PROCESS AUTOMATION

Freelance 2019

Engineering Manual OPC Server F

PROCESS AUTOMATION

Freelance 2019

Engineering Manual OPC Server F

Document Number: 3BDD012511-111
Revision: A
Release: March 2019

Notice

This document contains information about one or more ABB products and may include a description of or a reference to one or more standards that may be generally relevant to the ABB products. The presence of any such description of a standard or reference to a standard is not a representation that all of the ABB products referenced in this document support all of the features of the described or referenced standard. In order to determine the specific features supported by a particular ABB product, the reader should consult the product specifications for the particular ABB product.

ABB may have one or more patents or pending patent applications protecting the intellectual property in the ABB products described in this document.

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.

Products described or referenced in this document are designed to be connected, and to communicate information and data via a secure network. It is the sole responsibility of the system/product owner to provide and continuously ensure a secure connection between the product and the system network and/or any other networks that may be connected.

The system/product owners must establish and maintain appropriate measures, including, but not limited to, the installation of firewalls, application of authentication measures, encryption of data, installation of antivirus programs, and so on, to protect the system, its products and networks, against security breaches, unauthorized access, interference, intrusion, leakage, and/or theft of data or information.

ABB verifies the function of released products and updates. However system/product owners are ultimately responsible to ensure that any system update (including but not limited to code changes, configuration file changes, third-party software updates or patches, hardware change out, and so on) is compatible with the security measures implemented. The system/product owners must verify that the system and associated products function as expected in the environment they are deployed.

In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document and parts thereof must not be reproduced or copied without written permission from ABB, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

The software or hardware described in this document is furnished under a license and may be used, copied, or disclosed only in accordance with the terms of such license. This product meets the requirements specified in EMC Directive 2014/30/EU and in Low Voltage Directive 2014/35/EU.

Trademarks

All rights to copyrights, registered trademarks, and trademarks reside with their respective owners.

Table of Contents

About this book

1 OPC – General description

2 OPC server

3 OPC client

Appendix A Example of OPC access from MS-Excel and MS-Visual Basic

Appendix B Representation of the Freelance message types

Appendix C OPC keyword tracing

Appendix D OPC server – System messages

Index

About this book

Use of warning, caution, information, and tip icons

This publication includes **Warning**, **Caution**, and **Information** where appropriate to point out safety related or other important information. It also includes **Tip** to point out useful hints to the reader. The corresponding symbols should be interpreted as follows:

⚡ Electrical warning icon indicates the presence of a hazard which could result in *electrical shock.*

⚠ Warning icon indicates the presence of a hazard which could result in *personal injury.*

❗ Caution icon indicates important information or warning related to the concept discussed in the text. It might indicate the presence of a hazard which could result in *corruption of software or damage to equipment/property.*

ℹ Information icon alerts the reader to pertinent facts and conditions.

💡 Tip icon indicates advice on, for example, how to design your project or how to use a certain function.

Although **Warning** hazards are related to personal injury, and **Caution** hazards are associated with equipment or property damage, it should be understood that operation of damaged equipment could, under certain operational conditions, result in degraded process performance leading to personal injury or death. Therefore, comply fully with all **Warning** and **Caution** notices.

Terminology

The Glossary contains terms and abbreviations that are unique to ABB or have a usage or definition that is different from standard industry usage. Please make yourself familiar to that.

You will find the glossary at the end of the ***Engineering Manual System Configuration.***

Document conventions

The following conventions are used for the presentation of material:

- The words in names of screen elements (for example, the title in the title bar of a window, the label for a field of a dialog box) are initially capitalized.
- Capital letters are used for the name of a keyboard key if it is labeled on the keyboard. For example, press the ENTER key.
- Lowercase letters are used for the name of a keyboard key that is not labeled on the keyboard. For example, the **space bar**, **comma key**, and so on.
- Press CTRL+C indicates that you must hold down the CTRL key while pressing the C key (to copy a selected object in this case).
- Press **ESC, E, C** indicates that you press and release each key in sequence (to copy a selected object in this case).
- The names of push and toggle buttons are boldfaced. For example, click **OK**.
- The names of menus and menu items are boldfaced. For example, the **File** menu.
  - The following convention is used for menu operations:
    MenuName > MenuItem > CascadedMenuItem
    For example: Select **File** > **New** > **Type**.
  - The **Start** menu name always refers to the **Start** menu on the Windows Task Bar.

- System prompts/messages are shown in the Courier font, and user responses/input are in the boldfaced Courier font. For example, if you enter a value out of range, the following message is displayed:

  ```
  Entered value is not valid. The value must be 0 to 30.
  ```

  You may be told to enter the string TIC132 in a field. The string is shown as follows in the procedure:

  TIC132

  Variables are shown using lowercase letters.

  sequence name

1 OPC – General description

OPC is the abbreviation for OLE for process control. This is an open interface standard based on the Windows-based technology of OLE (Object Linking and Embedding) and COM (Component Object Model); it enables data to be exchanged simply and in a standard way between applications from the fields of industry and commerce as well as manufacturing.

Data from the process stations as well as the connected field bus instruments is transmitted to other systems with an OPC client interface through an **OPC server** interface.

Through the **OPC client** interface, Freelance (Engineering and Operations), with its operation and observation functions, can display data from automation instruments and systems from other manufacturers.

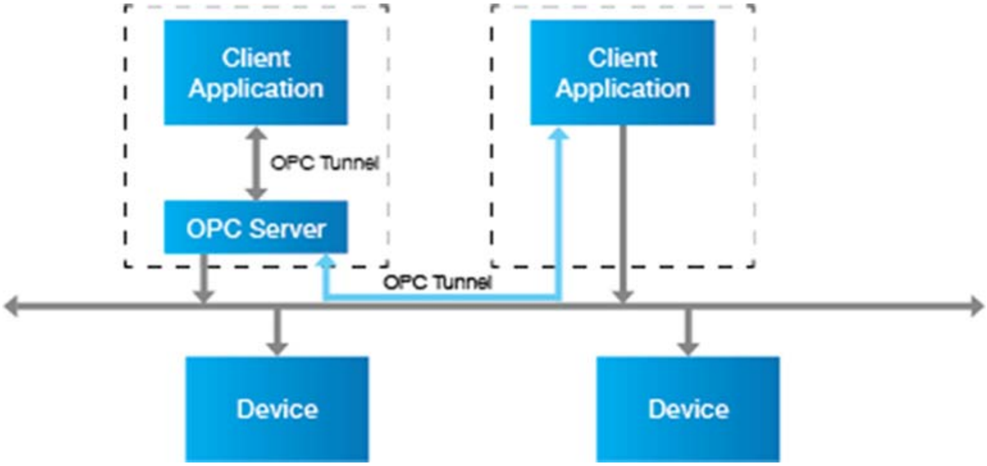
- Every value acquired through OPC has a time stamp and a status.
- OPC uses a block format for communication, and in a single call it can send several requests and receive several values.
- OPC allows client applications to specify required scan rates.
- OPC uses binary data representations (IEEE floating-point numbers, integers...).
- The variables for a server can be viewed and selected using a browser.
- The concept of creating groups allows different sets of variables with different scan rates.

Two interfaces are implemented

1. Custom Interface	Data Access 2.0 Alarms&Events 1.0
2. Automation Interface	Data Access 2.0A

The functions identified as ‘required’ in the OPC specifications have been implemented for all interfaces. The *browser interface* has also been implemented. These functions enable a remote client to read the list of known variables in an OPC server.

OPC overview: OPC server coupled directly with COM and distributed with OPC tunnel.



OPC.png

Term	Explanation
OPC	OPC stands for ‘OLE for Process Control’ and is based on Microsoft's core OLE technologies COM.
COM	When objects have been implemented on different platforms or computer architectures it is necessary to establish how these platforms interpret an object before the objects can be made mutually compatible. A so-called object model is required for this purpose. OLE uses the model COM (Component Object Model). This model defines the standard for the working relationship between the components. COM enables calls to be made within a process and to another process.
ABB OPC tunnel	Links the OPC server to the Freelance system (Freelance Engineering and Freelance Operations).

Depending on the operating systems where OPC server and client are running and depending on the requirements of the OPC client it might be necessary to adjust some DCOM settings. The detailed information for this DCOM settings are available in the Windows documentation of the used operating systems. If needed please contact ABB support.

1.1 OPC client

Freelance Engineering allows variable lists from other OPC servers to be read and imported. The imported data can be used for configuring graphic displays, trends and logs.

During commissioning, the imported variables can be displayed in the values window and trend window in exactly the same way as the variables from the process stations.

Using the OPC client functionality of Freelance Operations, other systems' variables can be read from their OPC server. As soon as a graphic object, a trend or a log is called up in an operator stations that requires variables from remote OPC servers, the OPC client that is integrated in Freelance Operations retrieves them from the OPC server as a group.

1.2 OPC server

With the OPC server, data from a Freelance project is made available to other systems. As a result, e.g. other visualization packages, such as 800xA Operator Workplace, PlantConnect, InTouch, The Fix, etc., can access data from the process stations and the connected field bus instruments.

The OPC server is a stand-alone program which is installed on any PC within the system network. An OPC type gateway is configured in the Freelance project. During commissioning the data that is to be made available to the remote system is downloaded to the gateway station.

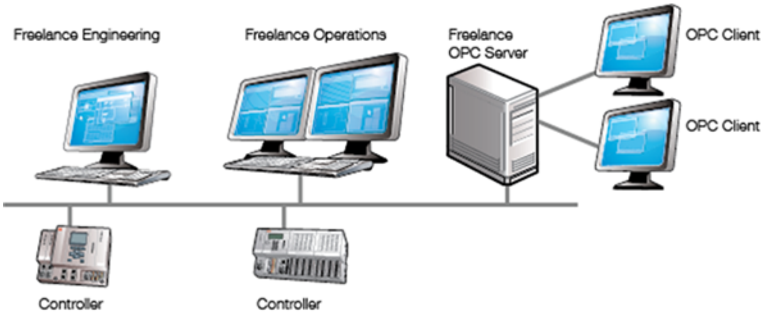
Thus any OPC compatible Windows program (with client function) can read and write the values of all the variables available in the gateway (inputs, outputs and parameters of function blocks and variables) "online" from the connected process stations and field instruments.

Using “MS Visual Basic” or “MS Visual Basic for Applications” in MS Excel, users can generate their own programs and thus read and write OPC data “online”. This data can then be displayed, recorded and evaluated.

1.3 System structure

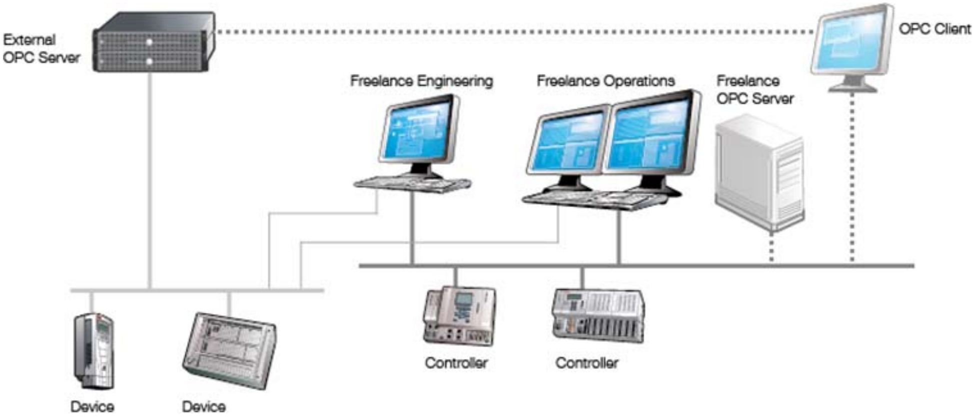
Note the different function of Freelance Engineering in the two diagrams.

OPC server:



OPC_Server.png

External OPC server:



OPC_Ext.png

1.4 Resources, maximum values and performance

OPC gateway stations in a Freelance project:
Unlimited

OPC gateway stations per process station:
Max.: 10
Each process station can communicate with max. 10 operator stations and/or gateway stations.

