_E_DIAG_OK	OK
OFS_E_DIAG_NO_BUFFER	Diag buffer not activated
OFS_E_DIAG_BUFFER_FULL	Diag buffer is full
OFS_E_DIAG_MMI_NOT_CONNECTED	OPC client not connected
OFS_E_DIAG_WRONG_ERROR_ID	Non authorized error identifier
OFS_E_DIAG_ERROR_NOT_USED	No error corresponds to this identifier

#DiagReadStatus specific item

Type	Access	Can be activated	Limitation
VT_UI4	R/W	no	

This item lets you to know the evolution of the status of a DFB error without having to wait to be notified of a change in the 120 bytes error table (See Diag buffer table formats, p. 181).

Value to write:

- the value on 4 bytes corresponding to the "Status Handle" zone starting by a read of the highest ranking byte.
  E.g. the value returned in the "Status Handle" zone of the #DiagReadNextError item table is such that:

  Var[8] = 98h, Var[9] = 01h, Var[10] = 76h, Var[11] = 25h

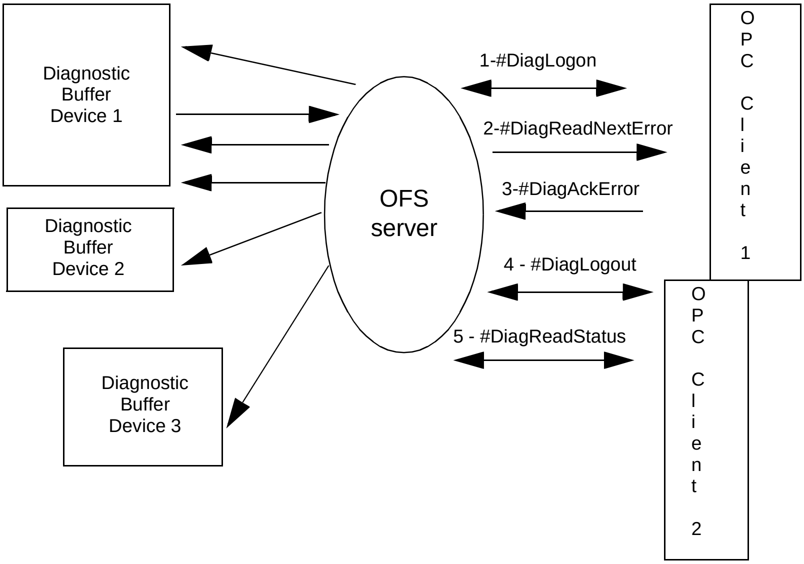
The value to write in the #DiagReadStatus item is 25760198h or 628490648d.

Value returned after read:

- the values of status 0 + status1, taking into account the value of the words from right to left.
  E.g. the value returned is 0010001Dh; status0 value is 001Dh; status1 value is 0010h.

Description of client functioning

The diagram below shows the functioning principals of an OPC client through specific items:



With the OFS server several PLCs can be monitored at the same time, it has a multi-station function (unlike the PL7 which can only manage one PLC at a time). To supervise several PLCs at once, simply create other aliases in the configuration tool and add them to another group belonging to the same client (minimum of 1 group per device to monitor).

Diag Buffer Management

Errors recorded in the diag buffer memory can have the following statuses:

- active or inactive,
- acknowledgement requested or acknowledgement not requested,
- if acknowledgement is requested, the error may can be acknowledged or not acknowledged.

Note: Only errors from the diag buffer can be acknowledged. An error displayed on several viewers will be deleted from all viewers once it has been acknowledged on one viewer.
An alarm is deleted from the buffer if:

- the alarm no longer exists,
- all viewers have read the alarm,
- the alarm has been acknowledged (after an acknowledgement request).

Use of the Diag buffer

General

Before starting up an OPC client, it is advisable to create aliases for each of the PLCs to be monitored. In order to make the installation of the diag buffer easier. With these aliases it will be easier to declare PLC addresses during the creation of an OPC client.

When an OPC client wishes to use the diag buffer, they must define a handle and use only this handle during the creation of a group.
To do this, at each call of the IOPCServer::AddGroup( ) method, the hClientGroup parameter (4th parameter) must contain a single value. This value corresponds to the client's clientHandle.
As this value must also be unique amongst all the OPC clients using the diag buffer, the following procedure must be considered:

- if during the connection, the return code **OFS_E_DIAG_MMI_ALREADY_CONNECTED** is transmitted, it means that the clientHandle is already in use. Another value must therefore be used.
  In order to do this, consult the window which can be accessed via the General Menu:>NetManXWayWindow and extend the branch Devices<> @Device<>DiagBuffer connections which gives the list of connected viewers (handle + MMI id).

  Possible values for the clientHandle are between 0 and 2^{32} - 2 (0 à 0xFFFFFFFE). The value 0xFFFFFFFF is reserved.

Example of the settings of the handle with the C++ test client delivered on the OPC Factory Server CD:

- create a short cut on the executable file OFSClient.exe,
- in the properties of the short cut, add onto the end of the line "Target"="C:\ ...\OFSClient.exe" -h10 for example to fix a handle = 10 for this OPC client.

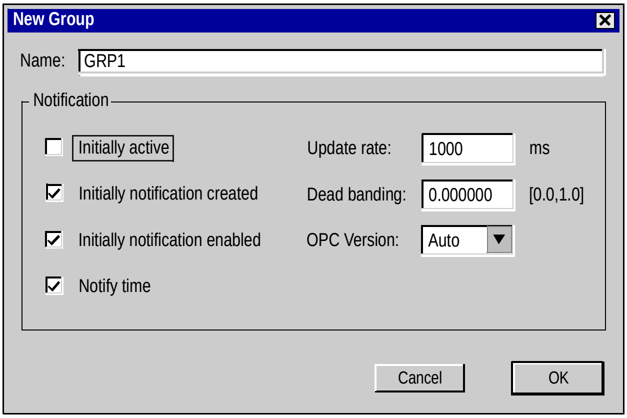
All the examples on the following pages use the test client supplied on the CD. For more information on the OPC client see the OFS Client (See OFS client, p. 195)section.

Procedure for installing the diag buffer

As a general rule you need to create two groups per OPC client and then respect the following sequence:

- create an inactive group,
- add the specific items (#DiagLogon, #DiagLogout, #DiagAckError, #DiagReadStatus),
- connect to the zone to be monitored (use of #DiagLogon),
- create an active group,
- add the item #DiagReadNextError.

- **an inactive group**:



New Group

Name: GRP1

Notification

□ Initially active

☑ Initially notification created

☑ Initially notification enabled

☑ Notify time

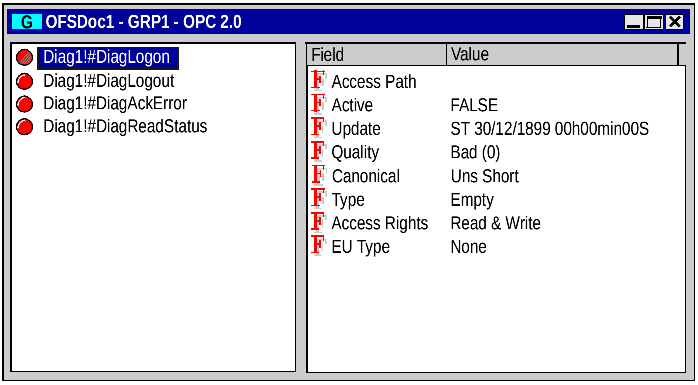
Update rate: 1000 ms

Dead banding: 0.000000 [0.0,1.0]

OPC Version: Auto

1- To connect to the diag buffer, the OPC client must add the #DiagLogon (See #DiagLogon specific item, p. 171) specific item to the group. The connection is established when the OPC client writes and validates the zone number of the PLC to be monitored in this item. If the write is successful, the client obtains its "viewer identifier" number by carrying out a read (1 if it is the first connected)
2- To disconnect from the diag buffer, the OPC client must add the #DiagLogout (See #DiagLogout specific item, p. 172) specific item to the group. The disconnection will be carried out when the client writes any value in this item.
3- To acknowledge an error, the OPC client requires the #DiagAckError (See #DiagAckError specific item, p. 174) specific item in the group.
4- To update the Diag Buffer status, the OPC client needs to add the #DiagRead-Status (See #DiagReadStatus specific item, p. 174) specific item.

The following screen shows the installation of specific items:

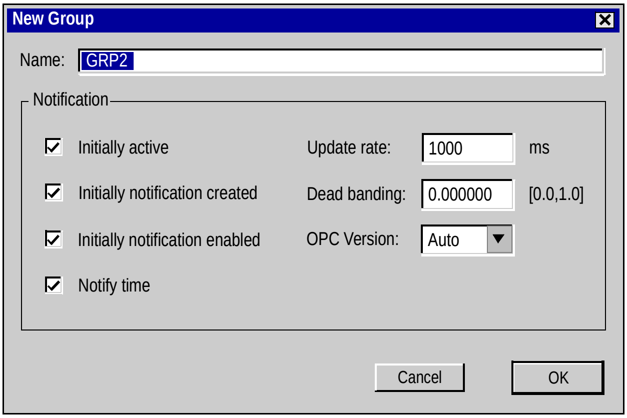


OFSDoc1 - GRP1 - OPC 2.0

- Diag1!#DiagLogon
- Diag1!#DiagLogout
- Diag1!#DiagAckError
- Diag1!#DiagReadStatus

Field	Value
Access Path	
Active	FALSE
Update	ST 30/12/1899 00h00min00S
Quality	Bad (0)
Canonical	Uns Short
Type	Empty
Access Rights	Read & Write
EU Type	None

- **An active group**: this group must be created or activated after actual connection with the #DiagLogon item.



New Group

Name: GRP2

Notification

☑ Initially active

☑ Initially notification created

☑ Initially notification enabled

☑ Notify time

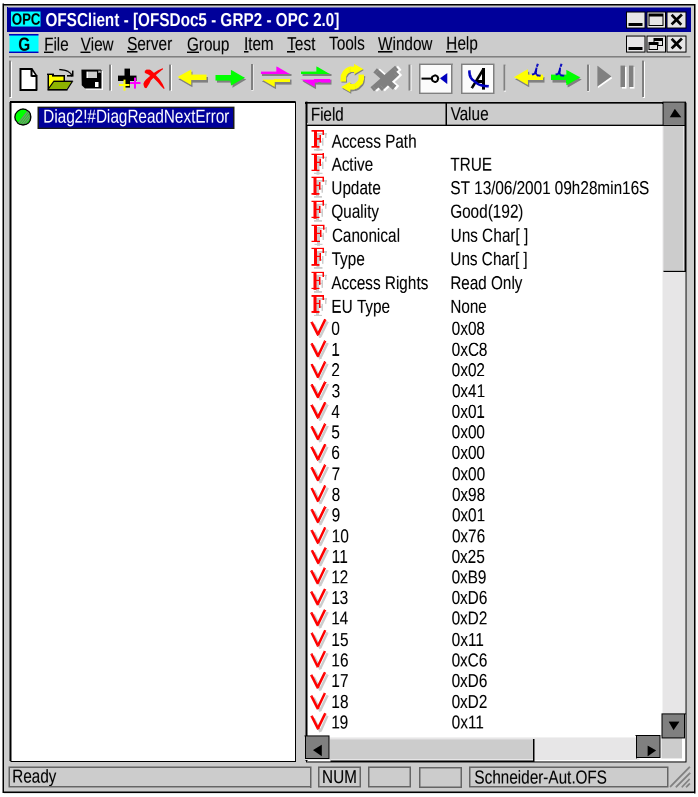
Update rate: 1000 ms

Dead banding: 0.000000 [0.0,1.0]

OPC Version: Auto

To reset the alarms coming from diag buffer, the client needs to add the #DiagReadNextError (See #DiagReadNextError specific item, p. 173) specific item to the group.

The screen below shows a 120 byte table (See Diag buffer table formats, p. 181) in which the error (See Diag buffer table formats, p. 181) code of the activated alarm can be found. Each byte represents a specific piece of information:


OPC OFSClient - [OFSDoc5 - GRP2 - OPC 2.0]
G File View Server Group Item Test Tools Window Help
Diag2!#DiagReadNextError
Ready
NUM
Schneider-Aut.OFS

Field	Value
F Access Path	
F Active	TRUE
F Update	ST 13/06/2001 09h28min16S
F Quality	Good(192)
F Canonical	Uns Char[ ]
F Type	Uns Char[ ]
F Access Rights	Read Only
F EU Type	None
V 0	0x08
V 1	0xC8
V 2	0x02
V 3	0x41
V 4	0x01
V 5	0x00
V 6	0x00
V 7	0x00
V 8	0x98
V 9	0x01
V 10	0x76
V 11	0x25
V 12	0xB9
V 13	0xD6
V 14	0xD2
V 15	0x11
V 16	0xC6
V 17	0xD6
V 18	0xD2
V 19	0x11

Example of translation of bytes V12 to V15 which represent the time of the beginning of the alarm. The table describing the time stamp format of the Diag buffer enables you to extract the different values.

The values read are: V15=11h, V14=D2h, V13=D6h, V12=B9h.

	V15		V14		V13		V12	
Hexadecimal	1	1	D	2	D	6	B	9
Binary	0001	0001	1101	0010	1101	0110	1011	1001

Decoding	4	7	9	13	26	57
Date	Years	Months	Days	Hours	Minutes	Seconds

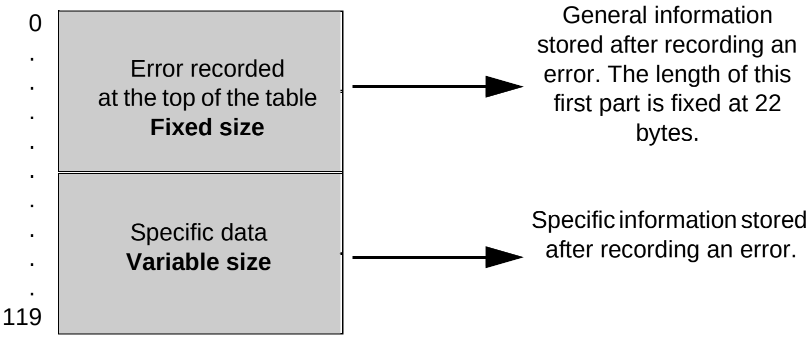
Calculation of the year: 4 + 1997 = 2001

The result is therefore 13h26min57s, 9/07/2001.

Diag buffer table formats

Description

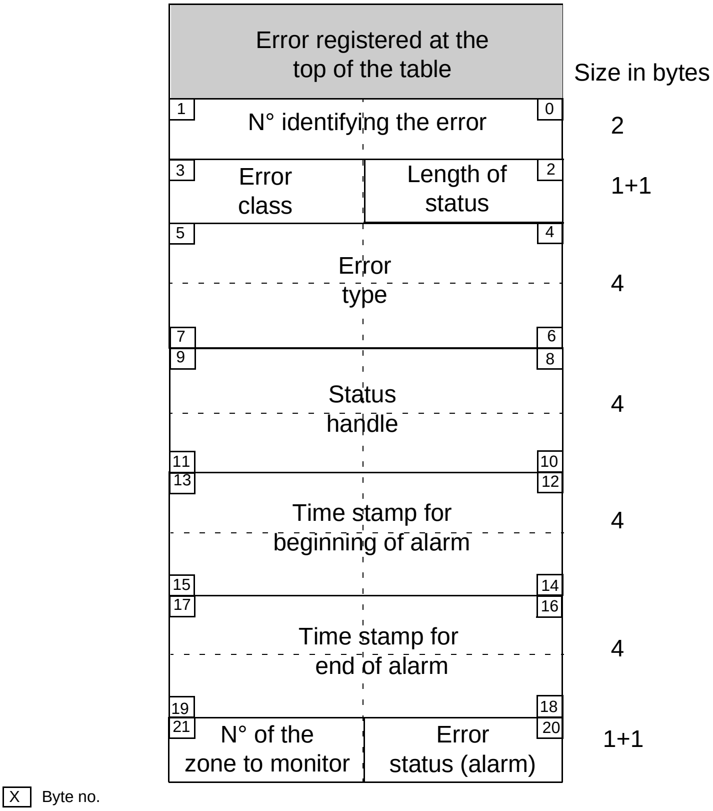
The 120 byte table (See Procedure for installing the diag buffer, p. 178) (alarm reset after a read on #DiagReadNextError) is structured in the following way:
Illustration of the structure of the bytes table:


0
119
Error recorded
at the top of the table
Fixed size
Specific data
Variable size
General information stored after recording an error. The length of this first part is fixed at 22 bytes.
Specific information stored after recording an error.

Information retrieved by the Diag buffer at the top of the table

Description

The illustration below details the contents of the first 22 bytes in the table:



Definition of the contents of the table

- Error identifying number (coded on 2 bytes): an identifying number which is given for acknowledgement. This number must be written in the #DiagAckError item to acknowledge an alarm,
- Length of status (coded on 1 byte): depends on the DFB which has been programmed. If the value is 2, it is "status 0" which means that in the "error type", the value status 0 will be read. If the value is 4, it is "status 0 & status 1" which means that in the "error type" status 0 & status 1 will be read,
- Error class (coded on 1 byte): determines the class of the error.

The table below gives the definition of the code retrieved in this byte:

Symbol	Value	Comment
OFS_DIAG_CLASS_DFB_EV_DIA	0x40	EV_DIA error
OFS_DIAG_CLASS_DFB_MV_DIA	0x41	MV_DIA error
OFS_DIAG_CLASS_DFB_NEPO_DIA	0x42	NEPO_DIA error
OFS_DIAG_CLASS_DFB_ALARM	0x43	ALRM error
OFS_DIAG_CLASS_DFB_USERA	0x4A	DFB USER error
OFS_DIAG_CLASS_DFB_USERB	0x4B	DFB USER error
OFS_DIAG_CLASS_DFB_USERC	0x4C	DFB USER error
OFS_DIAG_CLASS_DFB_USERD	0x4D	DFB USER error
OFS_DIAG_CLASS_DFB_USERE	0x4E	DFB USER error
OFS_DIAG_CLASS_DFB_USERF	0x4F	DFB USER error
System error class		
OFS_DIAG_CLASS_DFB_SYSTEM_ASI0	0x80	STGENE of ASI_DIA error
OFS_DIAG_CLASS_DFB_SYSTEM_ASI1	0x81	STSLABS of ASI_DIA error
OFS_DIAG_CLASS_DFB_SYSTEM_ASI2	0x82	STSLKO of ASI_DIA error
OFS_DIAG_CLASS_DFB_SYSTEM_ASI3	0x83	STSLNC of ASI_DIA error
OFS_DIAG_CLASS_DFB_SYSTEM_IO	0x84	IO_DIA error
New characteristic of the PL7v4		
OFS_DIAG_CLASS_DIAGSYSTEM	0x85	system error (Task, Arithm)
OFS_DIAG_CLASS_SYT_LOCALIO	0x86	LOCAL IO error
OFS_DIAG_CLASS_SYT_REMOTIO	0x87	REMOTE IO error
OFS_DIAG_CLASS_SYT_BUFFERFULL	0x88	Diag Buffer full

Note: For a description of the OFS error codes see (Lists of Diag buffer error codes, p. 231).

- Error type (coded on 4 bytes) this is the type of error which is retrieved by the diag buffer:
  - Diag-DFB error: status value, coding on 2 bytes for "length of status" = 2, 4 bytes for "length of status" =4.
  - Grafcet: system error, occurs when the execution time exceeds the expected time.
  For more information see the PL7 documentation on DFBs.

- Status handle (coded on 4 bytes): this value must be used during a #DiagRead-Status write,

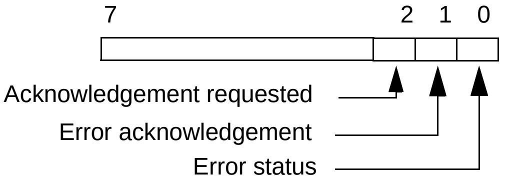
- Time stamp at the beginning of the alarm (coded on 4 bytes): time and date when the alarm was triggered,

- Time stamp at the end of the alarm (coded on 4 bytes): time and date when the alarm disappeared,

Time stamp format:

Field	Comment	Bits	Value	no. of bits
Sec	seconds	0 - 5	0 - 59	6
Min	minutes	6 - 11	0 - 59	6
Hour	hours	12 - 16	0 - 23	5
Day	days	17 - 21	1 - 31	5
Mon	month (January = 1)	22 - 25	1 - 12	4
Year	current year - 1997(2001 = 4)	26 - 31	0 - 63	6

- Error status (alarm): the instantaneous status of the current alarm,



  - bit 0: error status:
    0 : error has disappeared,
    1 : error is active.
  - bit 1: acknowledgement of the error:
    0: error has been acknowledged,
    1: error has not been acknowledged or acknowledgement has not been requested.
  - bit 2: type of alarm (with or without acknowledgement):
    0: acknowledgement not requested,
    1: acknowledgement requested.

- Number of the zone to be monitored: PLC zone from where the diag buffer retrieved the error. Grafcet errors always belong to the communal zone.

Specific information sent back by the Diag buffer in the table

Specific data errors

There are two types of specific data:

- DFB specific data,
- "other" specific data.

Diag buffer specific data

The diagram below describes the Variable Size Specific data section for error classes between OFS_DIAG_CLASS_DFB_EV_DIA and OFS_DIAG_CLASS_DFB_SYSTEM_IO (see Definition of the contents of the table, p. 182):

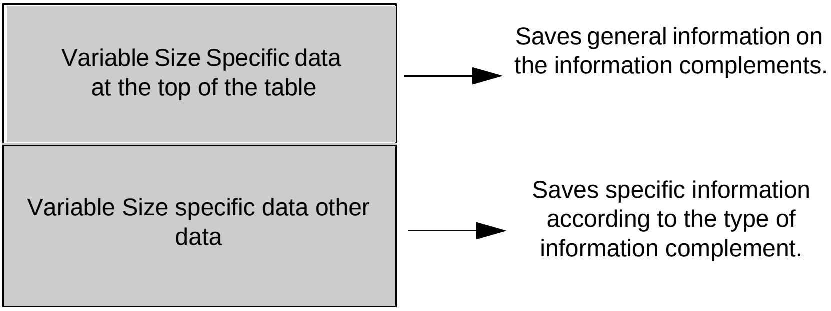
Specific DFB errors	Size in bytes
Length of comment (byte) + comment	1 + variable
Length of the "instantiated" name (byte) +	1 + variable
Length of the type of DFB (byte) + type of DFB	1 + variable
Length of the program address + program address	1 + variable

Definition of the contents of the table

- Length of comments + comments:
  Firstly, the content is a "length" variable, which gives the length of the comments and then the DFB error message.
- Length of the "instantiated" name + "instantiated" name:
  Firstly, the content is a "length" variable, which gives the length of the "instantiated" name then the DFB "instantiated" name.
- Length of file name + file name:
  Firstly, the content is a "length" variable, which gives the length of the file name then the file name.
- Length of the program address + program address:
  Firstly, the content is a "length" variable, which gives the program address length then the program address which corresponds to a DFB execution error.

"Other" specific data

The diagram below describes the Variable Size Specific data section for error classes between OFS_DIAG_CLASS_DIAGSYSTEM and OFS_DIAG_CLASS_SYST_BUFFERFULL (see Definition of the contents of the table, p. 182).
Specific data gives more information according to the error class recorded.
Illustration:


Variable Size Specific data at the top of the table
Saves general information on the information complements.
Variable Size specific data other data
Saves specific information according to the type of information complement.

Variable Size Specific data at the top of the table

The diagram below gives the structure of Variable Size Specific data at the top of the table:

Additional information at the top of the table		Size in bytes
Length of comment (bytes) + comment		1 + variable
Length of the "instantiated" name (bytes) + "instantiated" name		1 + variable
Type	Size information	1 + 1

- Length of comments + comments:
  Firstly, the content is a "length" variable, which gives the length of the comments and then the DFB diagnostics error message.
- Length of the "instantiated" name + "instantiated" name:
  Firstly, the content is a "length" variable, which gives the length of the "instantiated" name then the "instantiated" name of the diagnostic error.
- Size information:
  The content gives the size of the buffer's additional information.
- Type:
  The content gives the type of additional information of specific data.

6.28 Recommendations for using the server

Recommendations

At a Glance

This chapter contains several recommendations to enable optimum use of the server. As a general rule, you should remember that the limit on the number of items that are simultaneously accessible is linked to the communication resources between the OFS server and the devices. The limiting element is the entrance of communication modules on to PLCs.