OPC servers per gateway PC:
Dependent on the performance of the PC

Communication volume:
per OPC server to one process station approximately max. 2500 REAL variables with the maximum of 10 possible connections; 1400 bytes per connection = approximately 250 variables per connection to a process station

Update time for OPC server: Default = 1000 ms
The cycle time should be configured in accordance with the client.

Throughput capacity:
- Dependent on the gateway PC
- Dependent on the client (Most efficient when programmed in “C” Excel and Visual Basic cannot achieve the same level of performance).
- From a process station
which performs communication exclusively
Up to 5000 variables/sec can be sent to the OPC server.

If the time zone has not been set or has been set incorrectly, the server synchronizes the client’s system time cyclically. The time synchronization can be enabled or disabled in the Freelance **Settings** tool.

2 OPC server

An OPC server is used to make process data from a Freelance project available to other systems through the OPC interface.

2.1 Configuration using Freelance Engineering

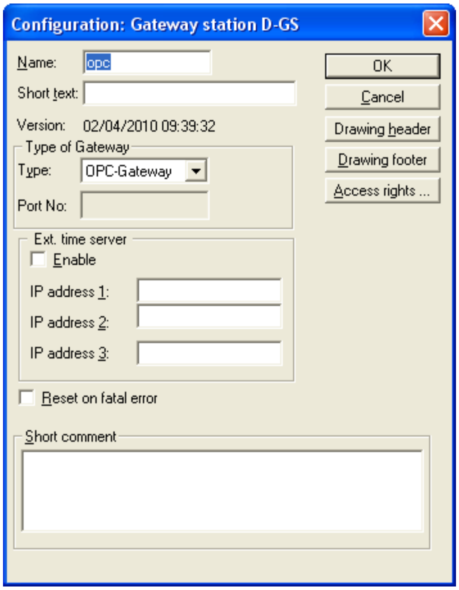
In order to make the data from a project available to other systems, a gateway must be configured in the Freelance project. A gateway resource is added in the project tree and subsequently configured to determine which tags and variables can be accessed through this gateway.

Server software must be installed on the gateway station. This software communicates with the Freelance process stations and supplies the data to the outside over the OPC interface.

In order to implement an OPC server interface, a gateway of type OPC is configured in the Freelance project tree. The OPC server software is installed on the OPC gateway computer. Following this, the configured project data is downloaded to the OPC gateway with Freelance Engineering in commissioning mode. Immediately after downloading, the Freelance data can be accessed through the OPC interface.

2.1.1 Adding a gateway station

Add resource **Gateway station D-GS** in the project tree > Select type *OPC gateway*



Configuration: Gateway station D-GS

Name: opc

Short text:

Version: 02/04/2010 09:39:32

Type of Gateway

Type: OPC-Gateway

Port No:

Ext. time server

☐ Enable

IP address 1:

IP address 2:

IP address 3:

☐ Reset on fatal error

Short comment

op004us.bmp

Name Name of the gateway station, max. 4 characters

Short text Max. 12 characters

Version Date and time the object was created

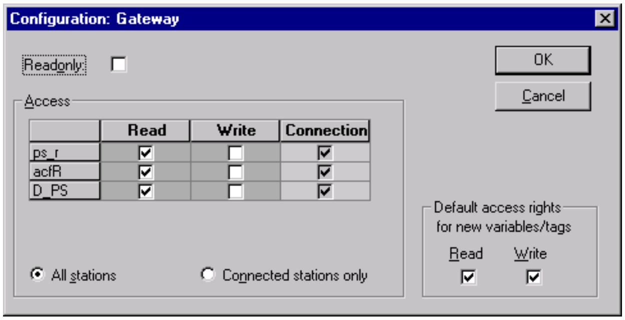
Drawing footer/ Drawing header
Enter the drawing footer or header. For more details, refer to ***Engineering Manual System Configuration, Documentation.***

Access rights Read and/or write rights for the gateway; default access for new variables / tags

Port No Port number for the ABB OPC tunnel communication; default value is 9950.

Short comment Enter the short comment. For more details, refer to ***Engineering Manual System Configuration***.

2.1.2 Access rights for a gateway



Readonly: □

Access

	Read	Write	Connection
ps_r	☑	□	☑
acfR	☑	□	☑
D_PS	☑	□	☑

● All stations ○ Connected stations only

Default access rights for new variables/tags

Read ☑ Write ☑

op005us.bmp

Read only The gateway is only allowed read access to the process variables. Any write accesses that are configured on the process stations, variables and tags are ignored.

Access A parameter is set for every process station that is already configured in the project to specify whether read-only access or read/write access is permitted. If read-only (not write) access is specified for a process station here, then any write accesses that are configured on the variables and tags are ignored. The parameter *read-only* for the entire gateway overrides any write access that is defined here.

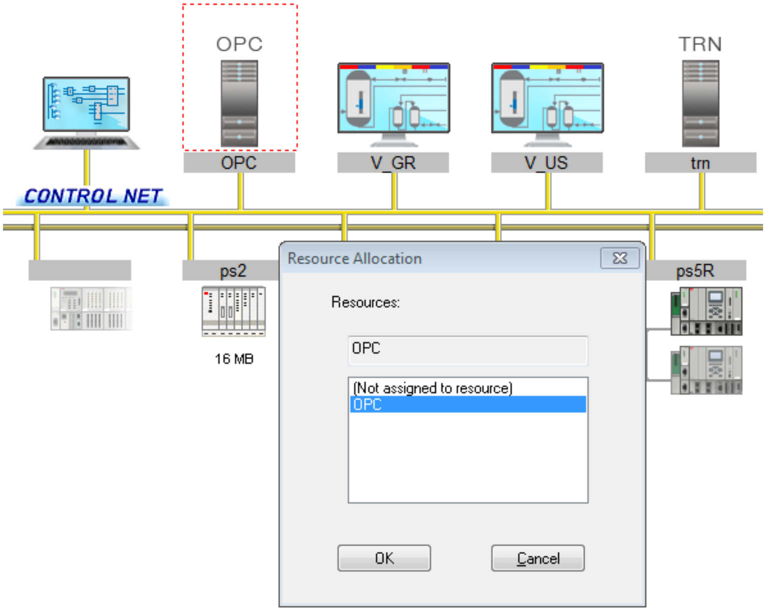
Default access for new variables / tags

All newly-created variables and tags are by default assigned the access rights defined here for this gateway.

Variables that are written through a gateway may not be simultaneously written through the process image. The values of such variables are thus not capable of redundant operation. If these variables are used in a redundant task, then they should be duplicated to other variables.

2.1.3 Adding the gateway station to the hardware structure

A free space within the control level must be selected in the hardware editor’s system view, and this must be assigned to a gateway resource that has already been created in the project tree.



Resources: OPC

(Not assigned to resource)
OPC

OPC09_us.png

2.1.4 Network addresses

Within the hardware editor, the stations IP addresses and resource IDs are configured under *Network*. The values specified here must correspond to the parameters defined during the installation of the OPC server software.

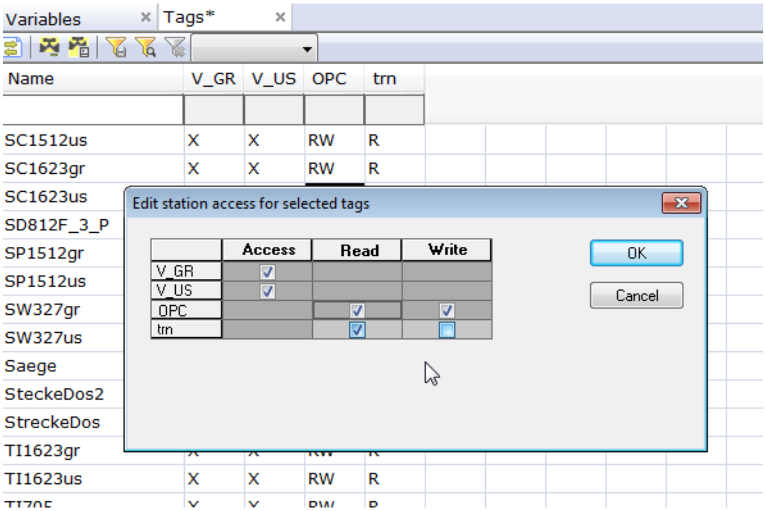
2.1.5 Enabling tags and variables for OPC access

A parameter is defined for each tag and each variable in the project to specify whether the data from the process station can be only read or read and written through the gateway.

In station view (**Tag list** or **Variable list** > **Station view**), the permitted access modes through a gateway can be defined for each list entry. The station accesses that have been defined are indicated in the variable list and tag list by single letters:

R	Read
W	Write

A double-click with the left mouse button in a gateway column or positioning the cursor in a line and choosing the menu item **Edit /Station access...** brings up the dialog for configuring the access rights. The station access for more than one list entry can be changed at the same time: select more than one entry in the list by means of Shift + arrow keys before choosing (Station access...) from the menu.



Name	V_GR	V_US	OPC	trn
SC1512us	X	X	RW	R
SC1623gr	X	X	RW	R
SC1623us				
SD812F_3_P				
SP1512gr				
SP1512us				
SW327gr				
SW327us				
Saege				
SteckeDos2				
StreckeDos				
TI1623gr	X	X	RW	R
TI1623us	X	X	RW	R
TI705	X	X	RW	R

Edit station access for selected tags

	Access	Read	Write
V_GR	☑		
V_US	☑		
OPC		☑	☑
trn		☑	☐

OPC21us.png

Left-hand column

List of all the operator stations and gateway stations that have been configured; in the variable list only the gateway stations are shown.

Read The previously selected variable(s) and/or tag(s) data can be read from the process station through the gateway.

Write The previously selected variable(s) and/or tag(s) data can be modified in the process station through the gateway (only user-defined variables can be written to, while pre-defined process station variables can only be read).

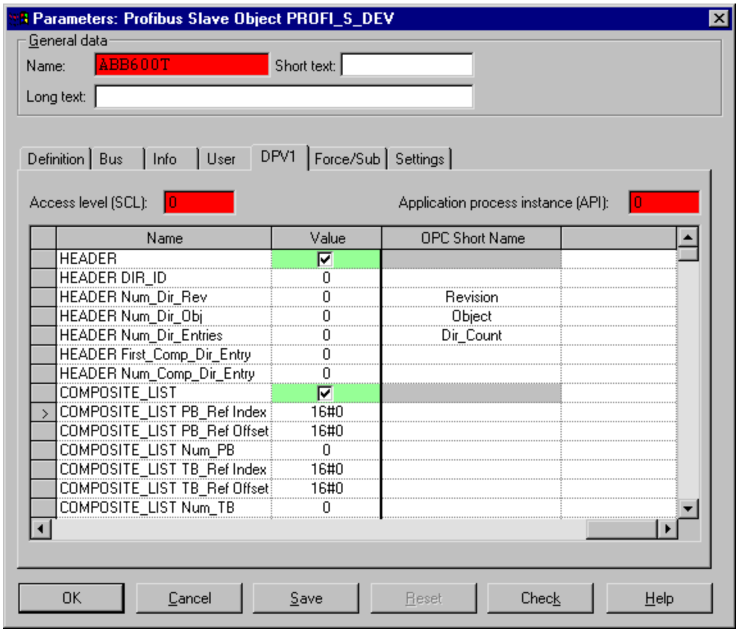
Variables that are written through a gateway may not be simultaneously written through the process image. When the process image is used, a variable that has been modified by the OPC server can be overwritten again by the task.

Variables with no process image are not capable of redundant operation. If such variables are used in a redundant task, then they should be duplicated to other variables

2.1.6 Enabling PROFIBUS parameters for OPC access

The PROFIBUS DPV1 and user parameters are configured to determine whether the data is available through the OPC gateway.

There are two columns for configuring OPC access in the overlays of the DPV1 parameters and the user parameters. A check box in the OPC access column is used to configure whether the parameter is loaded into the OPC gateway or not. The corresponding parameter can be accessed in the gateway under **<tag name>/<component name>**. If a name is entered in the **OPC short name** column, this name will be used in the gateway instead of the component name.



Name: ABB600T

Short text:

Long text:

Access level (SCL): 0

Application process instance (API): 0

Name	Value	OPC Short Name	
HEADER	☑		
HEADER DIR_ID	0		
HEADER Num_Dir_Rev	0	Revision	
HEADER Num_Dir_Obj	0	Object	
HEADER Num_Dir_Entries	0	Dir_Count	
HEADER First_Comp_Dir_Entry	0		
HEADER Num_Comp_Dir_Entry	0		
COMPOSITE_LIST	☑		
COMPOSITE_LIST PB_Ref Index	16#0		
COMPOSITE_LIST PB_Ref Offset	16#0		
COMPOSITE_LIST Num_PB	0		
COMPOSITE_LIST TB_Ref Index	16#0		
COMPOSITE_LIST TB_Ref Offset	16#0		
COMPOSITE_LIST Num_TB	0		

op050gr.bmp

OPC access The parameter can be accessed through an OPC gateway.

OPC short name
The parameter is not addressed in the OPC gateway by the tag name and component name in the **Name** column, but by the name entered here. This name must be unique within an object.

The components of a PROFIBUS object which are available through an OPC gateway are defined by means of these dialogs. The gateway station through which this data can be accessed is configured in the tag list for the object.

2.1.7 Initiating communication

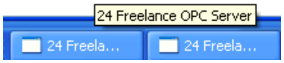
An OPC server is started automatically when requested by an OPC client.

It can be started manually by calling the command window in Windows from the installation default directory, <Freelance install dir>\exe\...:

OPCSRV.EXE /xx

xx = assign resource ID no., e.g. 24 as defined during Setup

When an OPC server has started, the following icon appears in the Windows taskbar:



op010us.bmp

Several OPC servers can be started simultaneously on one PC as long as they have different resource IDs.

2.1.8 Downloading project data to the gateway station

Like the process stations, a gateway station also needs to be commissioned.

After the project has been checked for plausibility and all the process stations loaded, the project data must also be downloaded to the gateway station. The first time a gateway is commissioned, **Load** > **Whole station** must be executed. Configuration changes are transferred to the gateway station through **Load** > **Changed objects**.

In the course of the downloading process from Freelance Engineering, the gateway station receives notification about all variables and tags that can be accessed through this gateway. The variable data and tag data can be read and written from the external client.

2.2 Addressing the Freelance variables

A variable is addressed using the same name within the system and over the OPC interface.

Input pins, output pins and parameters of a tag are addressed using the tag name and the pin or parameter name:

<Tag name> / <Pin name or parameter name>

Example: TIC123/SP Read the set-point input of controller TIC123

TIC123/Wi Read the internal set-point of controller TIC123

The component and parameter names of the Freelance standard function blocks can be found in the ***Engineering Manual Functions and Function Blocks***.

The component and parameter names of the user defined function blocks can be found in the class definition of the block. All the elements of the block interface that reside in the process station can be accessed through the OPC server.

The component names of the PROFIBUS objects can be modified, see Enabling PROFIBUS parameters for OPC access on page 22. The current names can be found in the associated configuration dialogs.

The OPC interface is case-sensitive. The variable names must therefore be identical in OPC and in the Freelance project.

2.3 Data types of the variables

The procedures for the OPC interface are based on the definitions for COM. Canonical data types are used for data transmission within these definitions. The data types involved here are those which the Microsoft environment supports as basic data types.

A data request through OPC can be issued either with or without data type. In the case of data requests with no data type (variant type), the Freelance system data types are mapped as follows: The unsigned data types must be read in Visual Basic applications as Long (VT_UI2) or as Single (VT_UI4). The optional parameter "RequestedDataTypes" of the AddItems method must be set to VT_I4 where the default Variant datatype is VT_UI2 and to VT_R4 where the default is VT_UI4.

Data type in Freelance	Canonical data types in OPC	Visual Basic data type
BOOL	VT_BOOL	Boolean
BYTE	VT_UI1	Byte
WORD	VT_UI2	Long
UINT	VT_UI2	Long
INT	VT_I2	Integer
DWORD	VT_UI4	Single
UDINT	VT_UI4	Single
DINT	VT_I4	Long
REAL	VT_R4	Single
TIME	VT_I4	Long
DT	VT_DATE	Date
STRING8.. STRING256	VT_BSTR	String

The Freelance data types are described in the ***Engineering Manual IEC 61131-3 Programming, Variables***.

Only data requests with a specific data type in which it is always possible to perform a conversion are accepted. The possible combinations are shown in the table below:

	Conversion is possible to **VT_xx**									
Freelance	**BOOL**	**UI1**	**UI2**	**UI4**	**I1**	**I2**	**I4**	**R4**	**R8**	**DATE**	**BSTR**
BOOL	X	X	X	X	X	X	X	X	X		X
BYTE	X	X	X	X		X	X	X	X		X

	Conversion is possible to **VT_xx**										
Freelance	**BOOL**	**UI1**	**UI2**	**UI4**	**I1**	**I2**	**I4**	**R4**	**R8**	**DATE**	**BSTR**
WORD	X		X	X				X	X		X
UINT	X		X	X				X	X		X
INT	X					X	X	X	X		X
DWORD	X			X				X	X		X
UDINT	X			X				X	X		X
DINT	X						X	X	X		X
REAL	X							X	X		X
TIME	X						X	X	X		X
DT	X									X	X
STRINGx											X

The system response with regard to the data type conversion requested can be set in the registry:

Windows 32bit:

[HKEY_LOCAL_MACHINE\SOFTWARE\Hartmann & Braun\Freelance \OPCServer]

Windows 64bit:

[HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Hartmann & Braun\Freelance \OPCServer]

With the default setting **“OptimisticDataConversion = 0”** the OPC server behaves as described above.

With the setting **“OptimisticDataConversion = 1”** data inquiries of a specific data type (requested data type) are always accepted as far as data conversion is, in principle, possible.

If the determined value should be out of the valid range, it will be transmitted with a 'Bad Quality' attribute.

2.4 Message priorities, severity

Freelance incorporates the message priorities 0 to 5. Message types are linked to these priorities. The system messages (priority 0) are also subdivided between “system message groups” S1, S2 and S3. These groups become visible at the operator interface and can be used for filtering and initiating actions.

In order to display a message priority there is the **severity** attribute within the OPC server. Severity values can range from 1 to 1000, whereby 1000 represents the highest priority. Incoming and outgoing messages are reported with the same severity. The value 0 is reserved: In response to a query of alarm status, severity value 0 is sent for an inactive, acknowledged message.

The priorities of the Freelance messages are mapped to the following severity values:

Process station		**OPC**
Prio	**Type**	**Severity**
0	System S1	900
0	System S2	500
0	System S3	100
1	Fault	1000
2	Fault	800
3	Fault	600
4	Switch	400
5	Hint	200

2.5 Message types, category and condition

The message types of the Freelance system are mapped to the attributes **category, condition and subcondition** in the OPC server.

The process messages of a Freelance system are subdivided into categories and conditions of OPC alarms. There are no subconditions for these alarms. In accordance with the OPC specifications, the strings which are then delivered for condition and subcondition are identical.

The index of the associated alarm point in the function block (starting with ‘0’ for the first alarm point) has been added to the list of strings in the table in the appendix. If a high-level alarm is configured for the first alarm point in a function block, then this alarm is reported with the string “H_0” for condition and subcondition. See for information, Appendix B, Representation of the Freelance message types.

The system messages of a Freelance system are all reported with the **category OPC_CATEGORY_SYSTEM**. The text contents correspond to the messages on an operator station. See Appendix D, OPC server – System messages for the list of the system messages.

The system messages are transmitted in the language in which the OPC server was installed. All other text, e.g. conditions and subconditions, are available only in English.

Depending on the message category, various attributes of the OPC message are supplied with current data from the Freelance system.

Attribute	**Description**	**Data type**	**Category**							
			Level	**RateOfChange**	**Deviation**	**Duration**	**Discrete**	**User**	**Limit**	**System**
Area	System area	VT_BSTR	X	X	X	X	X	X	X	X
Hint	Hint text	VT_BSTR	X	X	X	X	X	X	X	X
WaveFile	WAV file	VT_BSTR	X	X	X	X	X	X	X	X
LongText	Long text		X	X	X	X	X	X	X	X
ShortText	Short text	VT_BSTR	X	X	X	X	X	X	X	X
CV	Current value	VT_VARIANT	X	X	X	X	X	X	X	X
Dimension	Dimension	VT_BSTR	X	X	X	X	X	X	X	X

Attribute	Description	Data type	Category							
			Level	RateOfChange	Deviation	Duration	Discrete	User	Limit	System
AcknowledgeRule	Acknowledgement rule	VT_BSTR	X	X	X	X	X	X	X	X
ResourceID	Resource ID	VT_I2	X	X	X	X	X	X	X	X
ObjectNumber	Object number	VT_I2	X	X	X	X	X	X	X	X
AlarmIndex	Index of message point	VT_I2	X	X	X	X	X	X	X	X
ConditionSubType		VT_BSTR	X	X	X					
ErrorString	Error text	VT_BSTR								X
ErrorCode	Error number	VT_I4								X
SystemMessageGroup	Message group S1..S3	VT_BSTR								X

2.6 Project version control

When data is accessed within the OPC server for reading and writing, the internal project version is compared with that in the process stations. The response in the event of a version error is defined in the PC's registry entries. In this way, it is possible to ensure that no read/write operations are carried out on a process station that has been reconfigured unless the OPC server has been reloaded.

In general, following a configuration change to and subsequent loading of a process station, the OPC server must also be loaded. If no OPC related data has been modified, then only the project version number needs to be updated. This modification is post-loaded automatically from Freelance Engineering.

The version control for writing through the OPC server is by default set more restrictively than for reading. This means that, in the event of a version conflict, data can be read but not written.

No version controls of any kind are carried out for transmitting alarms and alarm acknowledgements. This means that every alarm message from the Freelance system is forwarded to the OPC client, and every acknowledgement of an alarm is likewise forwarded from a client to the process stations.

Registry entries for the OPC server:

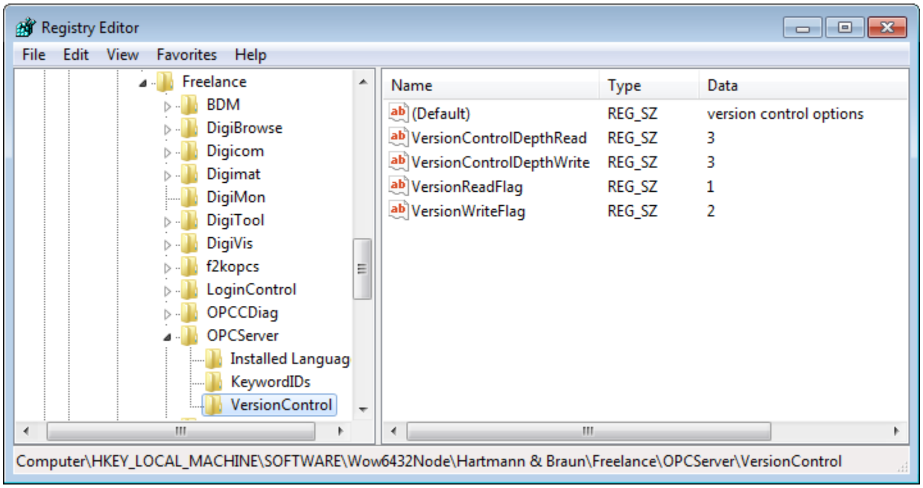
The version monitoring entries are located in the following path:

Windows 32bit:

> [HKEY_LOCAL_MACHINE\SOFTWARE\Hartmann & Braun\Freelance\VersionControl]

Windows 64bit:

> [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Hartmann & Braun\Freelance\VersionControl]



Name	Type	Data
(Default)	REG_SZ	version control options
VersionControlDepthRead	REG_SZ	3
VersionControlDepthWrite	REG_SZ	3
VersionReadFlag	REG_SZ	1
VersionWriteFlag	REG_SZ	2

Computer\HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Hartmann & Braun\Freelance\OPCServer\VersionControl

op025us.png

VersionControlDepthRead / Write

Specifies the differences between process stations and gateway stations that produce a version error.
0: No version control
1: Project names are checked
2: Project names + MajorVersionNo are checked
3: Project names + MajorVersionNo + MinorVersionNo are checked
4: Project names + MajorVersionNo + MinorVersionNo + PatchVersionNo are checked

The **default value is 3**, that is the name, MajorVersionNo and MinorVersion-No of the project must be identical on the process station and the gateway in order to avoid having a version error reported when a variable is accessed.