- for a group containing a large number of items (several thousands), the creation of items or the modification of the group properties (update rate, for example) is made much quicker by deactivating the group beforehand and then reactivating it when the operation is complete. This point is particularly important when using synchronous groups ($ and $$), as for each item created, the destination of the new item is checked with respect to the first item created in the group.
- when using a large number of items (several thousands), divide them up into several groups to adapt the update rate and therefore be able to desynchronize them. In order to avoid communication peaks with devices:
- when developing an application, it is preferable to use the "AddItems" method which is more powerful than the simple "AddItem" method.

User Guide

7

At a Glance

Aim of this Chapter

The aim of this chapter is to introduce the user guide.

What's in this chapter?

This chapter contains the following topics:

OPC Custom

Description

The OPC custom interface of the OFS product is compatible with version 1.0A and version 2.0.

The OPC custom interface 2.0 is just a more complete version of the OPC custom interface 1.0A. There is therefore no problem of COMPATIBILITY between interfaces 1.0A and 2.0.

The new interfaces introduced by version 2.0 are as follows:

- IOPCCommon,
- IOPCItemProperties,
- ConnectionPointContainer,
- IOPCAsyncIO2.

OPC specifications documents are included on the OFS CD-ROM (OPC_Spec\OPC1.0A and OPC_Spec\OPC 2,04):

- OPC 1.0A\Opcfinal.doc is document "OPC 1.0A standard document" (covering both the Custom and Automation interfaces),
- OPC 2,04\Opcovw.pdf is the document "OPC overview 1.0 document",
- OPC 2,04\Opccomn.pdf is document "OPC Common definitions 1.0 document",
- OPC 2,04\Opcda204_cust.pdf is "OPC Data Access Custom Interface 2.03 document".

This document can be downloaded from the OPC Foundation web site : www.opcfoundation.org

This documentation does not deal with the description of the direct use of this interface.

Description of the displayed OLE Automation interface

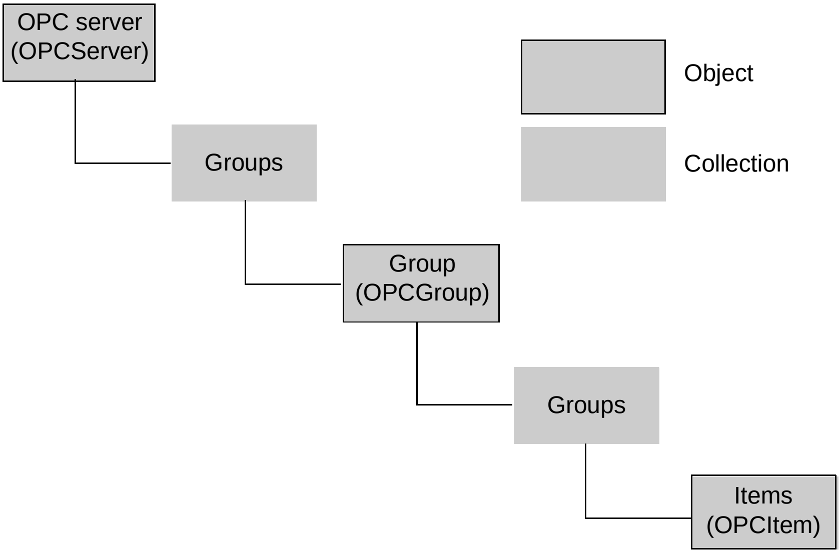
Description

The OPC Factory Server product is compatible with the two OPC Automation standards 1.0 and 2.03, a DLL for each standard version is provided with the product.
The OPC Automation standards 1.0 and 2.0 are completely different, there is **no compatibility** between these interfaces.
It is therefore only by compatibility with earlier versions that the OPC Automation standard 1.0 is supported by OPC Factory Server. The OPC Automation DLL 1.0 file supplied is exactly the same DLL file as the one that came with earlier versions of OFS.

The OPC product documents are included on the OFS CD-ROM (OPC_Spec\OPC1.0A and OPC_Spec\OPC 2.03):

- OPC 1.0A\Opcfinal.doc is the complete document on the OPC 1.0A standard (covering both the Custom and Automation interfaces),
- OPC 2.03\Opcda20_auto.pdf is the document on the OPC Automation 2.02 interface.

Object model of the OLE Automation interface of an OPC 2.03 server:



The OFS Manager

Description

The OFS Manager is a troubleshooting and tuning tool, which ONLY works with the OPC Factory Server (local or remote) or the OFS simulator.
All the troubleshooting features of the OFS Manager are available from the server debug interface. These features are especially useful for remote troubleshooting or when OFS is running without a debug interface (E.g.: when it is operating as an NT service).
The OFS Manager tuning features are only accessible through OFS Manager.
If the OFS Manager is not selected in the OFS configuration tool (See Options tab, p. 56), the OFS Manager operates in read only mode (no modification possible).
There is no link between the OFS Manager and the OFS Configuration tool. That means that any modifications (alias, debug modes,) are only valid for the current instance of the server. If you close the server and re-open it, all your modifications will be lost.

Connection to the server: Menu **Server** -> Connect
Alias Management: select an alias and then use the **Alias** menu or **right click**:

- Alias creation: it is possible to create up to N dynamic aliases. N is the value given in the Options (See Options tab, p. 56) tab page.
- Alias modification: is used to modify the network address or the name of the symbol table file
- Alias deletion: is used to delete an alias from the list

The modifications made only apply for the current server session. Elles ne sont pas mémorisées lors des lancements suivants.

Symbol tables:
It is possible to ask the server to close a symbol table file that is already open and then to reopen it. No changes will be made to the items that already exist. However the list of symbols will be updated (E.g.: for the OPC browse interface).
To do this, first select the symbol table filename from the file names shown in the Symbol tables window. Then use the **Symbol Table** menu or **right click**.

Debug mode management:
The OFS server has three debug options:

- Verbose Mode: this is a complete viewing mode, messages are shown in the Debug window.
- Symbol Mode: this allows messages with additional information relating to the symbol tables to be displayed in the Debug window.
- Request: it enables information relating to the generation of network requests to be displayed in the XWAY or MODBUS Debug window.

Management of log files :
The OFS server can store messages in 3 different log files (one for the main Diagnostic window, one for the MODBUS window and another for the XWAY window).
With OFS Manager you can open or close any of these files whilst the server is running.
To open or close the file, select the file in question in the "Log files" window and use the **Log** menu or **right click**.

Displaying information:
The information display is static by default (no refresh).
To refresh the display, use the **View** -> **Refresh** menu.
To automatically refresh the display, use the **View** -> **Auto Refresh** menu. By default the refresh rate is set to every second. You can modify this rate with the **View** -> **Options** menu.
If you select the **Status** window, the OFS Manager displays general information (identical content to the "Status Window" of the debug interface).
If you select the **Protocols** window followed by a protocol (XWAY or MODBUS), the OFS Manager displays statistical information relating to the chosen protocol. This information is exactly the same as that which appears in the XWAY or MODBUS windows of the server debug interface.
For each device connected to the server, you can display some debug information. Select the device below its protocol (XWAY or MODBUS), in the left hand part of the OFS Manager display. The information displayed is the same as that which appears on each device with the server debug interface.

Reload facility:
For devices associated with a Concept PRJ project, the symbol table can be reloaded manually using the device->reload and update menu. The menu is activated by selecting the device in the list of devices, under the MODBUS protocol.

Displaying error and debug messages:
As long as you are connected to the OFS server, all error messages are displayed in the Errors text zone.
If you are interested in all the messages (including the warning and information messages) you can activate the Debug text box from the **View**-> **Debug Messages** menu.

Saving information:
From the File -> Save As menu, you can save all the information in a .txt file held by the OFS Manager (Alias list, Messages, Counter values,)at any time. This is the only way of saving this type of information (impossible with the server debug interface).

Display of server information:

The heading **Server Information** displays the name of the server, type of product, version and functioning mode (normal or simulated). This is especially useful when the server operates in hidden mode and/or in NT service (without interface)

OFS client

Description

OFS Client is an OPC client supplied with the OFS server as a **test tool**. It is a standard OPC client that can be used with **any OPC** server. To install it onto your machine, select the "Sample application in C++" option during the installation of the server station or the option "OFS server test client" during the installation of the client station.

The target of the test is the custom interface (both OPC 1.0A and 2.0).
The main uses of this client are as follows:

- testing the OFS server installed (either local or remote), especially any feature which is not working correctly with your OPC client,
- tuning the OFS server using certain specific items that perhaps you do not want or are not even possible, in your OPC application. The specific (See Specific Items, p. 150) items corresponding to each device are available with the OPC browse interface.

Main features of the OFS client:

- connection: local (the OFS server by default) or remote : it should start the OFS server if it is not already running,
- OPC interfaces: with the Server -> Interfaces menu, it is possible to see the list of OPC interfaces supported,
- Common OPC: with the Server-> Common menu, it is possible to read/write the client name and the local identification,
- creating a group: with the Group -> New Group menu: you can assign values to parameters from the main group and select the interface version to use (OPC Version 1.0A or 2.0). This is a useful function for the notification mechanism (OnDataChange or IConnectionPoint) as well as for the asynchronous read/write (auto means OPC 2.0 if it is available),
- creating an item: using the Item -> New menu, you can select ONE item and add it to the group; using the Group -> Add Items menu, you can select and validate several items at once or add them to a group,
- group properties : using the Group-> Properties menu, you can activate/ deactivate the group and modify the update rate. You can choose to perform a group read from the cache or from the device,
- group read/write : using the Group menu, you can read/write all the values of a group with the OPC Sync or OPC Async interface functions (first select an existing group),
- item read/write : using the Item menu, you can add/remove or read/write a single item (first select the item in a group),
- display window : using the Window menu, it is possible to create (for an existing group) a List Window where you can see the values of all the items in a group, or a Trace Window which shows the information about the notifications,

- OPC item properties : using the browse interface (opened with the Item -> New menu), select a symbol and click on the Properties button. Select the property identification (on the left-hand side of the dialog box) then OK.

Performance

8

At a Glance

Aim of this Chapter

The aim of this chapter is to describe the OFS server's performance.

What's in this chapter?

This chapter contains the following Sections:

8.1 Introduction to the Performance Chapter

OFS Performance

Description

This part is dedicated to introducing two different aspects of OFS server performances : the evaluation of static performances, or the maximum number of items declarable for a given device according to certain usage criteria, and the evaluation of dynamic performances, or in a given configuration the measurement of the exchange times and value updates.

8.2 Static Performances

At a Glance

Aim of this Section

The aim of this section is to describe the static performances of the OFS.
OFS is provided with algorithms dedicated to optimizing the use of the network requests. The items declared on a given device are therefore regrouped as much as possible within the same requests so that the number of devices is minimized.

What's in this section?

This section contains the following topics:

Number of Lodgeable Items in a Single Request

Item Consistency

The following table shows the number of items, which are of the same type and consecutive, and which can be lodged in a single request. They also show the characteristic of being accessed in a single PLC cycle and therefore showing the coherent values with regards to the PLC (except in the particular case of gate 7 on XWAY TCPIP, for which access is achieved in an asynchronous cycle).
These measures have been carried out using table type items. In XWAY, when you use simple items, deduct about 8 from the total obtained (e.g: 985 instead of 992 on XWAY TCP-IP).
Reminder: For Series 7 and S1000 PLCs, the option "Series 7 PLC" has been selected in the properties page of the configuration tool.