VersionReadFlag / WriteFlag

Defines the response when a version error is detected (corresponds to the “VersionControlDepth” setting).
1: The variable is read or written even if a version error is detected.
2: Any version error that is detected will be reported, while the read- or write access requested is not carried out.

Defaults:
The default value for **read accesses is 1**, in other words the read command will be carried out even when a version error has been detected.
The default value for **write accesses is 2**, in other words if a version error is detected, it is reported and the write command is not carried out.

2.7 Alarm messages from the OPC server

The OPC server generates its own alarm messages. These messages indicate a fault in the connection state between the OPC server and the subordinate process station. The gateway name from the project tree is used as the source for these messages. These alarms are always single-value alarms, that is no acknowledgment is expected.

The following alarm messages are implemented:

Status of the connection with a process station

Source: <Gateway name in project tree>

Condition: **<Name of process station >_CONNECTION_STATE_0**

with the following subconditions:

NOT_CONNECTED	“No connection to station %d”
ESTABLISHING_CONNECTION	“Establishing connection to station %d”
PROJECT_VERSION_ERROR	“Version error, project name different in station %d”
MAJOR_VERSION_ERROR	“Version error, major version number different in station %d”
MINOR_VERSION_ERROR	“Version error, minor version number different in station %d”

Read/write status of the individual process stations

The connection status and the registry values are evaluated for the version control system.

Source: <Gateway name in project tree>

Condition: **<Name of process station>_READ_WRITE_STATE_1**

with the following subconditions:

NO_READ_ACCESS	“No read access to station %d”
NO_WRITE_ACCESS	“No write access to station %d”
NO_ACCESS	“No access to station %d”

OPC server offline:

This is activated if the OPC server has lost its connection to all process stations.

Source: <Gateway name in project tree>

Condition: **OPC_SERVER_OFFLINE**

with the following subcondition:

OPC_SERVER_OFFLINE_0	“OPC server %d offline”

Handling of NOT_CONNECTED with redundant process stations

If a connection is lost with a redundant resource, it is assumed that a redundancy transfer is taking place. Only if the connection has still not been re-established after 4 seconds does the disconnection have an effect on the quality of items and alarms. The relevant internal alarms are also generated at this point.

The system response can be set in the registry:

Windows 32bit:

> [HKEY_LOCAL_MACHINE\SOFTWARE\Hartmann & Braun\Freelance\OPCServer]

Windows 64bit:

> [HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Hartmann & Braun\Freelance\OPCServer]

“RedToggleQualityGood = 0” Disconnection is reported immediately,

“RedToggleQualityGood = 1” Disconnection is only reported after a time-out (default setting)

The length of the time-out can likewise be set in the same registry path:

“RedToggleTimeOut” Value in seconds (default setting = 4)

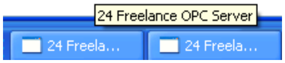
2.8 Tools and utilities for error diagnosis

2.8.1Testing on the process station

The connection monitoring block M_CONN can be used to check from a process station the connection to the OPC server. In the event of a disruption to the connection, specific safety values can be adopted in the process, or a message can also be generated for other operator stations that are not connected through this OPC server.

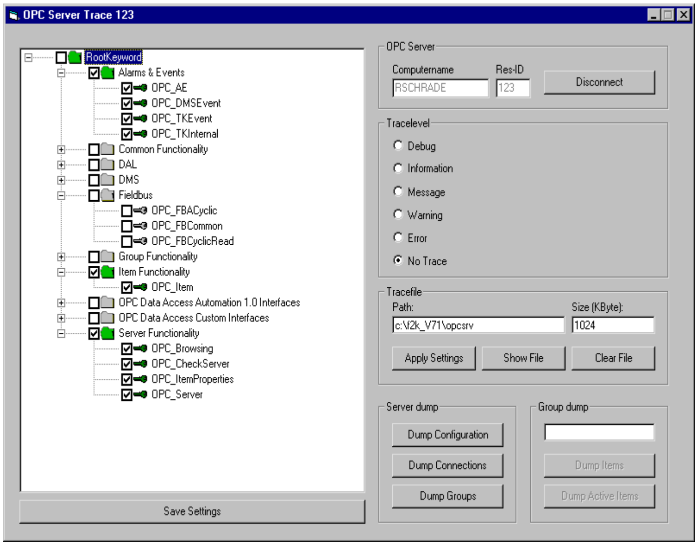
2.8.2 Testing the PC on which the OPC server is executed

If the OPC server has been installed for an interactive Windows user, a running OPC server can be detected on the gateway PC by an entry in the taskbar.



op010gr.bmp

A dialog for monitoring the various sequences of the OPC server can be started in the context menu of this entry. The trace dialog is invoked by means of “Trace...” from the context menu or from the Windows start menu:


OPC Server Trace 123

- ☐ RootKeyword
  - ☑ Alarms & Events
    - ☑ OPC_AE
    - ☑ OPC_DMSEvent
    - ☑ OPC_TKEvent
    - ☑ OPC_TKInternal
  - ☐ Common Functionality
  - ☐ DAL
  - ☐ DMS
  - ☐ Fieldbus
    - ☐ OPC_FBACyclic
    - ☐ OPC_FBCommon
    - ☐ OPC_FBCyclicRead
  - ☐ Group Functionality
  - ☑ Item Functionality
    - ☑ OPC_Item
  - ☐ OPC Data Access Automation 1.0 Interfaces
  - ☐ OPC Data Access Custom Interfaces
  - ☑ Server Functionality
    - ☑ OPC_Browsing
    - ☑ OPC_CheckServer
    - ☑ OPC_ItemProperties
    - ☑ OPC_Server

OPC Server
Computername: RSCHRADE
Res-ID: 123

Tracelevel
○ Debug
○ Information
○ Message
○ Warning
○ Error
● No Trace

Tracefile
Path: c:\f2k_V71\opcsrv
Size (KByte): 1024

op500gr.bmp

OPC Server

Computer name, Res-ID

The linked OPC server is described by means of the two entries. The two fields can be edited after actuation of the **Disconnect** button. A connection to the registered OPC server is established by means of the **Connect** button.

Keyword trace The various functions of the OPC server can be hierarchically selected in the left-hand half of the dialog. Depending on the entries selected, the respective function calls in the OPC server are recorded together with the current parameters.
As a result, it is possible to track specifically defined functions or functional groups of the OPC server. See Appendix C, OPC keyword tracing.

Trace level The degree of detail of the trace outputs is specified by means of the trace level.

Trace file

Path The directory for the trace file is specified. A file named **opc<Res-ID>.trc** is created in the specified directory.

Size [KByte] The maximum size of the trace file in KBytes. If the size of the trace file reaches maximum, the file is renamed **opc<Res-ID>.bak** and a new **opc<Res-ID>.trc** file is created.

Apply Settings The settings for directory and size of the trace file are transferred.

Show File The current trace file is displayed in the standard Windows editor. If information from the *.bak file is required, the latter has to be called separately.

Clear File The current trace file is deleted. the *.bak file is not affected by this action.

Server dump

Dump Configuration

All the configuration data of the OPC server is written to the trace file. The entries are marked with the text **Dumping configuration data...** and **...configuration dump finished**.

Dump Connections

An entry is generated in the trace file for all active connections. The entries are marked with the text **Dumping server objects...** and **...server object dump finished**.

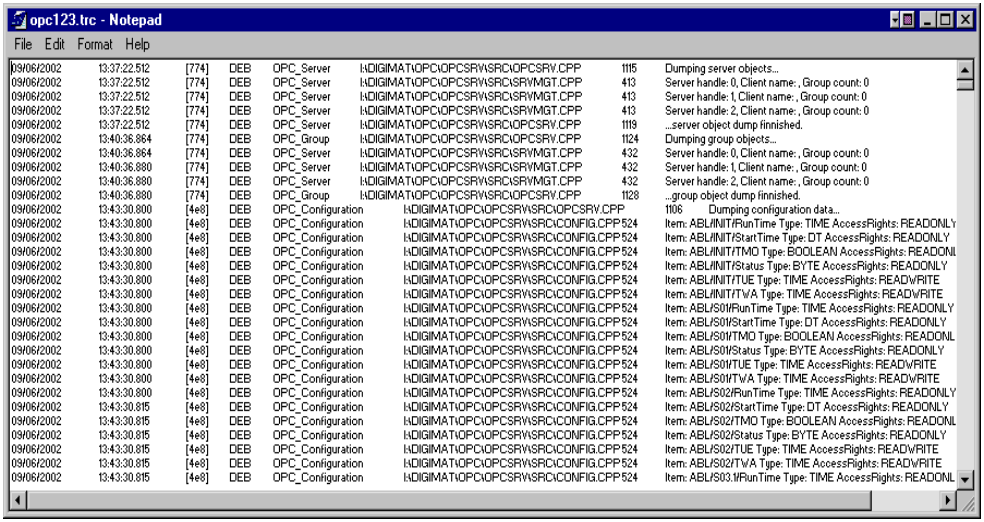
Dump Groups

An entry is generated in the trace file for all installed groups of the

OPC server. The entries are marked with the text **Dumping group objects...** and **...group object dump finished**.

Save Settings All settings of the dialog are stored and preset on the start of a new trace dialog or restart of the OPC server.

Example of a trace file



```
09/06/2002  13:37:22.512  [774]  DEB  OPC_Server         I:\DIGIMAT\OPC\OPCSRV\SRC\OPCSRV.CPP      1115  Dumping server objects...
09/06/2002  13:37:22.512  [774]  DEB  OPC_Server         I:\DIGIMAT\OPC\OPCSRV\SRC\SRVMGT.CPP      413   Server handle: 0, Client name: , Group count: 0
09/06/2002  13:37:22.512  [774]  DEB  OPC_Server         I:\DIGIMAT\OPC\OPCSRV\SRC\SRVMGT.CPP      413   Server handle: 1, Client name: , Group count: 0
09/06/2002  13:37:22.512  [774]  DEB  OPC_Server         I:\DIGIMAT\OPC\OPCSRV\SRC\SRVMGT.CPP      413   Server handle: 2, Client name: , Group count: 0
09/06/2002  13:37:22.512  [774]  DEB  OPC_Server         I:\DIGIMAT\OPC\OPCSRV\SRC\OPCSRV.CPP      1119  ...server object dump finnished.
09/06/2002  13:40:36.864  [774]  DEB  OPC_Group          I:\DIGIMAT\OPC\OPCSRV\SRC\OPCSRV.CPP      1124  Dumping group objects...
09/06/2002  13:40:36.864  [774]  DEB  OPC_Server         I:\DIGIMAT\OPC\OPCSRV\SRC\SRVMGT.CPP      432   Server handle: 0, Client name: , Group count: 0
09/06/2002  13:40:36.880  [774]  DEB  OPC_Server         I:\DIGIMAT\OPC\OPCSRV\SRC\SRVMGT.CPP      432   Server handle: 1, Client name: , Group count: 0
09/06/2002  13:40:36.880  [774]  DEB  OPC_Server         I:\DIGIMAT\OPC\OPCSRV\SRC\SRVMGT.CPP      432   Server handle: 2, Client name: , Group count: 0
09/06/2002  13:40:36.880  [774]  DEB  OPC_Group          I:\DIGIMAT\OPC\OPCSRV\SRC\OPCSRV.CPP      1128  ...group object dump finnished.
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\OPCSRV.CPP      1106      Dumping configuration data...
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/INIT/RunTime Type: TIME AccessRights: READONLY
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/INIT/StartTime Type: DT AccessRights: READONLY
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/INIT/TMO Type: BOOLEAN AccessRights: READONL
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/INIT/Status Type: BYTE AccessRights: READONLY
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/INIT/TUE Type: TIME AccessRights: READWRITE
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/INIT/TWA Type: TIME AccessRights: READWRITE
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S01/RunTime Type: TIME AccessRights: READONLY
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S01/StartTime Type: DT AccessRights: READONLY
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S01/TMO Type: BOOLEAN AccessRights: READONL
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S01/Status Type: BYTE AccessRights: READONLY
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S01/TUE Type: TIME AccessRights: READWRITE
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S01/TWA Type: TIME AccessRights: READWRITE
09/06/2002  13:43:30.800  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S02/RunTime Type: TIME AccessRights: READONLY
09/06/2002  13:43:30.815  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S02/StartTime Type: DT AccessRights: READONLY
09/06/2002  13:43:30.815  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S02/TMO Type: BOOLEAN AccessRights: READONL
09/06/2002  13:43:30.815  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S02/Status Type: BYTE AccessRights: READONLY
09/06/2002  13:43:30.815  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S02/TUE Type: TIME AccessRights: READWRITE
09/06/2002  13:43:30.815  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S02/TWA Type: TIME AccessRights: READWRITE
09/06/2002  13:43:30.815  [4e8]  DEB  OPC_Configuration      I:\DIGIMAT\OPC\OPCSRV\SRC\CONFIG.CPP524   Item: ABL/S03.1/RunTime Type: TIME AccessRights: READONL
```

op028us.bmp

2.8.3 Testing in Freelance Engineering

During commissioning, like the process station variables, the OPC variables can also be displayed in the value window and trend window of Freelance Engineering.

2.8.4 Performance monitor for OPC server

A performance monitor included in the Windows standard package can be used to check and monitor the performance of the Windows PC. The software package from Microsoft allows all general variables, e.g. free disk space, CPU workload to be displayed and monitored.

An add-on component also allows **application-specific Freelance process station variables**, such as statistical data on read/write errors, to be displayed on the OPC server.

Preparing the Windows operating system

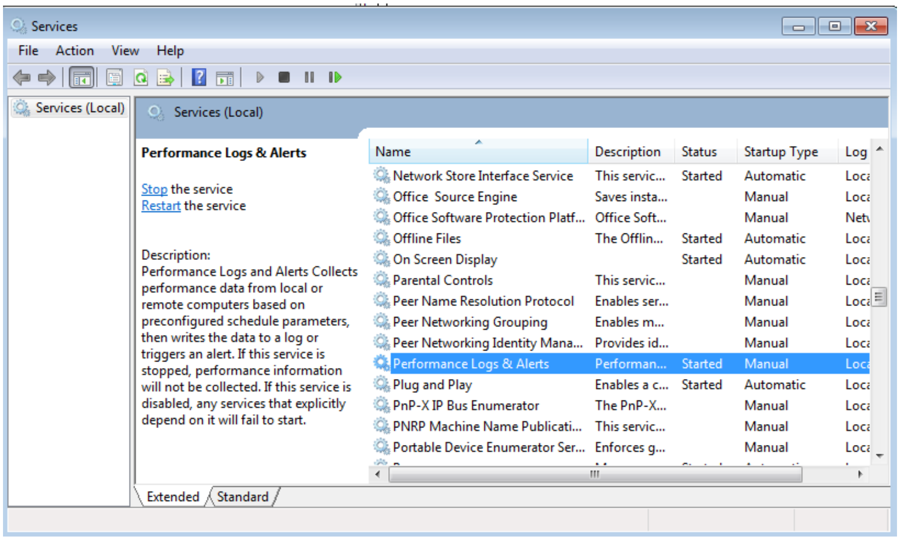
Make sure that the Freelance OPC server is running.

Start the Windows services.

Control Panel > Administrative Tools > Services

or

type `services.msc` in the search field of the Windows start menu.



Performance Logs & Alerts

Stop the service
Restart the service

Description:
Performance Logs and Alerts Collects performance data from local or remote computers based on preconfigured schedule parameters, then writes the data to a log or triggers an alert. If this service is stopped, performance information will not be collected. If this service is disabled, any services that explicitly depend on it will fail to start.

Name	Description	Status	Startup Type	Log
Network Store Interface Service	This servic...	Started	Automatic	Loca
Office Source Engine	Saves insta...		Manual	Loca
Office Software Protection Platf...	Office Soft...		Manual	Netw
Offline Files	The Offlin...	Started	Automatic	Loca
On Screen Display		Started	Automatic	Loca
Parental Controls	This servic...		Manual	Loca
Peer Name Resolution Protocol	Enables ser...		Manual	Loca
Peer Networking Grouping	Enables m...		Manual	Loca
Peer Networking Identity Mana...	Provides id...		Manual	Loca
Performance Logs & Alerts	Performan...	Started	Manual	Loca
Plug and Play	Enables a c...	Started	Automatic	Loca
PnP-X IP Bus Enumerator	The PnP-X...		Manual	Loca
PNRP Machine Name Publicati...	This servic...		Manual	Loca
Portable Device Enumerator Ser...	Enforces g...		Manual	Loca

PerfMon1_us.png

Select **Performance Logs & Alerts** and start this service.

Starting the performance monitor in Windows 7/1032 bit

Control Panel > Administrative Tools > Performance Monitor

or

type `perfmon.exe` or `perfmon.msc` in the search field of the Windows start menu.

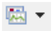
Starting the performance monitor in Windows 7/10 64 bit

Type `c:\windows\syswow64\perfmon.exe` in the search field of the Windows start menu.

Configuring the Performance Monitor

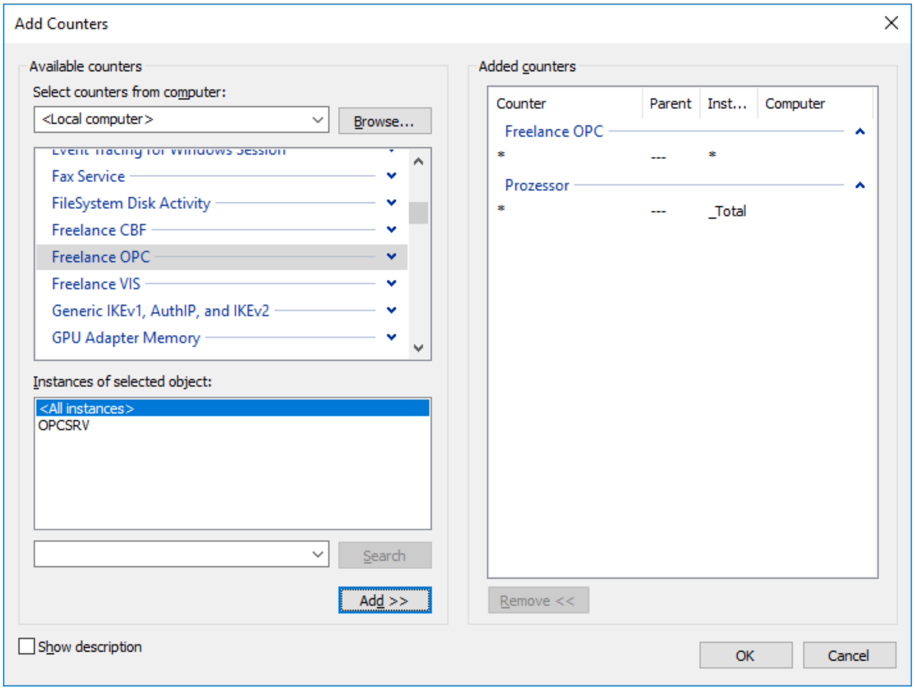
Select in tree in the left part of the dialog: **Performance / Monitoring Tools / Performance Monitor**.

In the right part of the dialog a diagram is shown. The operation is mainly done with the buttons in the tool bar:



Select the symbol *Change graph type* from the tool bar to select another view for the collected data.

Select the green + sign from the tool bar to add counters to be observed. In the next dialog, browse the computer for available counters. Select first the Freelance component, e.g. **Freelance OPC**. In any cases select **<All instances>** and press the button **Add >>** to select the counters.



PerfMon5_us_1.png

If required, process counters supplied by Windows can be added. When the counter specification is complete, close the dialog with button **OK**.

The display parameters can be set through the dialog **Performance Monitor Properties**.

Meaning of the variables

General variables

Name	Description
% Communication buffers	% Communication buffers is the percent amount of “in use” communication buffers. Normally this value is 0.
Communication buffers	Total amount of “in use” communication buffers. Normally this value is 0.
% Connections	% Connections is the percent amount of active connections.
Connections	Total number of active connections.
% DMS-Handles	% DMS-Handles is the percent amount of “in use” DMS-Resources. These value includes the dynamic SEG-Handles.
DMS-Handles	Total amount of “in use” DMS-Resources. These value includes all dynamic SEG-Handles.
% SEG-Handles	% SEG-Handles is the percent amount of “in-use” DMS-Resources for cyclic requests.
SEG-Handles	Total amount of “in-use” DMS-Resources for cyclic requests.
Process-ID	PID of the running process.
Station-Number	First station number

OPC variables

Name	Description
Item No	Number of configured items.
Event No	Number of configured event points.
Client connections	Number of OPC Data Access client connections.

Name	**Description**
Groups	Number of OPC Groups.
Active items	Number of active OPC Items, the same item in different groups is counted once.
AddItem failures (bandwidth)	Number of OPC Items which could not added due to limited bandwidth.
AddItem failures (interaction)	Number of OPC Items which could not added due to wrong interaction between client and server.
AddItem failures (configuration)	Number of OPC Items which could not added due to configuration errors.
Pending jobs	Number of pending acyclic jobs.
Acyclic read errors	Number of failed acyclic reads, counted on item base.
Acyclic write errors.	Number of failed writes, counted on item base
Station No. for Station n	Station No. for Station n.
% used cyclic bandwidth St. N	Percentage of used cyclic bandwidth for Station n. If this value reaches 100% it is not possible to communicate additional cyclic values from this Station. This could also happen due to fragmentation when the value is a little bit below 100%.
Changed cyclic items/second	Number of changed cyclic items per second, items sent to different clients are counted once.
Acyclic reads/second	Number of successfully acyclic read items from device per second.
Acyclic writes/second	Number of successfully written items per second.

3 OPC client

OPC client functionality is implemented both in Freelance Engineering and Freelance Operations. There is no need for any special setup.

During the configuration process in **Freelance Engineering**, the variable lists from external OPC servers can be read through the OPC browser interface. The variables that are to be displayed or logged in the operator stations are selected from a list and imported into the current project. This data can then be used in the configuration of graphic displays, trend displays and logs.

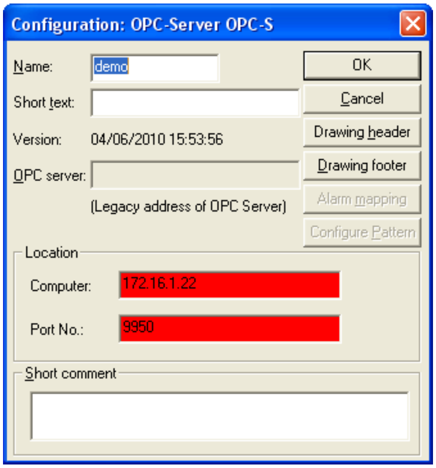
When an operator station is loaded, all the necessary data is transferred. The variables from the external OPC servers are read and, if necessary, also written through the OPC client interface that forms an integral part of Freelance Operations.

3.1 Integrating an external OPC server

3.1.1Adding the OPC server function to the project tree

A new resource of type OPC server is created in the Project tree. The location of the external OPC server with computer name or IP address and used port number is entered in the associated parameter dialog.