Reading:

Communication medium	No. of elements consistent with one another (dep. on type)			
	bits	words	Double and floating words	Characters (string)
XWAY/TCP-IP	992	124	62	248
XWAY TCP-IP and ETHWAY service 1K	4064	508	254	1016
ETHWAY	992	124	62	248
FIPWAY	480	60	30	120
ISAWAY	920	115	57	230
UNI-TELWAY (terminal port)	480	60	30	120
MODBUS Located variables	2000	125	62	-
MODBUS Unlocated variables Concept 2.5 and above	246	122	61	-
MODBUS Unlocated variables Concept 2.2 and below	246	123	61	-
MODBUS TCP-IP on Premium	1000	125	62	-
ETHWAY/FIPWAY/ UNITELWAY on Series7	480	60	30	-
ETHWAY on S1000	4088	511	255	1022

Writing:

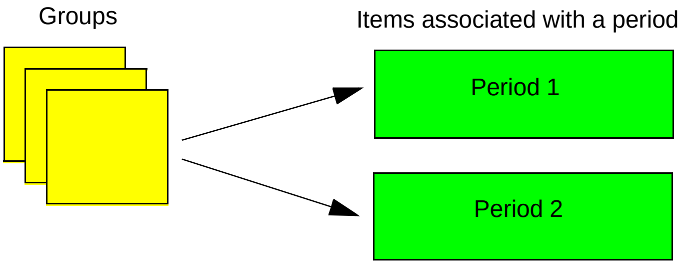
Communication medium	No. of elements consistent with one another (dep. on type)			
	bits	words	Double and floating words	Characters (string)
XWAY/TCP-IP	1960	122	61	244
XWAY/TCP-IP and ETHWAY service 1K	8104	506	253	1012
ETHWAY	1960	122	61	244
FIPWAY	928	58	29	116
ISAWAY	1808	113	56	226
UNI-TELWAY (terminal port)	928	58	29	116
MODBUS Located variables	800	100	50	-
MODBUS Unlocated variables Concept 2.5 and above	1	1	1	-
MODBUS Unlocated variables Concept 2.2 and below	1	1	1	-
MODBUS TCP-IP on Premium	800	100	50	-
ETHWAY/FIPWAY/ UNITELWAY on Series 7	496	57	28	-
ETHWAY on S1000	8128	508	254	1016

Use of groups

Description

If you divide the items into different groups, it can have an effect on the construction of network requests. For each device, the items are separated into independent sets if necessary. However, the sets will not be determined by the groups themselves but by the update periods of the groups.

Illustration:



In fact:

- there is no difference between declaring the items in different groups with the same update period, and declaring the same items in a single group with the same period.
- when groups of different periods are declared, the number of network requests needed to read the items will almost certainly be higher. On the other hand, the frames are sent at more appropriate frequencies, which can lead to a significant decrease in the overall load on devices and the network.

Optimizing requests

Description

The optimization is carried out individually for each set of items corresponding to a device and a frequency.
Optimization algorithms follow two stages:

- **Compacting**: grouping in tables of items of the same type which have similar addresses. For the write, this regrouping is only carried out if the items are strictly consecutive. The regrouping criteria consists of evaluating whether it is more worthwhile to send read/write orders of the two items individually or of the table obtained. From the original items you obtain a list of elements to send to the PLC to read or write. Compacting is also applied for non-located data if the version of Concept used is 2.5 or higher. On Series 7 type PLCs, compacting is not carried out for unitary bits. For tables of bits, it is only carried out if their number is multiplied by 8.
- **Concatenation**: construction of requests by optimizing the possibilities offered by the protocol. OFS automatically adjusts the size of requests to the maximum that is admissible:

Size of the requests in bytes according to the protocol:

Network	Maximum size in bytes
XWAY TCP-IP	256
XWAY TCP-IP service 1K	1024
ETHWAY	128
ETHWAY service 1K	1024
FIPWAY	128
ISAWAY	238
UNI-TELWAY	128
All media on Series 7	128
ETHWAY on S1000	1024
MODBUS	256

UNITE V2 Protocol (Premium, Micro): the read request enables you to group together several different read orders. OFS takes charge of filling each request by inserting as many elements as possible, whatever their type and their address. Optimization is therefore very efficient.
On the other hand, the write request only allows you to send a single write order on a single element. Therefore only compacting is performed.
UNITE V1Protocol (Series 7, S1000): the read and write requests only work on a single element. Only compacting is performed. The read performance on old ranges is therefore a lot lower than Premium and Micro PLCs.

Modbus Protocol (Quantum, Premium): same principle as for UNITE V1. However, there are fewer types of basic data and the possibilities for compacting are therefore greater.

Modbus Protocol for non-located data (Quantum): the read and write requests enable you to group several different elements. If the version of Concept used is equal to or higher than 2.5, the optimization that combines compacting and concatenation is very efficient.

Write of Concept structure type variables

Description

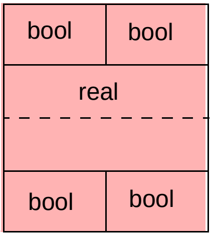
Concept gives you the opportunity to build data structures, made up of members of different types.

Unitary write of bits: it is important to note that when the fields of bit or byte type are declared in the structure, they are not compacted. In fact, each of these fields is the object of a write request. Example: a structure made up of 2 bits and three consecutive words would give rise to 3 requests if the fields are written unitarily. Please note however that the write of the structure in its entirety would give rise to a single request.

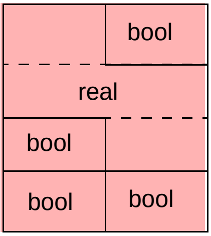
Write of non-aligned fields. When the members are not aligned on the 16 bit boundaries, the write of one of these members, which cannot be carried out with a single request is made forbidden.

Illustration:

Aligned structure:



Non-aligned structure:



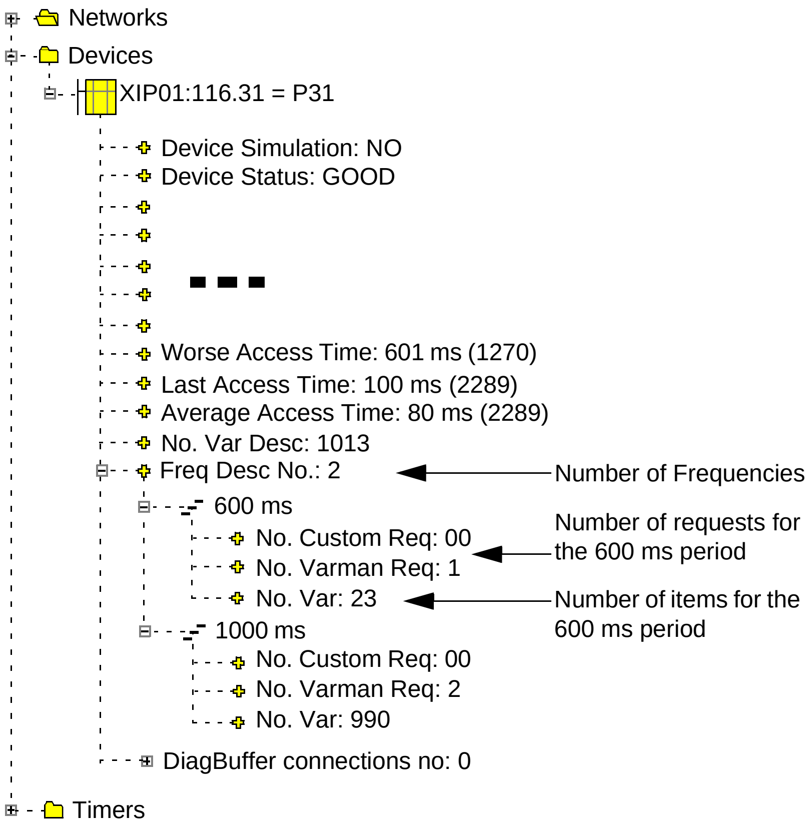
The real type member is considered as being stored on 3 addresses. The write would require 2 access bytes and 1 access word. When that is possible, it is preferable to build structures while taking into account the alignment criteria. **Important**: the write of a complete structure, including when the members inside are not aligned remains possible.

Display of the number of requests generated by OFS

Description

In "diagnostic mode" or "extended diagnostic" the server supplies the list of active frequencies corresponding to the different frequencies of the groups declared and for each one, it provides the number of items declared and the number of network requests generated.

To do this, open the "Netman XWAY" or "netman Modbus" window.



```
Networks
Devices
  XIP01:116.31 = P31
    Device Simulation: NO
    Device Status: GOOD

    ...

    Worse Access Time: 601 ms (1270)
    Last Access Time: 100 ms (2289)
    Average Access Time: 80 ms (2289)
    No. Var Desc: 1013
    Freq Desc No.: 2
      600 ms
        No. Custom Req: 00
        No. Varman Req: 1
        No. Var: 23
      1000 ms
        No. Custom Req: 00
        No. Varman Req: 2
        No. Var: 990
    DiagBuffer connections no: 0
Timers
```

Number of Frequencies
Number of requests for the 600 ms period
Number of items for the 600 ms period

8.3 Dynamic Performance

At a Glance

Aim of this Section

This section introduces the dynamic performances.

What's in this section?

This section contains the following topics:

Test Configuration

Description

- **XWAY**: a Premium TSX57453 PLC equipped with an ETY110 card, a FPP20 (FIPWAY) card in the UC slot. The application is in periodic mode (allows activation of service 1K), with a scan time of 10 ms. It should be noted that the PLC model can have a non negligible effect on the performance, since the number of UNITE requests, which can be accepted in parallel varies greatly depending on the model. In this case, the maximum is 16 per cycle. For further details, please see the Premium catalog, chapter 8. For ISAWAY specifically , a TPCX 57353 card will be used.
- **Modbus**: a Quantum PLC equiped with a 140CPU53414A CPU and an Ethernet 150NOE77110 card (Modbus TCPIP).

A PC of PIII 450Mhz spec, 256 Mb of RAM, Windows 2000 SP1 is used for OFS.

- Adjusting the OFS server:
  - MbpRcvRate = 10 ms,
  - XwayRcvRate = 20 ms,
  - Min Update Rate = 500 ms.
  - Group Update Rate = 1000ms

Fan-out

Description

The fan-out corresponds, in a given situation, to the measurement of the maximum number of items that can be declared without the communication being altered. Once this measurement has been exceeded, the OFS server continues to function but conditions deteriorate: communication errors, update time of the group not being respected etc.

The fan-out is evaluated by measuring an increasing number of PLCs until the threshold is reached where the maximum number of requests will not increase any further. The limiting factor is therefore no longer linked to the PLCs; so the fan-out is truly the fan-out of the OFS product on a given network.

These measures have been implemented with the "check device" option checked

Note: The PC used has a considerable impact on the results. A more powerful PC would give even better results.

Read:
The table indicates the number of network requests obtained, a more significant figure than the details of the items declared, which are unlikely to match the user's requirements ! For information only, however, we will give the maximum number of words or bits to which it corresponds.
Illustration:

Communication medium	Number of network requests	Bits	words
XWAY on TCPIP	92	90988	11408
XWAY on TCPIP service 1K	22	91476	11176
ETHWAY	54	53406	6696
FIPWAY	34	16218	2040
ISAWAY	179	164143	20585
UNI-TELWAY (terminal port)	8	3816	480
Modbus TCPIP	20 (1 channel) 38 (2 channels) 88 (5 channels *)	40000 76000 176000	2500 4750 11000
Modbus plus	20 (1 channel *) 40 (2 channels *)	40000 80000	2500 5000
Modbus RTU	2	4000	250

* adjustment of the number of channels (See Setting the alias properties, p. 44). The figures obtained are strongly dependent on the number of channels allocated and the frame time out and device time out values. With the test values (1000/5000), on Modbus TCPIP, there is no point going beyond 2 as the results do not change any more for higher values. With a couple of values 2000/10000, you can have up to 5 channels and you obtain a fan-out of 88 requests.

Write:

As the quantity of data that can be written during a single write order is unlimited (the requests are put into sequence when the maximum exchange capacity is reached), we will measure here the quantity of data that can be written within a second.

Illustration:

Communication medium	Number of network requests	Bits	words
XWAY on TCPIP	18	35136	2196
XWAY on TCPIP service 1K	7	56672	3542
ETHWAY	20	39040	2440
ETHWAY Service 1K	25	202400	12650
FIPWAY	28	25984	1624
ISAWAY	282	509856	31866
UNI-TELWAY (terminal port)	8	7424	464
Modbus TCPIP	20	16000	2000
Modbus plus	20	16000	2000
Modbus RTU	3	2400	300

Update time

Description

We will measure the average time (in milliseconds) needed to update the following set of consecutive items:

- 128 bits,
- 128 words,
- 64 bits and 64 words,

Update time:

Communication medium	128 bits	128 words	64 bits and 64 words
XWAY on TCPIP	150	160	150
XWAY on TCPIP service 1K	150	160	150
ETHWAY	140	150	140
FIPWAY	160	270	200
UNI-TELWAY (terminal port)	160	400	300
Modbus TCPIP	50	100	100
Modbus +	50	90	100
Modbus RTU	160	580	440
ETHWAY on Series 7	150	150	180
ETHWAY on S1000	120	120	120
UNI-TELWAY on Series 7	250	680	520
FIPWAY on Series 7	210	280	300

Time elapsed between sending and receiving a request

At a Glance

On each of the networks, a write order of 2 words is sent. A series of measurements of the elapsed time recorded between sending and receiving the request is carried out.

Typical situation : a reasonable number of items declared (4000 words on Ethernet and ISAWAY networks, 1500 in FIPWAY, 500 on the series networks, 3000 in Modbus+, with half bits, half words in all cases), polling activated, average network load. The write request is implemented quickly,

The test is carried out in three situations:

- optimum situation : no item declared, polling not activated, no network load. The write request is implemented without delay,
- typical situation: a reasonable number of items declared (4000 words on Ethernet and ISAWAY networks, 1500 in FIPWAY, 500 on the series networks, 3000 in Modbus+, with half bits, half words in all cases), polling activated, average network load. The write request is implemented quickly,
- deteriorated situation : the server is saturated in polling, the network charge is considerable.

As the spread of the measurements obtained is considerable, the results are presented in the form of a min-max range.

The "Frame time out" parameter is set to 10000ms, in order to guarantee that the write does not fail in the deteriorated situation.

Illustration:

Network	**Time in ms**		
	Optimal	**Typical**	**Deteriorated**
XWAY TCPIP	100 - 250	110 - 500	up to the Frame time out value Beyond that, failure
ETHWAY	30 - 200	30 - 460	
FIPWAY	100 - 250	200 - 600	
UNITELWAY (terminal port)	140 - 300	150 - 800	
ISAWAY	60 - 180	60 - 210	
Modbus TCPIP	100	110 - 150	
Modbus +	70 - 110	80 - 150	
Modbus RTU	150	150 - 250	
ETHWAY on Series 7	70 - 210	100 - 1000	
ETHWAY on S1000	80 - 180	50 - 200	

.

Multi-PLC write

Description

In order to illustrate the efficiency of implementing parallelism on the transmission of write requests to different PLCs, we will compare the total time needed to implement a write order of 2 words per PLC, on a single PLC and then 10 different ones. It was only carried out on XWAY TCPIP and FIPWAY, under optimum conditions (see description above).
Illustration:

Network	Time in ms (min - max)	
	1 PLC	10 PLCs
XWAY TCPIP	40 - 170	60 - 230
FIPWAY	140 - 270	300 - 600

OPC Interface

Description

- the values of variables are always different (worst case),
- the network interface is idle,

The remote client is a Pentium 133 with NT4.0 SP3.

The test is considered OK if the notification rate measured by the testing client is the expected rate and if the number of items notified is the number expected.

Performance Table:

OPC client	**Update rate (ms)**	**Maximum number of items**	**Number of items notified per second**
Local	200	20000	100000
Remote	200	2000	10000
Remote	800	10000	12500
Remote	1000	14000	14000

Appendices

9

At a Glance

Aim of this Chapter

The aim of this chapter is to introduce the appendices for this book.

What's in this chapter?

This chapter contains the following Sections:

9.1 Compatibility of the OFS server

OFS server compatibility

Definition

OFS is compatible with OPC 1.0A and 2.0.

In particular, the OFS server accepts SYNCHRONOUS group mono-request, mono-PLC notions. Syntactically, the name of a synchronous group starts with "$" (see The Different Groups of Items, p. 109).

The OFS server is also compatible with the notion of a SYSTEM group dedicated to the PLC driver and address couple:
The system groups refer to a given device and are used to manage specific items related to that device. System groups are distinguished from user groups by their name, which must include the prefix "_SYS=".
A system group only contains the following specific items starting with "#":

- #PLCStatus" for managing the PLC operating mode,
- #TimeOut" for managing a communication medium time-out,
- #NbrRequest" to find out the number of requests sent to this device

The OFS server shares certain error names with eOLEs. These names/codes start with the prefix "EOL_", for eOLEs.
Specific items and system groups CANNOT be activated. Notification and asynchronous read/write are not possible.

9.2 Exhaustive list of error codes

At a Glance

Aim of this Section

The aim of this section is to present an exhaustive list of error codes in the form of a table.

What's in this section?

This section contains the following topics:

Main errors defined by OLE

List of error codes

The table below summarizes the list of main OLE errors:

HRESULT	NAME	COMMENT
00000000	S_OK	The primitive has been executed (S)uccessfully
00000001	S_FALSE	(S)uccess status for the primitive, but with at least one error on an item
80004005	E_FAIL	Primitive failed
80010105	RPC_E_SERVERFAULT	The server has released an exception
8007000E	E_OUTOFMEMORY	Insufficient memory
800706BA	RPC_C_S_SERVER_UNAVAILABLE	The Concept project is accessed from a remote machine, which cannot be located

Errors defined by the OPC standard

List of error codes

The table below summarizes the list of OPC errors:

HRESULT	NAME	COMMENT
C0040001	OPC_E_INVALIDHANDLE	An invalid handle was passed
C0040002	OPC_E_DUPLICATE	A duplicate parameter was passed where it is not authorized
C0040003	OPC_E_UNKNOWNLCID	The server does not support the local LCID language code indicated
C0040004	OPC_E_BADTYPE	The server cannot convert the type of data passed or requested with the canonical type
C0040005	OPC_E_PUBLIC	The requested operation cannot be done in a public group
C0040006	OPC_E_BADRIGHTS	The item's access rights do not permit the operation
C0040007	OPC_E_UNKNOWNITEMID	The definition of the item does not exist in the server's address space
C0040008	OPC_E_INVALIDITEMID	The definition of the item does not match the server's syntax
C0040009	OPC_E_INVALIDFILTER	The filter string is not valid
C004000A	OPC_E_UNKNOWNPATH	The item path is unknown to the server
C004000B	OPC_E_RANGE	The value to be written was outside the range
C004000C	OPC_E_DUPLICATE_NAME	A group with the same name already exists in the server
0004000D	OPC_S_UNSUPPORTEDRATE	The server does not support the period requested, however it will use the closest usable one
0004000E	OPC_S_CLAMP	A value to be written was accepted but has been clamped
0004000F	OPC_S_INUSE	The operation cannot be performed because the object is still referenced
00040010	OPC_E_INVALIDCONFIGFILE	The server's configuration file is invalid (problem with format)
00040011	OPC_E_NOTFOUND	The server cannot find the object requested

HRESULT	NAME	COMMENT
00040012	OPC_E_INVALID_PID	The server does not recognize the item property identifier passed to a parameter

Errors defined by the OFS server

List of error codes

The table below shows the error codes defined by OFS

	HRESULT	NAME	COMMENT
1	EFEF61A8	OFS_E_NO_DEFAULT_DIRECTORY	No default directory for the symbols table
2	EFEF61A9	OFS_E_UNABLE_TO_RELOAD_ST	Unable to reload the symbol table
3	EFEF61AA	OFS_E_UNSUPPORTED_ST_TYPE	Unsupported symbol table type
4	EFEF61AB	OFS_S_ST_NOT_CURRENTLY_USED	The symbols table is not currently in use
5	EFEF61AC	OFS_E_ALIAS_ALREADY_DEFINED	Alias already defined
6	EFEF61AD	OFS_E_ALIAS_OVERFLOW	Maximum number of aliases reached
7	EFEF61AE	OFS_E_ALIAS_NOT_FOUND	Alias not found
8	EFEF61AF	OFS_E_UNABLE_TO_LOAD_ST	Impossible to load the symbol table
9	EFEF61B0	OFS_E_CF_FORMAT_NOT_SUPPORTED	Clipboard format not supported
10	EFEF61B1	OFS_E_CANNOT_SET_SYSTEM_GROUP_STATE	Unsupported set state of a system group
11	EFEF61B2	OFS_E_DUPLICATE_CLIENT_HANDLE	An item with the same client handle already exists
12	EFEF61B3	OFS_E_BAD_TYPE_REQUESTED	The type requested cannot be instanced or is incompatible with the canonical type
13	EFEF61B4	OFS_E_READ_FAILURE	Read failure
14	EFEF61B5	OFS_E_SET_ITEM_STATE_SYNCHRO	Changes of item status in synchronous group forbidden
15	EFEF61B6	OFS_E_SAFEARRAY_CREATION_FAILURE	The creation of a SAFEARRAY has failed
16	EFEF61B7	OFS_E_SAFEARRAY_EMPTY	SAFEARRAY empty
17	EFEF61B8	OFS_E_ASYNC_READ_REJECTED	Asynchronous read rejected
18	EFEF61B9	OFS_E_THREAD_CREATION_FAILURE	Thread creation failure
19	EFEF61BA	OFS_E_NO_ACTIVE_ITEM	There is no active item
20	EFEF61BB	OFS_E_GROUP_NOT_ACTIVE	The group is not active
21	EFEF61BC	OFS_E_TRANSACTION_NOT_FOUND	The transaction identifier was not found
22	EFEF61BD	OFS_E_BAD_GROUP_TYPE_FOR_ITEM	Cannot create the specific item in this type of group
23	EFEF61BE	OFS_E_ST_NOT_DEFINED	Symbols table not defined
24	EFEF61BF	OFS_E_ASYNC_WRITE_REJECTED	Asynchronous write rejected
25	EFEF61C0	OFS_E_MODBUS_MBT_RCV_TO	Time-out on the Modbus TCP-IP network

	HRESULT	NAME	COMMENT
26	EFEF61C1	OFS_E_MODBUS_MBT_RCV_ERROR	Reception error on the Modbus TCP-IP network
27	EFEF61C2	OFS_E_MODBUS_MBT_SEND_ERROR	Transmission error on the Modbus TCP-IP network
28	EFEF61C3	OFS_E_MODBUS_NCB_WAIT_ERROR	Receive error on Modbus+ network
29	EFEF61C4	OFS_E_MODBUS_NCB_SEND_ERROR	Transmission error on Modbus+ network
30	EFEF61C5	OFS_E_MODBUS_NCB_BAD_ANSWER	Bad Modbus response (object out of configuration?)
31	EFEF61C6	OFS_E_CONCEPT_IEC_NOT_SUPPORTED	Runtime IEC Concept not supported by this device
32	EFEF61C7	OFS_E_MODBUS_DEVICE_NOT_CONSISTENT	Consistency error between Modbus Device and Symbol file
33	EFEF61C8	OFS_E_MODBUS_DETECTION_DEVICE_ERROR	Modbus device not found or not accessible
34	EFEF61C9	OFS_E_MODBUS_TO_CHECK_CNX	Time-out while detecting Modbus device
35	EFEF61CA	OFS_E_MODBUS_DRIVER_ID_ERROR	Driver identification error
36	EFEF61CB	OFS_E_NETLIB_OPEN_ERROR	Error opening the Netlib DLL
37	EFEF61CC	OFS_E_NETLIB_LOAD_ERROR	Error loading Netlib DLL
38	EFEF61CD	OFS_E_MODBUS_PLUS_DRIVER_NOT_FOUND	Modbus+ driver not found (not installed ?)
39	EFEF61CE	OFS_E_NCB_OPEN_ERROR	MODBUS +: error opening a DM or PM path (all paths used?)
40	EFEF61CF	OFS_E_NCB_REOPEN_ERROR	MODBUS +: error reopening a DM or PM path (all paths used?)
41	EFEF61D0	OFS_E_MODBUS_MBT_CONNECT1	Internal server error
42	EFEF61D1	OFS_E_MODBUS_MBT_BAD_SOCKET	Internal server error
43	EFEF61D2	OFS_E_MODBUS_MBT_SETSOCKOPT1	Internal server error
44	EFEF61D3	OFS_E_MODBUS_MBT_SETSOCKOPT2	Internal server error
45	EFEF61D4	OFS_E_MODBUS_MBT_SETSOCKOPT3	Internal server error
46	EFEF61D5	OFS_E_MODBUS_MBT_DNS_FAILED	Impossible DNS/IP address conversion (Name not found?)
47	EFEF61D6	OFS_E_MODBUS_MBT_BAD_HOST	Internal server error
48	EFEF61D7	OFS_E_MODBUS_MBT_BAD_INET	Error in IP address definition
49	EFEF61D8	OFS_E_MODBUS_MBT_BAD_SOCKSTATE1	Internal server error
50	EFEF61D9	OFS_E_MODBUS_MBT_BAD_SEND	Internal server error
51	EFEF61DA	OFS_E_MODBUS_MBT_BAD_STATE	Internal server error