> **Edit** > **Insert next level** > Select **OPC server** > **OK**



Configuration: OPC-Server OPC-S

Name: demo

Short text:

Version: 04/06/2010 15:53:56

OPC server:

(Legacy address of OPC Server)

Location

Computer: 172.16.1.22

Port No.: 9950

Short comment

op007us

Name Name of the OPC Server-resource in the project tree

Short text Max. 12 characters

Version Date and time of object creation

Location Specifies the computer on which the OPC software is installed

Computer Name or IP address of the computer where OPC Server is running

Port No used port for the OPC server

Short comment Max. 159 characters

Drawing header / Drawing footer
Refer to ***Engineering Manual System Configuration, Documentation***.

For further details, refer to ***Engineering Manual System Configuration, Project tree***.

3.1.2 Importing OPC variables

The OPC variables (OPC Items) can be imported from the OPC server resource after it is configured in the project tree and can then be used for configuration in the operator station. These variables are associated to the OPC server resource and not to any Process station.

Refer to ***Engineering-Manual IEC 61131-3 Programming, OPC items***.

3.1.3 Enabling for OPC access

With external OPC servers, enabling for read and/or write access is performed in the external systems.

3.1.4 Initiating communication

An external OPC server should be already started, or able to be started automatically when requested by an OPC client.

3.1.5 Example for an OPC link between two systems

Example scenario: Data from an external system called “System B” are to be imported into a Freelance system called “System A”

The external system can be another Freelance system or a third party system.

Preparing the external system called **System B**:

1. In System B, an OPC gateway is configured.
2. On the PC that is to be used as the gateway for System B, the OPC server is installed.
3. On the PC, on which the OPC server for System B is run, the ABB OPC tunnel is installed and configured. Even if System B is no Freelance system, the tunnel must be installed and configured on the OPC server PC.
4. The OPC server is loaded with the correspondingly configured data from System B. If this is a Freelance system, the OPC gateway station is loaded from the Project tree.

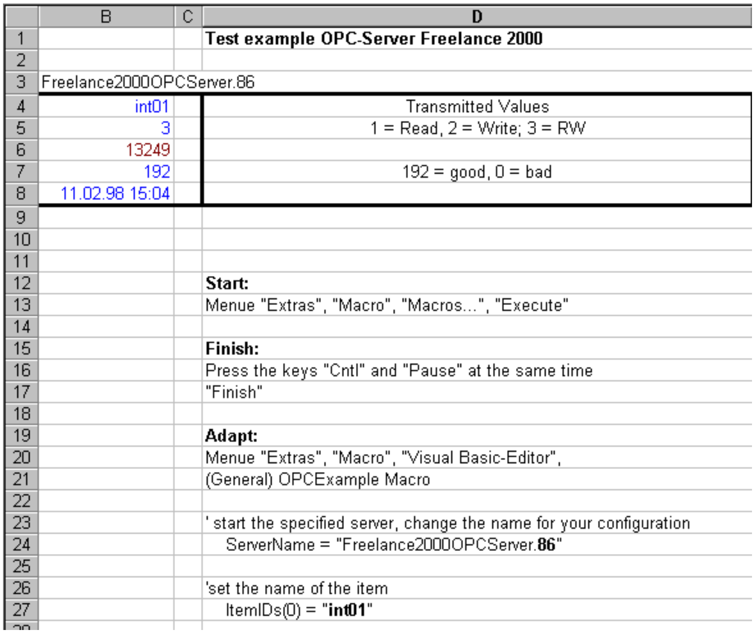
Configuring the Freelance system called **System A**:

1. In System A, the OPC server from System B is configured as an external OPC server. For this purpose, an OPC server object is created in the Freelance project tree. In the parameter dialog, the computer name and port number from the tunnel configuration of System B are entered.
2. Once the OPC server from System B has been started, the data of this external OPC server can be imported through the Browse interface in into the project of System A (Freelance).
3. The imported variables are used in the configuration.
4. The Freelance operator stations of System A are loaded.
5. Depending on the started logs and/or displays on the operator stations of System A, the required variables are read from and written to the OPC server of System B.

Appendix A Example of OPC access from MS-Excel and MS-Visual Basic

A.1 Overview of the Excel example

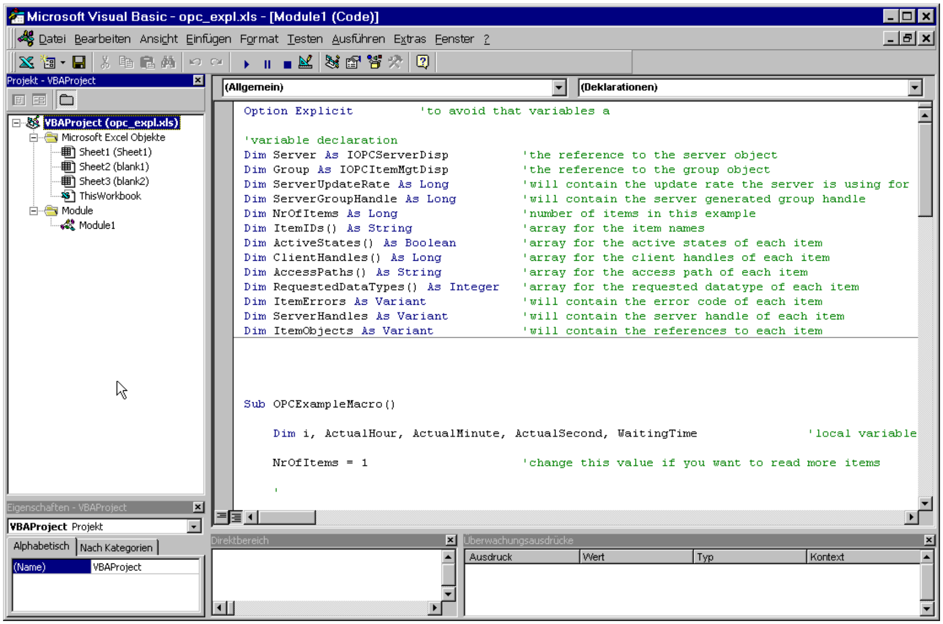
Overview of the example file “opc_expl.xls”:



	B	C	D
1			**Test example OPC-Server Freelance 2000**
2			
3	Freelance2000OPCServer.86		
4	int01		Transmitted Values
5	3		1 = Read, 2 = Write; 3 = RW
6	13249		
7	192		192 = good, 0 = bad
8	11.02.98 15:04		
9			
10			
11			
12			**Start:**
13			Menue "Extras", "Macro", "Macros...", "Execute"
14			
15			**Finish:**
16			Press the keys "Cntl" and "Pause" at the same time
17			"Finish"
18			
19			**Adapt:**
20			Menue "Extras", "Macro", "Visual Basic-Editor",
21			(General) OPCExample Macro
22			
23			' start the specified server, change the name for your configuration
24			ServerName = "Freelance2000OPCServer.**86**"
25			
26			'set the name of the item
27			ItemIDs(0) = "**int01**"

op200us.bmp

A.1.1 Program overview



```
Option Explicit         'to avoid that variables a

'variable declaration
Dim Server As IOPCServerDisp            'the reference to the server object
Dim Group As IOPCItemMgtDisp            'the reference to the group object
Dim ServerUpdateRate As Long            'will contain the update rate the server is using for
Dim ServerGroupHandle As Long           'will contain the server generated group handle
Dim NrOfItems As Long                   'number of items in this example
Dim ItemIDs() As String                 'array for the item names
Dim ActiveStates() As Boolean           'array for the active states of each item
Dim ClientHandles() As Long             'array for the client handles of each item
Dim AccessPaths() As String             'array for the access path of each item
Dim RequestedDataTypes() As Integer     'array for the requested datatype of each item
Dim ItemErrors As Variant               'will contain the error code of each item
Dim ServerHandles As Variant            'will contain the server handle of each item
Dim ItemObjects As Variant              'will contain the references to each item



Sub OPCExampleMacro()

    Dim i, ActualHour, ActualMinute, ActualSecond, WaitingTime              'local variable

    NrOfItems = 1                       'change this value if you want to read more items

    '
```

op0201us.bmp

A.1.2 Main program

Main program in Visual Basic (in this case in “Module1”):

The library/class “OPCDisp” must be loaded beforehand through **Tools** > **Reference** so that “IOPCServerDisp” is known.

```
Attribute VB_Name = "Module1"

Option Explicit 'to avoid that variables automatically
'variable declaration

Dim Server As IOPCServerDisp 'the reference to the server
object
Dim ServerName As String' name of the server'
Dim Group As IOPCItemMgtDisp'the reference to the group
object
```

```
Dim ServerUpdateRate As Long'contains update rate, server is
using for groups
Dim ServerGroupHandle As Long 'will contain the server
generated group handle
Dim NrOfItems As Long 'number of items in this example
Dim ItemIDs() As String 'array for the item names
Dim ActiveStates() As Boolean 'array for the active states of
each item
Dim ClientHandles() As Long 'array for the client handles of
each item
Dim AccessPaths() As String 'array for the access path of each
item
Dim RequestedDataTypes() As Integer'array for the requested
datatype of each item
Dim ItemErrors As Variant 'will contain the error code of each
item
Dim ServerHandles As Variant'will contain the server handle
of each item
Dim ItemObjects As Variant 'will contain the references to
each item

Sub OPCExampleMacro()

Dim i, ActualHour, ActualMinute, ActualSecond, WaitingTime
'local variables

NrOfItems = 1'change this value if you want to read more items
'
'resize all arrays to the necessary size
'
ReDim ItemIDs(NrOfItems)
ReDim ActiveStates(NrOfItems)
ReDim ClientHandles(NrOfItems)
ReDim AccessPaths(NrOfItems)
ReDim RequestedDataTypes(NrOfItems)
'
' start the specified server, change the name for your
configuration
'
ServerName = " ... "‘ that is "Freelance2000OPCServer.86")
```

```
Set Server = CreateObject("ServerName")
'
' add a group to the server
'
Set Group = Server.AddGroup( _
  "Group 1", _
  True, _
  1000, _
  1, _
  0, _
  &H409, _
  ServerGroupHandle, _
  ServerUpdateRate)
'
'set the name of the item
'
ItemIDs(0) = "int01"'change the name for your configuration
'
'set client handles and active state of the items
'
For i = 0 To (NrOfItems - 1)
  ClientHandles(i) = i + 1'use a different client handle for
each item
  ActiveStates(i) = True'all items are active
Next
'
'add items to the group
Group.AddItems _
  NrOfItems, _
  ItemIDs, _
  ActiveStates, _
  ClientHandles, _
  ServerHandles, _
  ItemErrors, _
  ItemObjects, _
  AccessPaths, _
  RequestedDataTypes

'as long as CTRL-BREAK is not pressed update every 2 seconds
the sheet
```

```
'
Worksheets(1).Visible = True
While True
  ActualHour = Hour(Now())
  ActualMinute = Minute(Now())
  ActualSecond = Second(Now()) + 2 'change 2 to x if you want
faster execution
  WaitingTime = TimeSerial(ActualHour, ActualMinute,
ActualSecond)
  Application.Wait WaitingTime

  For i = 0 To (NrOfItems - 1)
    Worksheets(1).Cells(1 + 2, i + 1) = "Server:"
    Worksheets(1).Cells(1 + 2, i + 2) = ServerName
    Worksheets(1).Cells(1 + 3, i + 1) = "Name:"
    Worksheets(1).Cells(1 + 3, i + 2) = ItemObjects(i).ItemID
    Worksheets(1).Cells(2 + 3, i + 1) = "Access Rights:"
    Worksheets(1).Cells(2 + 3, i + 2) =
ItemObjects(i).AccessRights
    Worksheets(1).Cells(3 + 3, i + 1) = "Value:"
    Worksheets(1).Cells(3 + 3, i + 2) = ItemObjects(i).Value
    Worksheets(1).Cells(4 + 3, i + 1) = "Quality:"
    Worksheets(1).Cells(4 + 3, i + 2) = ItemObjects(i).Quality
    Worksheets(1).Cells(5 + 3, i + 1) = "Timestamp:"
    Worksheets(1).Cells(5 + 3, i + 2) =
ItemObjects(i).Timestamp
  Next
Wend
End Sub
```

Appendix B Representation of the Freelance message types

The process messages of a Freelance system are subdivided into various **categories**. The **condition** and **subcondition** are always identical for a process alarm.

The index of the initiating message point is added to the condition/subcondition strings.

Freelance message type	**CATEGORY “OPC_CATEGORY_xxx”**	**CONDITION = SUBCONDITION (String)**
No configuration	UNKNOWN	UNKNOWN
H	LEVEL	H
L		L
HH		HH
LL		LL
H_V	LEVEL	H
L_V		L
HH_V		HH
LL_V		LL
DHs, DHm, DHh	RATE_OF_CHANGE	DH
DLs, DLm, DLh		DL
DHHs, DHHm, DHHh		DHH
DLLs, DLLm, DLLh		DLL

Freelance message type	**CATEGORY “OPC_CATEGORY_xxx”**	**CONDITION = SUBCONDITION (String)**
H_XD L_XD HH_XD LL_XD LH_XD	DEVIATION	H_CE L_CE HH_CE LL_CE LH_CE
FLT_TIME	DURATION	FLT_TIME
END_POS	DISCRETE	END_POS
ERROR	DISCRETE	ERROR
PROTECT	DISCRETE	PROTECT
LOC_OP	DISCRETE	LOC_OP
MAN/AUTO	DISCRETE	MAN/AUTO
TORQUE	DISCRETE	TORQUE
TIME	DISCRETE	TIME
BINARY	DISCRETE	BINARY
LH	LEVEL	LH
USER	USERDEFINED	USER
TIME_OV	DURATION	TIME_OV
NOISE	LIMIT	NOISE
STAT	LIMIT	STAT
MOVE	LIMIT	MOVE
Y_LIMIT	LIMIT	PIV_LIMIT
TASK_OVERLOAD	LIMIT	TASK_OVERLOAD
MSR_SFC_TUE	DURATION	SFC_TUE

Appendix C OPC keyword tracing

C.1 Structure of the trace entries

Each entry of the trace file contains the following information:

- Current date and time of the trace entry
- Internal number - not relevant for the user
- Trace level (depending on the selected trace level more (Debug) or less (No trace) entries are generated in the trace file)
- Keyword, that caused this entry
- Name of the source file - not relevant for the user
- Internal number - not relevant for the user
- Function with parameters

Example:

`05/12/2003`	Date
`14:56:15.853`	Time
`[764]`	Internal number
`DEB`	Trace level = Debug
`OPC_DALCyclicNotification`	Keyword
`I:\DIGIMAT\OPC\OPCSRV\SRC\OPCSRV.CPP`	File name
`1115`	Internal number
`ValueChangedCB(): ShelfID: 2, Magic: 15, Value: 820, TimeStamp: 05/12/2003 13:56:58.705, ProtocolQuality: 0,0x0`	Function with parameters; in this case a value has changed to value 820.

C.1.1 Trace of keyword selection

Selection or de-selection of a keyword is traced with

Keyword <name> switched on or

Keyword <name> switched off.

With <name> = keyword, as shown in the tree structure.

It is not necessary to start or stop the trace functionality. Immediately after selection or de-selection of a keyword the accompanying functions are traced or the trace is stopped.

C.1.2 Traced functions

With the section of the keywords the different functions of the OPC server can be traced. All keywords and the accompanying functions are listed below.

Keyword: Alarms & Events

OPC_AE: Creating and deleting of alarm tags
Creating of alarm conditions
Activating of the alarm lists

OPC_DMSEvent:
Update of alarm conditions
Acknowledgment handling
Functionality of the resource table

OPC_TKEvent:
not used

OPC_TKInternal:
not used

Keyword: Common Functionality

OPC_Common:
General functionality like
* Initialization of logs
* Import of the configuration data
* Generation of event log entries

OPC_Configuration:
Changes of OPC server configuration after download

OPC_Shelf: Items, that are read from a client, are located in the shelf.
All operations at this shelf, e.g:
* Add
* Remove
* Clean Up
* Value Changed
* Set Value

Keyword: DAL

The Data Acquisition Layer DAL is the protocol layer within the OPC server which is positioned above the different field bus protocols and the controller protocol (DMS).

DAL Notification:

- OPC_DALCyclicNotification:
  Cyclic information to value changes (ValueChanged Callback)
- OPC_DALReadNotification:
  Log of asynchronous Read Callback
- OPC_DALWriteNotification:
  Log of asynchronous Write Callback

OPC_DALCyclicRead:
Adding and removing of OPC items.

OPC_DALJob: All operations which are called during a Job:
* Start/Finish Job
* Read/Write Job
* Sync/Async Read/Write

Keyword: DMS

The DMS protocol is used for communication from OPC server to the process stations.

OPC_DMSACyclic:
Acyclic read and write operations

OPC_DMSCommon:
Information about the state of the process stations

OPC_DMSCyclicRead:
Cyclic read and write actions, and connecting and de-connecting of process stations and variables

Keyword: Fieldbus

Fieldbus protocols (PROFIBUS, HART) are used for communication from OPC server to the field devices.

OPC_FBACyclic:
Asynchronous read and write at the fieldbus

OPC_FBCommon:
Mapping of error numbers from PROFIBUS or HART communication into OPC error numbers

OPC_FBCyclicRead:
Connecting and de-connecting of cyclic fieldbus variables

Keyword: Group Functionality

OPC_AsyncNotification:
Notifications of the group, like read and write of the cache, refresh and append

OPC_Group: Operations with the OPC groups, e.g.:
* Add item
* Remove item
* Refresh
* Read action finished (on read complete)
* Write action finished (on write complete)
* Configuration changes

Keyword: Item Functionality

OPC_Item: Configuration change of an item
Activation and de-activation of an item
Status change of a superior object

Keyword: OPC Data Access Automation 1.0 Interfaces

These keywords are used for clients which support OPC Specification Version 1.0 only.

Group Object (Automation 1.0):

- IOPCAsyncIODisp:
  Adding and deleting of callbacks
  Asynchronous read and write
  Refresh of the OPC groups
- IOPCGroupStateMgtDisp:
  Status information about the OPC groups
- IOPCItemMgtDisp:
  Adding and deleting items into a group
  Setting the data type
  Activating and de-activating of items and the accompanying enumerators
- IOPCSyncIODisp:
  Synchronous Read and Write at OPC groups

Item Object (Automation 1.0):

- IOPCItemDisp:
  Operation at OPC items, e.g.:
  * Read and Write values (get/put Value)
  * Read access rights (get AccessRights)
  * Read access path (get AccessPath)
  * Read internal association number (GetItemID)

ServerObject (Automation 1.0):

- IOPCBrowseServerAdressSpaceDisp:
  Browsing operations
  all OPC items, that are accessed with the browse operation
- IOPCServerDisp:
  All operations that are performed at the server, like e.g.:
  Adding and removing groups
  State of the OPC server

Keyword: OPC Data Access Custom Interfaces

Client Interfaces:

- IAdviseSink:
  Value change within a group. (Function of OPC Spec 1.0)
- IOPCDataCallback:
  Finished read and write operations
  Value changes
- IOPCShutdown:
  not used

GroupObject:

- IDataObject:
  Adding and removing groups
- IEnumOPCAttributes:
  All operations of item enumerator
- IOPCAsyncIO:
  Asynchronous read and write of groups
  Refresh and cancel of a group (following OPC Spec Version 1.0)
- IOPCAsyncIO2:
  Asynchronous read and write of groups
  Refresh and cancel of a group
- IOPCGroupeStateMgt:
  Renaming od groups
  Status of groups
  Multiplying of groups

- IOPCItemMgt:
  Adding and removing items within a group
  setting the data type
  Activating and de-activating of items and the accompanying enumerators

- IOPCAsyncIO:
  Synchronous read and write to groups

IConnectionPointContainer:
Adding and removing of groups

ServerObject:

- IOPCBrowseServerAdressSpace:
  All operations that are used during browsing the configuration of an OPC server, e.g.:
  * Change of internal position (ChangeBrowsePosition)
  * Internal assignments of the OPC items (BrowseOPCItemIds)

- IOPCCommon:
  Information about available OPC server
  Setting of the client name

- IOPCItemProperties:
  All operations that are done with the OPC item properties, e.g.:
  * QueryAvailableProperities
  * GetItemProperties
  * LookUpItemIDs

- IOPCServer:
  All operations that are carried out at the server, like e.g.:
  * Adding and removing groups
  * Status of the OPC server

Keyword: Server Functionality

OPC_Browsing:
Information about the browsing enumerator

OPC_CheckServer:
not used

OPC_ItemProperties:
Handling of the item properties:
* GetItemProperties
* IsItemValid
* GetConfProperties

OPC_Server: Information about the OPC server state:
* CheckServer
* GetNumberOfObjects
* GetLockStateOfGroup
* OPCServerManagement
* GetStart/CurrentTime

C.1.3 Examples of use

Which values are sent from OPC server to OPC client?

With selecting **Common Functionality / OPC_Shelf** value changes are logged in this way:

```
COPCItemShelf::ValueChanged() Item: TIC1623gr/SP Index: 1
Magic: 7 Value: 918 Time stamp: 05/09/2003 10:42:47.310 rc:
0x0
```

Item = name of the variable (OPC item name)
Index = internal index
Magic = internal assignment number
Value = value of variable
Time = time stamp of value change (Local time)

In this example the variable ’TIC1623gr/SP’ has changed to value 918.

With selecting **DAL / DAL Notification / OPC_DALCyclicNotification** value changes are logged in this way:

```
ValueChangedCB(): ShelfID: 1, Magic: 15, Value: 720,
TimeStamp: 05/12/2003 13:56:58.705, ProtocolQuality: 0,0x0
```

In this example a variable has changed to value 720.

Which OPC items belong to the OPC groups?

Building of the OPC groups is traced with the keyword selection **Group Functionality / OPC_Group**:

```
COPCGroupRoot::COPCGroupRoot() called (actual number=1)
```

For each variable that is added into a group this entry is generated:

```
COPCGroupRoot::InternalAddItems(): add item 'F705/SP' of
canonical type VT_R4, requested type VT_R4 to group 'G3'
successful, client handle: 0
```

Each sending action to the client is traced with:

```
COPCGrouppCust::SendCust() for group G3 called
```

Which variables are requested by the client?

With the keyword selection **Common Functionality / OPC_Shelf** new client requests are traced.

For each variable to remove:

```
COPCItemShelf::Remove(0) called
```

```
COPCItemShelf::Remove(): UnadviseVariable: remove Item
F705/SP at index 0
```

For each variable to add:

```
COPCItemShelf::Add(F705/SP) called
```

```
COPCItemShelf::Add(): AdviseVariable() for item 'F705/SP'
returns 0x0,0 Index: 0
```

```
COPCItemShelf::SetValue() Item: F705/SP Index: 0 Value: 0 Time
stamp: 05/09/2003 10:47:20.150 rc: 0x0
```

Which client connections exist to the OPC server and how long do these connections exist?

With the keyword selection **OPC Data Access Custom Interface / Server Object / IOPCServer** these three messages are generated each second for each client connection:

```
GetStatus() called
GetStatus() succeeded, return S_OK
```

```
GetStatus(ftStartTime=05/09/2003 09:45:57.348,
ftCurrentTime=05/09/2003 10:04:00.053,
ftLastUpdateTime=05/09/2003 10:03:40.178, dwServerState=0x1,
dwGroupCount=1, dwBandWidth=0, wMajorVersion=7,
wMinorVersion=1, wBuildNumber=2494, szVendorInfo=Freelance
2000 DigiOpc - V7.1 BETA Station 123) returns S_OK
```

StartTime = Start time of this connection (When was the connection to the client established?)

CurrentTime = current time

LastUpdateTime = time stamp, when the last data communication was done.

All time stamps are generated with GMT time.

Appendix D OPC server – System messages

Pr	Priority level S1, S2 or S3
MR	Message rate: 1 = Message with one state, e.g. ‘Redundancy toggle occurred’ 2 = Message with two states, e.g. first state ‘Battery low’, second state ‘Battery low is gone’
[...]	The text in the brackets is not displayed in every case.