	HRESULT	NAME	COMMENT
52	EFEF61DB	OFS_E_MODBUS_MBT_OVERFLOW	Transmission queue of Modbus / TCP-IP requests full
53	EFEF61DC	OFS_E_MODBUS_IOCTL1	Internal server error
54	EFEF61DD	OFS_E_MODBUS_IOCTL2	Internal server error
55	EFEF61DE	OFS_E_MODBUS_IOCTL3	Internal server error
56	EFEF61DF	OFS_E_MODBUS_IOCTL4	Internal server error
57	EFEF61E0	OFS_E_MODBUS_IOCTL5	Internal server error
58	EFEF61E1	OFS_E_MODBUS_IOCTL6	Internal server error
59	EFEF61E2	OFS_E_MODBUS_IOCTL7	Internal server error
60	EFEF61E3	OFS_E_XWAY_IOCTL1	Internal server error
61	EFEF61E4	OFS_E_XWAY_IOCTL2	Internal server error
62	EFEF61E5	OFS_E_XWAY_IOCTL3	Internal server error
63	EFEF61E6	OFS_E_XWAY_IOCTL4	Internal server error
64	EFEF61E7	OFS_E_XWAY_IOCTL5	Internal server error
65	EFEF61E8	OFS_E_XWAY_IOCTL6	Internal server error
66	EFEF61E9	OFS_E_XWAY_IOCTL7	Internal server error
67	EFEF61EA	OFS_E_XWAY_IOCTL8	Internal server error
68	EFEF61EB	OFS_E_XWAY_IOCTL9	Internal server error
69	EFEF61EC	OFS_E_XWAY_IOCTL10	Internal server error
70	EFEF61ED	OFS_E_XWAY_IOCTL11	Internal server error
71	EFEF61EE	OFS_E_XWAY_IOCTL12	Internal server error
72	EFEF61EF	OFS_E_XWAY_IOCTL13	Internal server error
73	EFEF61F0	OFS_E_XWAY_IOCTL14	Internal server error
74	EFEF61F1	OFS_E_XWAY_IOCTL15	Internal server error
75	EFEF61F2	OFS_E_MODBUS_LIST1	Internal server error
76	EFEF61F3	OFS_E_MODBUS_LIST2	Internal server error
77	EFEF61F4	OFS_E_MODBUS_LIST3	Internal server error
78	EFEF61F5	OFS_E_MODBUS_LIST4	Internal server error
79	EFEF61F6	OFS_E_MODBUS_LIST5	Internal server error
80	EFEF61F7	OFS_E_MODBUS_LIST6	Internal server error
81	EFEF61F8	OFS_E_MODBUS_LIST7	Internal server error
82	EFEF61F9	OFS_E_MODBUS_LIST8	Internal server error
83	EFEF61FA	OFS_E_MODBUS_LIST9	Internal server error
84	EFEF61FB	OFS_E_MODBUS_LIST10	Internal server error
85	EFEF61FC	OFS_E_XWAY_LIST1	Internal server error

	HRESULT	NAME	COMMENT
86	EFEF61FD	OFS_E_XWAY_LIST2	Internal server error
87	EFEF61FE	OFS_E_XWAY_LIST3	Internal server error
88	EFEF61FF	OFS_E_XWAY_LIST4	Internal server error
89	EFEF6200	OFS_E_XWAY_LIST5	Internal server error
90	EFEF6201	OFS_E_XWAY_LIST6	Internal server error
91	EFEF6202	OFS_E_XWAY_LIST7	Internal server error
92	EFEF6203	OFS_E_XWAY_LIST8	Internal server error
93	EFEF6204	OFS_E_XWAY_LIST9	Internal server error
94	EFEF6205	OFS_E_MODBUS_BAD_READCPU	Internal server error
95	EFEF6206	OFS_E_MODBUS_CMD_FAILURE	This device has already been reserved by another tool (Concept?)
96	EFEF6207	OFS_E_MODBUS_LOGIN_FAILURE	Internal server error
97	EFEF6208	OFS_E_MODBUS_LOGOUT_FAILURE	Internal server error
98	EFEF6209	OFS_E_XWAY_NET_OPEN	Internal server error
99	EFEF620A	OFS_E_XWAY_NET_ATTACH	Internal server error
100	EFEF620B	OFS_E_XWAY_NET_LOCALINFO	Internal server error
101	EFEF620C	OFS_E_XWAY_TEST_PDU_SIZE	Internal server error
102	EFEF620D	OFS_E_XWAY_DRIVER_ID_ERROR	Internal server error
103	EFEF620E	OFS_E_XWAY_CHECK_DEVICE	Internal server error
104	EFEF620F	OFS_E_XWAY_BAD_READCPU	Internal server error
105	EFEF6210	OFS_E_XWAY_READCPU_FAILURE	Internal server error
106	EFEF6211	OFS_E_XWAY_WRITECPU_FAILURE	Internal server error
107	EFEF6212	OFS_E_XWAY_BAD_CMD	Internal server error
108	EFEF6213	OFS_E_BAD_PATH_SPECIFIC_ITEM	Internal server error
109	EFEF6214	OFS_E_DIAG_CNX_OPEN	Internal server error
110	EFEF6215	OFS_E_NO_DIAG_CNX	Internal server error
111	EFEF6216	OFS_E_SPECIFIC1	Internal server error
112	EFEF6217	OFS_E_SPECIFIC2	Internal server error
113	EFEF6218	OFS_E_DRVMDB1	Internal server error
114	EFEF6219	OFS_E_REQMDB1	Internal server error
115	EFEF621A	OFS_E_REQMDB2	Internal server error
116	EFEF621B	OFS_E_MISCXWAY1	Internal server error
117	EFEF621C	OFS_E_TMDB1	Internal server error
118	EFEF621D	OFS_E_TMDB2	Internal server error
119	EFEF621E	OFS_E_TMDB3	Internal server error

	HRESULT	NAME	COMMENT
120	EFEF621F	OFS_E_TMDB4	Internal server error
121	EFEF6220	OFS_E_TMDB5	Internal server error
122	EFEF6221	OFS_E_TMDB6	Internal server error
123	EFEF6222	OFS_E_TMDB7	Internal server error
124	EFEF6223	OFS_E_TMDB8	Internal server error
125	EFEF6224	OFS_E_TMDB9	Internal server error
126	EFEF6225	OFS_E_TMDB10	Internal server error
127	EFEF6226	OFS_E_TMDB11	Internal server error
128	EFEF6227	OFS_E_MISCXWAY2	Internal server error
129	EFEF6228	OFS_E_NETLOCAL1	Internal server error
130	EFEF6229	OFS_E_T_VM1	Internal server error
131	EFEF622A	OFS_E_T_VM2	Internal server error
132	EFEF622B	OFS_E_T_VM3	Internal server error
133	EFEF622C	OFS_E_T_VM4	Internal server error
134	EFEF622D	OFS_E_T_VM5	Internal server error
135	EFEF622E	OFS_E_T_VM6	Internal server error
136	EFEF622F	OFS_E_T_VM7	Internal server error
137	EFEF6230	OFS_E_T_VM8	Internal server error
138	EFEF6231	OFS_E_T_VM9	Internal server error
139	EFEF6232	OFS_E_T_VM10	Internal server error
140	EFEF6233	OFS_E_T_VM11	Internal server error
141	EFEF6234	OFS_E_READ_SYNC_FAILURE	Internal server error
142	EFEF6235	OFS_E_SRVM1	Internal server error
143	EFEF6236	OFS_E_SRVM2	Internal server error
144	EFEF6237	OFS_E_SRVM3	Internal server error
145	EFEF6238	OFS_E_RW1	Internal server error
146	EFEF6239	OFS_E_BAD_SYS_GROUP	System Group definition error
147	EFEF623A	OFS_E_BAD_ITEM_NUMBER	Internal server error
148	EFEF623B	OFS_E_BAD_POSTFIX	Incorrect postfix for this item
149	EFEF623C	OFS_E_DUPL_POSTFIX	Postfix redefinition error for this item
150	EFEF623D	OFS_E_DUPL_ARRAY_LEN	Table length redefinition error for this item
151	EFEF623E	OFS_E_BAD_RANGE_ID	Range_id error for a Modbus address
152	EFEF623F	OFS_E_WRONG_DIGIT_NUMBER	Wrong number of digits for the Modbus addresses (6 digits in any case)
153	EFEF6240	OFS_E_WRONG_ARRAY_LEN	Table Length Definition Error

	HRESULT	NAME	COMMENT
154	EFEF6241	OFS_E_NO_ARRAY_OF_ARRAY	Array of Array not supported (Redefinition Error ?)
155	EFEF6242	OFS_E_MODBUS_UNSUPPORTED_DATA_TYPE	Type of MODBUS data not supported
156	EFEF6243	OFS_E_NO_MODBUS_REQ_GEN_ERROR	Internal server error
157	EFEF6244	OFS_E_NO_MODBUS_REQ_GENERATED	Internal server error
158	EFEF6245	OFS_E_BAD_MODBUS_ITEM_DEF	MODBUS item definition error
159	EFEF6246	OFS_E_WRITESYNC_FAILURE	Write failure
160	EFEF6247	OFS_E_BAD_IEC_WRITE_ANSWER	Write request rejected by the Quantum Runtime IEC
161	EFEF6248	OFS_E_NO_REQ_TO_SEND	Internal server error
162	EFEF6249	OFS_E_MODBUS_SEND_FAILURE	Internal server error
163	EFEF624A	OFS_E_REMOVE_ERR1	Internal server error
164	EFEF624B	OFS_E_REMOVE_ERR2	Internal server error
165	EFEF624C	OFS_E_DEACTIVE_ERR	Internal server error
166	EFEF624D	OFS_E_MOD_RATE_ERR	Internal server error
167	EFEF624E	OFS_E_MODBUS_THREAD_FAILURE	Internal server error
168	EFEF624F	OFS_E_MODBUS_READ_FAILURE	Modbus read failure
169	EFEF6250	OFS_E_XWAY_BAD_ITEM_DEF	XWAY item definition error
170	EFEF6251	OFS_E_XWAY_UNSUPPORTED_DATA_TYPE	Type of XWAY data not supported
171	EFEF6252	OFS_E_NO_XWAY_REQ_GEN_ERROR	Internal server error
172	EFEF6253	OFS_E_NO_XWAY_REQ_GENERATED	Internal server error
173	EFEF6254	OFS_E_XWAY_READ_FAILURE	XWAY read failure
174	EFEF6255	OFS_E_XWAY_BAD_WRITE_ANSWER1	Internal server error
175	EFEF6256	OFS_E_XWAY_BAD_WRITE_ANSWER2	Internal server error
176	EFEF6257	OFS_E_MODBUS_MBP_RCV_TO	Internal server error
177	EFEF6258	OFS_E_MODBUS_MBP_SEND_TO	Internal server error
178	EFEF6259	OFS_E_MODBUS_NCB_OVERFLOW	Internal server error
179	EFEF625A	OFS_E_MODBUS_MBP_REQ_TOO_OLD	Internal server error
180	EFEF625B	OFS_E_MODBUS_MBP_SEND_ERROR	Internal server error
181	EFEF625C	OFS_E_MODBUS_RCV_TO	Internal server error
182	EFEF625D	OFS_E_MODBUS_MBT_NO_CONNECT	Internal server error
183	EFEF625E	OFS_E_MODBUS_MBT_NO_CNX	Internal server error
184	EFEF625F	OFS_E_MODBUS_MBT_BAD_ANSWER	Internal server error
185	EFEF6260	OFS_E_MODBUS_MBT_REQ_TOO_OLD	Internal server error

	HRESULT	NAME	COMMENT
186	EFEF6261	OFS_E_XWAY_EXEC_READ_FAILURE	Internal server error
187	EFEE4971	EOL_E_INVALID_GROUP_NAME	Invalid group name
188	6FEE4972	EOL_S_INVALID_APPLI_NAME	The name of the application in the PLC is different to that registered in the symbols file
189	6FEE4973	EOL_S_INVALID_APPLI_VERSION	The evolution number of application in PLC differs from the number registered in symbol file
190	EFEE4974	EOL_E_INVALID_ARG1	Argument 1 is not valid
191	EFEE4975	EOL_E_INVALID_ARG2	Argument 2 is not valid
192	EFEE4976	EOL_E_INVALID_ARG3	Argument 3 is not valid
193	EFEE4977	EOL_E_INVALID_ARG4	Argument 4 is not valid
194	EFEE4978	EOL_E_INVALID_ARG5	Argument 5 is not valid
195	EFEE4979	EOL_E_INVALID_ARG6	Argument 6 is not valid
196	EFEE497A	EOL_E_INVALID_ARG7	Argument 7 is not valid
197	EFEE497B	EOL_E_INVALID_ARG8	Argument 8 is not valid
198	EFEE497C	EOL_E_INVALID_ARG9	Argument 9 is not valid
199	EFEE497D	EOL_E_INVALID_ARG10	Argument 10 is not valid
200	EFEE497E	EOL_E_INVALID_ARG11	Argument 11 is not valid
201	EFEE497F	EOL_E_INVALID_ARG12	Argument 12 is not valid
202	EFEE4980	EOL_E_INVALID_TYPE	The type of variable is not valid
203	EFEE4981	EOL_E_MISSING_NEUTRAL_FILE	The neutral symbol definition file is missing
204	EFEE4982	EOL_E_MISSING_OPCerror_DLL	The OPCerror.dll resource file cannot be loaded
205	EFEE4983	EOL_E_MISSING_EOLESerr_DLL	The EOLESerr.dll resource file cannot be loaded
206	EFEE4984	EOL_E_RESOURCE_NOT_FOUND	The resources DLL cannot be found
207	EFEE4985	EOL_E_MSG_NOT_FOUND	The message cannot be found
208	EFEE4986	EOL_E_EXPECTED_ARRAY_VALUES	A TABLE of values is expected
209	EFEE4987	EOL_E_EXPECTED_SINGLE_VALUE	A SINGLE value is expected and not a table
210	EFEE4988	EOL_E_EXPECTED_DIM1	A table of values is expected with only ONE dimension
211	EFEE4989	EOL_E_UNEXPECTED_BOUNDS	The table limits are invalid
212	EFEE498A	EOL_E_INVALID_ITEMS_NBR	The number of items is incorrect

	HRESULT	NAME	COMMENT
213	EFEE498B	EOL_E_INVALID_PATH	The driver or PLC address is different from previous items
214	EFEE498C	EOL_E_ERR16bit_OUTofRANGE	The 16-bit error code is out of range
215	EFEE498D	EOL_E_ONLY_ONE_REQUEST	Too many items: this group should be written on A SINGLE PLC request
216	EFEE498E	EOL_E_FEATURE_DISABLED	This feature has been disabled by the administrator
217	EFEE498F	EOL_E_GROUP_DELETED	This group has been logically deleted
218	EFEE4990	EOL_E_INVALID_COMMAND	Use 1 to upload from the PLC to the PC and 2 for the reverse process
219	EFE913EE	EOL_E_ILLEGAL_CHARACTER	Syntax error: illegal character
220	EFE913EF	EOL_E_INDEX_NUMBER_ERROR	Syntax error: the object has too great an index
221	EFE913F0	EOL_E_INVALID_SYNTAX	Syntax error: the syntax is incorrect
222	EFE913F1	EOL_E_LEXEM_TOO_LARGE	Syntax error: the name of the object is too long
223	EFE913F2	EOL_E_INVALID_OBJECT	Syntax error: the object is incorrect
224	EFE913F3	EOL_E_KEYWORD_ERROR	Syntax error: word reserved
225	EFE913F4	EOL_E_SYNTAX_ERROR_2	Syntax error
226	EFE913F5	EOL_E_INVALID_END_OF_PHRASE	Syntax error: The end of the phrase is incorrect
227	EFE913F6	EOL_E_SYNTAX_ERROR	Syntax error
228	EFE913F7	EOL_E_INCORRECT_SYMBOL	Syntax error: the symbol is incorrect
229	EFE913F8	EOL_E_INCORRECT_INDEX_SYMBOL	Syntax error: the index symbol is incorrect
230	EFE913F9	EOL_E_INCORRECT_BIT_INDEX_SYMBOL	Syntax error: the index symbol of the extracted bit is incorrect
231	EFE913FA	EOL_E_UNDEFINED_INDEX_SYMBOL	Syntax error: the index symbol is not defined
232	EFE913FB	EOL_E_INVALID_SYMBOL	Syntax error: the symbol is not defined
233	EFE913FC	EOL_E_OBJECT_NOT_SYNTHESIZED	Synthesis error: the object cannot be synthesized
234	EFE913FD	EOL_E_INVALID_ARRAY_SIZE	Syntax error: incorrect table length
235	EFE913FE	EOL_E_INCORRECT_PATH_SIZE	Syntax error: the path is too large
236	EFE913FF	EOL_E_OBJECT_FORMAT_TOO_LARGE	Syntax error: the format of the object is too large
237	EFE91400	EOL_E_INCORRECT_OBJECT_FORMAT	Syntax error: the format of the object is incorrect

	HRESULT	NAME	COMMENT
238	EFE91401	EOL_E_UNSIGNED_OBJECT_ERROR	Syntax error: the object has no sign
239	EFE91402	EOL_E_EMPTY_EXPRESSION	Syntax error: the expression is empty
240	EFE91403	EOL_E_ILLEGAL_INDEXES	Syntax error: indexes are illegal
241	EFE91404	EOL_E_ILLEGAL_ARRAYS	Syntax error: These tables are illegal
242	EFE91405	EOL_E_NO_SEPARATOR	Syntax error: no separator in front of the object
243	EFED4845	EOL_E_CODE4INIT_ERROR	CodeBase error during the CODE4INIT function
244	EFED4846	EOL_E_D4APPENDSTART_ERROR	CodeBase error during the D4APPENDSTART function
245	EFED4847	EOL_E_D4CREATE_ERROR	CodeBase error during the D4CREATE function
246	EFED4848	EOL_E_I4CREATE_ERROR	CodeBase error during the I4CREATE function
247	EFED4849	EOL_E_D4FIELD_ER	CodeBase error during the D4FIELD function
248	EFED484A	EOL_E_D4TAG_ERROR	CodeBase error during the D4TAG function
249	EFED484C	EOL_E_D4APPEND_ERROR	CodeBase error during the D4APPEND function
250	EFED484D	EOL_E_RELATE4INIT_ERROR	CodeBase error during the RELATE4INIT function
251	EFED484E	EOL_E_RELATE4QUERYSET_ERROR	CodeBase error during the RELATE4QUERY_SET function
252	EFED484F	EOL_E_RELATE4TOP_ERROR	CodeBase error during the RELATE4_TOP function
253	EFED4850	EOL_E_RELATE4SKIP_ERROR	CodeBase error during the RELATE4_SKIP function
254	EFED4851	EOL_E_INVALID_SYMBOLS_FILE	The contents of the symbols file is invalid
255	EFED4852	EOL_E_OPEN_SYMBOLS_FAILURE	Failure opening the symbols file: invalid name or path, file not found,...
256	EFED4853	EOL_E_ERRORS_IN_SYMBOLS_FILE	Errors in the symbols file: syntax error in a line, double definition of the same address or symbol
257	EFED4854	EOL_E_NO_OBJECT_FOR_SYMBOL	No object for the symbol
258	EFED4855	EOL_E_INPUT_BUFFER_TOO_LARGE	The input buffer is too large
259	EFED4856	EOL_E_OUTPUT_BUFFER_TOO_SMALL	The output buffer is too small

	HRESULT	NAME	COMMENT
260	EFE84DBD	EOL_E_REQUESTS_OPTIMIZATION_ERROR1	Internal error in requests optimization: the address settings mode of the object is not on 2 bytes
261	EFE84DBE	EOL_E_REQUESTS_OPTIMIZATION_ERROR2	Internal error in requests optimization: the address settings mode of the object is not on 5 bytes
262	EFE84DBF	EOL_E_REQUESTS_OPTIMIZATION_ERROR3	Internal error in requests optimization: invalid object addressing mode
263	EFE84DC0	EOL_E_REQUESTS_OPTIMIZATION_ERROR4	Internal error in requests optimization: unknown ROL object
264	EFE84DC1	EOL_E_REQUESTS_OPTIMIZATION_ERROR5	Internal error in requests optimization: unknown Node object
265	EFE84DC2	EOL_E_REQUESTS_OPTIMIZATION_ERROR6	Internal error in requests optimization: unknown Info object
266	EFE84DC3	EOL_E_REQUESTS_OPTIMIZATION_ERROR7	Internal error in requests optimization: ROL object number higher than the number of objects defined
267	EFE84DC4	EOL_E_REQUESTS_OPTIMIZATION_ERROR8	Internal error in requests optimization: Element object number higher than the number of objects defined
268	EFE84DC5	EOL_E_REQUESTS_OPTIMIZATION_ERROR9	Internal error in requests optimization: Info object number higher than the number of objects defined
269	EFE84DC6	EOL_E_REQUESTS_OPTIMIZATION_ERROR10	Internal error in requests optimization: option not processed
270	EFE84DC7	EOL_E_REQUESTS_OPTIMIZATION_ERROR11	Internal error in requests optimization: option planned for a future version
271	EFE84DC8	EOL_E_REQUESTS_OPTIMIZATION_ERROR12	Internal error in requests optimization: object not supported by this version
272	EFE84DC9	EOL_E_REQUESTS_OPTIMIZATION_ERROR13	Internal error in requests optimization: invalid type of access
273	EFE84DCA	EOL_E_REQUESTS_OPTIMIZATION_ERROR14	Internal error in requests optimization: incoherent parameters
274	EFE84DCB	EOL_E_REQUESTS_OPTIMIZATION_ERROR15	Internal error in requests optimization: size of request data is greater than the size of the buffer
275	EFE84DCC	EOL_E_OUT_OF_BITSARRAY_BOUNDARY	The size of the table of bits exceeds the permitted limit

	HRESULT	NAME	COMMENT
276	EFE84DCD	EOL_E_BAD_IO_TYPE	Internal error in requests optimization: Bad IO Type
277	EFE84DCE	EOL_E_BAD_ADDR_LENGTH	Internal error in requests optimization: the length of the I/O address is not managed
278	EFE84DCF	EOL_E_READ_ARRAY_FORBIDDEN	The read of this type of table is not allowed
279	EFE84DD0	EOL_E_MBI_NOT_ARRAY	This type of object should be named in table form

Lists of Diag buffer error codes

The table below shows the error codes that the Diag buffer (See Diag Buffer, p. 169) can generate:

	HRESULT	NAME	COMMENT
1	0	OFS_E_DIAG_OK	OK
2	2FE30000	OFS_E_DIAG_OK_READ_NOTHING	Read successful, no modification is recorded in the 128 byte table
3	2FE30001	OFS_E_DIAG_OK_READ_MODIFIED	Read successful, modifications recorded in the 128 byte table (the error has been acknowledged or has disappeared)
4	2FE30002	OFS_E_DIAG_OK_READ_NEW	Read successful, a new table has been created (a new error has appeared)
5	EFE30001	OFS_E_DIAG_NO_BUFFER	Diag buffer not activated
6	EFE30002	OFS_E_DIAG_BUFFER_FULL	Diag buffer is full
7	EFE30003	OFS_E_DIAG_TOO_MUCH_MMI	All possible viewers (15) are connected
8	EFE30004	OFS_E_DIAG_WRONG_MMI_ID	The viewer identifier is not valid (outside range 1 to 15)
9	EFE30005	OFS_E_DIAG_MMI_ALREADY_CONNECTED	The viewer is already connected
10	EFE30006	OFS_E_DIAG_MMI_NOT_CONNECTED	OPC client not connected
11	EFE30007	OFS_E_DIAG_WRONG_ERROR_ID	non authorized error identifier
12	EFE30008	OFS_E_DIAG_ERROR_NOT_USED	No error corresponds to this identifier
13	EFE30009	OFS_E_DIAG_WRONG_OFFSET	OFFSET error

9.3 MODBUS and XWAY Request Codes used by OFS

MODBUS and XWAY request codes used by OFS

Description

This is a list of all the request codes used by the OFS server. If your device does not support a request code, this feature will not be available. If you do not use the feature, the request code WILL NOT BE generated (you have no problem if your device does not support the request code in question).
For Modbus devices, if the request code is supported but the maximum length is not, there may be a problem.

MODBUS request codes used by OPC Factory Server:

Request code	Function name	Max. length used	OFS features using the request code
1	Read Coil Status	2000	Read of 0x items
2	Read Input Status	2000	Reading 1x items
3	Read Holding Registers	125	Read of 4x items and device detection (with 0 reg)
4	Read Input Registers	125	Read of 0x items
5	Force Single Coil		Write of a single 0x item
15	Force Multiple Coil	800	Write of several 0x items
16	16 Preset Multiple Registers	100	Write of any number of 4x items
17	Report Slave ID		Read of device operating mode
22	Mask Write 4X registers		Write of a byte item located in the 4x area
42	IEC Runtime FC		Access to unlocated Concept variables (read/write)
126	Modsoft FC		Starting / Stopping the device

UNITE request codes used by OPC Factory Server:

Request code	Function name	OFS features using the request code
0x38	Read Object List	Read of all items on TSX Premium, Micro
0x36	Read Object	Read of all items on TSX Series 7, S1000
0x83	Write Generic Object	Write of all items on TSX Premium, Micro
0x37	Write Object	Write of all items on TSX Series 7, S1000
0x0F	Identify	Device detection
0xFA	Mirror	Detection of max PDU size
0x4F	Read CPU	Device detection and read of the device's operating mode on TSX Premium, Micro
0x24	Start	Starting the device
0x25	Stop	Stopping the device
0x33	Initialize	Initializing the device

9.4 Recommendations in case of problems

Location of a problem

Description

The table below shows a number of problem situations that can be got out of without any difficulty. If the solution suggested does not solve the problem, contact the Schneider Automation support desk.

Solutions:

Component	Problem	Solution
Configuration tool Installation	Program start up fails with the code "Ox1AD "	Reinstall the MDAC component. It is supplied on the CD in the REDIST directory.
Configuration tool Start	Serious problem while executing the program (untimely stopping of the PC, etc.) or incorrect start up (for example, damaged database)	If you have backed-up aliases: If the configuration tool still starts up, retrieve the last alias file using the "get archive" menu. If the configuration tool does not start up, try to copy the back-up file saved in "alias2K.mdb" in the configuration tool directory manually. This operation will overwrite the working database, which is probably corrupted. Then try to restart the program. If you have not backed-up: Uninstall the configuration tool, then reinstall it. The compatibility window will be displayed during the first start up of the configuration tool, select YES to start the recovery procedure.
Configuration tool Installation	After upgrading from version 2.0 to the current version, the parameters set with the v2.0 configuration tool no longer appear.	It is likely that this is due to the fact that no alias has been declared, the only case in which recovery is not possible.
Configuration tool Backup copy	After an aliases back-up, the backed-up file cannot be located	If you have backed up via the network neighborhood, it is imperative that a directory name is given. If not, the file will be backed-up in the configuration tool's default directory or in the directory containing the configuration tool start up short cut.

Component	Problem	Solution
Configuration tool **Start up**	Alias recovery does not work	Check in the properties of the short cut used to start up the configuration tool that the brackets in the chain of ofsconf.exe characters are not double brackets.
Configuration tool **On line help**	Online help does not work	Online help requires at least Internet Explorer v3.02. This problem usually occurs with W95.
Configuration tool	My new configuration tool parameters are not taken into account	Confirm the configuration tool and close and then reopen the server
Configuration tool	NOK connection between the client and the remote server	Check not only the DCOM parameters, but also the 'DCOM Security' option of the configuration tool
Driver	Communication with the PLC is not working.	For XWAY, check with the "XWAY TEST" utility of the "XWAYDriver Manager" whether communication is possible. If not: The problem is either linked to the driver or to the connections (see appropriate manuals), If yes: Check the consistency level requested; it is likely that the PLC application version may be different from that of the symbols file.
Installation	There may be a few problems if the destination directory access path (the directory under which the server and configuration tool should be installed) is too long and the hard drive on your PC has been converted from FAT to NTFS.	In this case, try to use short file names (for example,C:\OFS - instead of C:\Program Files\Modicon Telemecanique\OFS).
Server, Target device: TSX Premium	Performance is greatly inferior to that expected and/or indicated in the documentation.	Make sure: ● that you have checked the "Check Device" option, ● that you have not inadvertently checked the "Series 7" option in the properties page.
Server **Installation**	The message "QueryInterface(IID_IOPCServer) returned E_NOINTERFACE for server Schneider-Aut.OFS" appears when an OPC client tries to connect to the server.	It generally appears when you have chosen not to restart the PC after installation. Restart it.
Server **Error codes**	Error messages referenced by a numerical code are displayed in the diagnostics window or in the client.	A decoding program is supplied : start up scoder.exe from the server's installation directory.
Server **Concept link**	The message "cannot connect to local cc2cat" or "unable to load Concept .prj file" appears.	Check that the cc2cat is installed correctly and saved. See readme.txt in the \ConceptLink directory of the CD.

Component	**Problem**	**Solution**
Server **System configuration**	Use of screen savers	The use of screen savers is not recommended with the OFS server, except the blank screen with a password (« lock computer » or "lock WorkStation" option). It is not advisable to use any energy savers
Server **Installation**	User rights in Windows 2000 or NT 4.0	The OFS server can only be installed if the session is opened with an ADMINISTRATOR account. The server and any local client can work perfectly well under the same non ADMINISTRATOR account.
Server **Remote client**	Remote access problems with Windows 2000 or NT 4.0	In order to be used remotely by an OPC DCOM client, the OFS server must be started up using an ADMINISTRATOR account or as an NT utility. Check that the user rights are correctly managed: adjust with the DCOMCNFG.exe. tool
Server **Development of a VB client**	"User-defined type not defined" error during the compilation of the declaration of the "OPC Server" object in Visual Basic.	Instruction with an error in: Dim WithEvents OpcFactoryServer As OPCServer. The interfaces presented by the OFS server are not recognized by Visual Basic. You should save these interfaces using the "Tools" menu > VB6 SP3 "References", then select SA OPC Automation 2.0 as this OPC Automation manager contains the server's "Library Type".
Server **Use of symbols**	EOL_E_OPEN_SYMBOLS_FAILURE error obtained when creating a user group with a symbols file.	If you have not given an absolute path for the file, check that the "Symbols" option is correct in the registry. If you have not defined this option, check that the C:\OPC_SYMB directory exists and that it contains your file. If you have put a space after "=" in the group name, then it must be removed. For example: "grpName= symb.scy" becomes "grpName=symb.scy".
Server Target device: Series 7	Failure in the declaration of an item and display in the "XWAY error path: Answer Thread: Invalid Protocol (UNITE V1?) for Answer 253"	Check in the device's alias address that the "Series 7" label is present (validation of the API series 7 parameter in the assisted address entry screen).
Server **OFS manager**	The "Device Identity" field shows "????"	If it is a type S1000 device, this is normal, the PLC's Ethernet module does not return any identification.
Server **Unitelway**	The message "XWAY: Build Request for UNTLW01 :0.254.0 : No Free Socket" appears frequently in the diagnostics window	The UNITELWAY driver is adjusted with too low a number of slave addresses. This can be changed depending on the number of requests used.

Component	Problem	Solution
Server Diag buffer	The quality attribute of the #DiagReadNextError item still says "bad" even when the connection with the device is open.	The "handle" of the OPC client, which contains the specific items of the Diag Buffer, is already in use or the client has connected to the server without specifying their "handle".
Remote station	Installation fails, displaying the message « Automatic save error. One or several files are not saved automatically (OPCAutoSA2.dll, SAProxy.dll)	The DCOM component is necessary. Install it from the Redist\DCOM folder, then restart the PC. Then resume the installation of the remote station.
OPC server or client	The modification of the period of a group takes a lot of CPU time if the group contains a large number of items.	The « Netman. » window, which shows all the requests generated, should be closed since the quantity of traces displayed penalizes the operation. As a general rule, before you modify the period of a group, it is recommended that you first deactivate the group then reactivate it when the operation is complete.
Server Communication	The server no longer responds when lots of items are simulated. "Spurious" messages appear in the Diagnostics window.	Adjust the probability of notifications from the simulator to reduce the number of notifications sent to the OPC client. Increasing the power of the PC can improve the limit.
Server View	In the Windows Task Manager you notice an increase in the amount of memory taken up by OFS.EXE.	If the server is in Diagnostic mode, a large quantity of displayed messages causes an increase in the amount of memory used (up to 4 Mb). This is normal. By closing the HMI windows of the server, the memory is freed up.
Server Installation	The message "CoCreateInstance returned REGDB_E_CLASSNOTREG for Server Schneider" appears when an OPC client tries to connect to the server.	This appears when the MSVCP60.DLL component is missing or is not installed correctly. Put this DLL in the Windows directory and reinstall the server.
Server Concept link	With Concept 2.5, opening the symbols file fails, producing the message "SdkConcp : can’t open project"	Consult the file \ConceptLink\Version2-5\Readme.txt
OPC client Address format of the items	All the items in a group are set to Quality = Bad (24)	This status can result from the addition of a non existent item in the target device ‘Eg : read of a %MW10000 word even though the PLC only contains 8000. Delete the items that are out of range.
Server Write of items	During the write of a large number of items, the following messages appear in the diagnostics window : "SyncWriteFailure" followed by "Write Error"	These messages appear due to the fact that the write operation lasted longer than the authorized time limit. Increase the frame time out for the devices concerned.

Component	Problem	Solution
Server Notification	Absence or delay in the notification of certain items. The message "Error, request too old" appears in the diagnostics window of the server.	Increase the period of the group. If this is not sufficient, alter the time out values of the alias. For Modbus, increase the number of channels allocated for communication.
Server Items access	Access via Modbus to more than 1000 consecutive bits on a Premium PLC causes errors.	Make sure that the option /T is indicated at the end of the device address.
Server Activation of items	The activation of a significant number of items fails if they were created in the active group. This problem usually occurs in W95 and with communication with a low flow.	This problem occurs due to a PC overload. The following recommendations can solve the problem: create the items in an inactive group, then activate the group. Check that the "Verbose" server mode is not selected in order to keep the traces to a minimum. Increase the power of the PC. If the operating system is W95, update it to a more recent system.
Server Push Data	In Modbus, the values of the items defined in the Push Data zone are not updated.	The base address to be indicated in the alias properties should only contain the offset of the address. So, to indicate a base address in Modbus corresponding to 402000 for example, it is sufficient to indicate 2000 as the value in the "Base address" field.

Glossary

A

Address

Control system variable manufacturer name. For example "%MW1".

C

Client application

Software using the primitives provided by a server application, via mechanisms (interfaces) implemented by the OLE.

COM

Component **O**bject **M**odel: foundations of the OLE 2.0 standard.

D

DCOM

Distributed **COM**: COM model distributed in a network under TCP-IP.

F

FIP

Factory **I**nstrumentation **P**rotocol.

H

Handle Single value identifying the object.

L

LCID **L**anguage **C**ode **ID**entifier.

M

Multi-clients Several client applications simultaneously access the same server application.

O

OFS **OPC F**actory **S**erver: OLE server for exchanging data with the PLC.

OLE **O**bject **L**inking and **E**mbedding: object for linking and embedding. In particular supplies the OLE Automation interface, a technique which enables a server to display the methods and properties to a client.

OPC **OLE** for **P**rocess **C**ontrol.

OPC group Controls a collection of **OPC** items, that is a list of PLC variables.

OPC item PLC variable on a PLC and a given communication medium.

OPC server Controls a collection of OPC groups. Hierarchical root of the OPC model.

P

PLC **P**rogrammable **L**ogical **C**ontroller: programmable controller (industrial).

Primitive OPC function

R

Remote server The client and server application are located on 2 separate stations, linked by the Microsoft TCP-IP network.

S

Server application Software showing the primitives to the client applications, via mechanisms (interfaces) implemented by the **OLE**.

Socket Communication channel established between the OFS server and one or more PLCs, on a given communication media. The number of sockets available depends on the communication medium.

SP **S**ervice **P**ack: operating system corrections and upgrades

Symbol Identifier attributed by a designer to a control system. For example "PUMP". A symbol cannot start with the prefix ‘%’.

U

UNC **U**niversal **N**aming **C**onvention.

V

VB **V**isual **B**asic: consumer language supporting OLE Automation.

VBA **V**isual **B**asic for **A**pplication: script language with Basic syntax included in the MS-Office suite.

W

Wintel **Windows/Intel**: describes a PC equipped with a 32-bit Windows operating system and an Intel x86 processor.

Index