Message text	Pr	MR	Cause of message	Remedy
Battery low [(IP1/2)]	S2	2	CPU module battery is flat or disconnected.	Replace or connect battery.
Battery low in CPU module	S2	2	CPU module battery is flat or disconnected.	Replace or connect battery.
Battery low in link module	S2	2	Link module battery is flat or disconnected.	Replace or connect battery.
Battery low in module E1/E2 [(IP1/2)]	S2	2	Battery of the module is flat or disconnected.	Replace or connect battery.
Channel xxx, edge detection	S1	2	In a DDI01 a change of edge of the input signal was detected from channel xxx and reported with the accompanying time stamp. The report is written only in the record and not in the report page.	

Message text	Pr	MR	Cause of message	Remedy
Counter overrun of channel ’xxxx’	S2	2	The internal counter in the frequency input module has overrun.	Contact your authorized service engineer.
Counter overrun of channel 'xxxx'	S2	2	The internal counter in the frequency input module has overrun.	Contact your authorized service engineer.
Data lost for all modules of process bus 1 [to 3]	S2	1	Data may have been lost by all subscribers as a result of a CAN bus error.	Check the terminating resistors and the cable lengths or contact your authorized service engineer.
DDI01 module not exist	S2	2	Internal error when invoking the module method.	Check the configuration or contact your service technician.
Diagnosis fault for module n l m	S3	2	Diagnosis faults were reported for the specified module.	Check the associated PROFIBUS module or contact the slave manufacturer’s service technician.
Diagnosis fault for unit	S2	2	A PROFIBUS unit has sent diagnosis faults.	Check the connected PROFIBUS units or contact your authorized service engineer.
EN-Wire break of channel ’xxxx’	S3	2	The cable connection for the control input EN (Enable) is faulty.	Check the relevant process signal cable.
Extended diagnostic data overflow	S3	2	A PROFIBUS slave has given an overflow of diagnostic data. Probably a ‘flutter’ signal.	Check the PROFIBUS slave or contact slave manufacturer’s service technician.
Fault in diagnosis data	S3	1	The module has announced the unknown diagnostic data.	Check the effected I/O module or contact your service
Fault in output channel ’xxxx’	S2	2	There is a short circuit on output channel ’xxxx’.	Check the process signal connection, eliminate short-circuit.

Message text	Pr	MR	Cause of message	Remedy
FF configuration identity mismatch	S2	2	The configuration that is to be loaded into the Foundation Fieldbus devices is not suitable for the devices which are physically present.	Adapt the configuration in Freelance Engineering according to the physical conditions.
FF cycle overrun	S2	2	The configured HSE bus cycle time was exceeded.	Change the FF configuration, e.g. increment the bus cycle time.
FF data exchange	S2	2	The communication of FF data has been reestablished.	
FF HSE initialization error	S2	2	An internal error occurred during initialization of the HSE protocol function block.	
FF PNA error	S2	2	An internal software error has occurred in one of the Foundation Fieldbus devices	Check the fieldbus devices and replace the defective device.
FF protocol error	S2	2	An error has been detected in the communication protocol of the Foundation Fieldbus line.	
FF state not ready	S2	2	Not possible to communicate FF data.	Check the hardware installation and the configuration.
FI840 IP [1/2] link broken	S2	2	Communication disruption on the Ethernet link of the FI840 module, possibly a faulty cab-le, or cable not plugged in.	Check the Ethernet cable.
FI840 IP [1/2] link running	S2	2	Ethernet connection of FI840 module established.	
First adjustment not finished	S1	2	An internal error has occurred on the secondary CPU during data balancing between the CPU modules.	Contact your authorized service engineer.

Message text	Pr	MR	Cause of message	Remedy
Frame-Error of channel ’xxxx’	S2	2	An error has occurred in the measurement timing on the channels indicated.	The time sequence of input signals at the frequency input module DFI 01 should be checked, depending on operating mode.
High temperature in CPU [IP1/2]	S2	2	The temperature of the CPU module is more than 70 °C.	The ambient temperature is probably more than 50 °C. Reduce ambient temperature
High temperature on I/O module	S2	2	The temperature at the I/O module is above 70 °C.	Reduce ambient temperature. The ambient temperature is probably more than 50 °C.
HW module assembly is different	S1	1	The processing station cannot accept ‘sync’ state, because the assembly of the two process station modules is different.	Check the assembly of the two process station modules.
IN-Wire break of channel ’xxxx’	S3	2	The cable connection for the input signal IN is faulty.	Check the relevant process signal cable.
IO Bus: Module not ready	S2	2	The IO module is detected, but it is not ready for cyclic data exchange.	Check the IO module and check the process voltage of the IO modules.
IO Bus: Parameterization failed	S2	2	The module could not parametrized.	Check the IO module or con-tact your authorized service engineer.
IO Module: Checksum error (19)	S2	2	The module has announced the diagnostic value 19 - Checksum error.	Check the IO module or con-tact your authorized service engineer.
IO Module: Diagnosis overflow (9)	S2	2	The module has announced the diagnostic value 9 - Overflow of diagnostic messages.	Check the IO module or con-tact your authorized service engineer.

Message text	Pr	MR	Cause of message	Remedy
IO Module: Different versions (40)	S2	2	The module has announced the diagnostic value 40 - Version error.	Check the IO module or con-tact your authorized service engineer.
IO Module: Internal data exchange (36)	S2	2	The module has announced the diagnostic value 36 - Error during internal data ex-change.	Check the IO module or con-tact your authorized service engineer.
IO Module: Internal error (43)	S2	2	The module has announced the diagnostic value 43 - Internal error.	Check the IO module or con-tact your authorized service engineer.
IO Module: Low process voltage (11)	S2	2	The module has announced the diagnostic value 11 - Low process voltage.	Check the IO module or con-tact your authorized service engineer.
IO Module: No Communication / Slot empty	S2	2	The module has not communicated properly for a specified length of time or the module is not plugged.	Check the IO module or con-tact your authorized service engineer.
IO Module: Parameter error (26)	S2	2	The module has announced the diagnostic value 26 - wrong parameter values found.	Check the IO module or con-tact your authorized service engineer.
IO Module: Process voltage switched off (45)	S2	2	The module has announced the diagnostic value 45 - No process voltage.	Check the IO module or con-tact your authorized service engineer.
IO Module: Timeout (3)	S2	2	The module has announced the diagnostic value 3 - Time out.	Check the IO module or con-tact your authorized service engineer.
IO Module: Type not supported	S2	2	The detected module type is not supported with this soft-ware version.	Check the plugged modules, perhaps an AC 500 module has been used.
IO Module: Wrong type plugged	S2	2	The configured and plugged module types are different.	Correct the configuration.

Message text	Pr	MR	Cause of message	Remedy
Lateral communic: Timeout from station ’xxxx’	S2	2	Station ’xxxx’ has sent its data in the specified interval time.	Check that the process station is functioning correctly, and check its system bus connection. Check the possibility that the station may be operating under overload and therefore be unable to send its lateral data in the specified interval.
Maximum of redundancy data exceeded	S1	2	The volume of redundancy data per task is limited to 64 KB. One of the tasks has exceeded this limit, and the redundancy has been completely deactivated.	Dividing the program of the task in question into several parts reduces the volume of redundancy data.
Message connection to station ’xxxx’ lost	S1	2	The message connection for logs and messages to station ’xxxx’ is lost.	Check that station ’xxxx’ is functioning correctly and check its system bus connection.
Messages lost: Overflow for prio ’xxxx’	S1	1	The internal buffers for messages of priority ‘xxxx’ are no longer large enough.	An operator station is unable to process the messages as quickly as they are generated. Reduce the number of messages.
Miss. extern. power supply output channel ’xxxx’	S3	2	No auxiliary power, or disrupted auxiliary power to the frequency input module in question.	Check the connection for this auxiliary power supply, or the supply itself.
Miss. external power supply channel group 1 [to 4]	S3	2	No auxiliary power, or disrupted auxiliary power to the channel group in question.	Check the connection for this auxiliary power supply, or the supply itself-
Missing transmitter supply channel group 1 [to 2]	S3	2	No external power supply for the channel in question.	Check the connection for this power supply, or the voltage itself.

Message text	Pr	MR	Cause of message	Remedy
Module fault	S2	2	Errors occurred in the module.	Change the module or contact your authorized service engineer.
Module fault: Boot test error	S2	2	Errors occurred in the module's boot test.	Change the module or contact your authorized service engineer.
Module fault: Communication error	S2	2	The module has not communicated properly for a specified length of time.	Change the module or contact your authorized service engineer.
Module fault: Configuration error	S2	2	The configuration of the module is not valid.	Change the configuration or contact your authorized service engineer.
Module fault: Identification error	S2	2	The module cannot be identified.	Change the module or contact your service technician.
Module fault: No master	S2	2	No master can be found on the PROFIBUS.	Check the PROFIBUS connection.
Module fault: Parameter error on master	S2	2	The configuration of PROFIBUS master do not match to the connected device.	Change the configuration or contact your authorized service engineer.
Module fault: Parameter error on slave	S2	2	The configuration of PROFIBUS slave do not match to the connected device.	Change the configuration or contact your authorized service engineer.
Module fault: Self test error	S2	2	In its cyclical self-test the module has detected a fatal error.	Contact your authorized service engineer.
Module fault: Slave not existent	S2	2	The configured PROFIBUS slave cannot be connected.	Change the configuration or contact your authorized service engineer.
Module fault: Slave not ready	S2	2	The configured PROFIBUS slave does not communicate correctly.	Change the configuration or contact your authorized service engineer.

Message text	Pr	MR	Cause of message	Remedy
Module fault: Slot empty	S2	2	A specific module type has been configured for a slot, but no module has been plugged in.	Alter the configuration to match the hardware installed.
Module fault: Wrong firmware version	S2	2	The module has an incorrect firmware version.	Load a new firmware version in the module, change the module, or contact your service technician.
Module [IP1/2]: Number self test errors “nnn"	S1	2	The number of self test errors detected in the module of process station is dumped. If the number of self test errors reaches 5, then the module will no longer boot up.	Contact your authorized service engineer.
Module self test error IP1/2	S2	2	In its cyclical self-test the module has detected a failure that cannot be accepted (error number ‘xxxx’).	Contact your authorized service engineer.
Network [1 to 4] connection failure [IP1/2]	S1	1	The system bus network connection of the is incorrect.	Check the system net connection of the process station.
Network buffer error, warmstart recommended	S1	1	There are no more net work buffers available, communication with the process stations does not work correctly.	A warm start will solve the current problem. Check and clean your net work to avoid this problem for the future.
Network buffer low, warmstart recommended			Only few net work buffers are available, communication with the process stations is endangered.	A warm start will solve the current problem. Check and clean your net work to avoid this problem for the future.
No prim. and second. master connection	S2	2	The redundant PROFIBUS connection is disturbed.	Check the connection and configuration of the redundant PROFIBUS line.

Message text	Pr	MR	Cause of message	Remedy
No RED Link connection	S1	2	Communication disruption on the RED link, possibly a faulty cable, or cable not plugged in.	Check cable connection.
No valid daylight saving time defined	S2	2	Time cannot be converted to summer time. The summer-time table's period of validity has expired, or the table does not contain any times in standard chronological order. Another possible cause is that the time to be converted lies precisely within the one undefined hour when summer time starts.	Correct or extend your summer-time table, or check the time to be converted against the last error source declared.
Object error, first adjustment not possible	S1	2	The user program could not be loaded fully.	Update the project data with "Load whole station" or "Load changed objects".
Output 10V faulty at I: 'xxxx'	S3	2	Output voltage 10 V faulty.	Check output load.
Over / under ranging / wire break channel 0 [to 7]	S3	2	The measuring range has been over/under-ranged for the channel in question, or the cable is damaged there.	Ensure that values remain within the permissible measuring range and/or check the process signal cable.
Overflow/broken wire at C: 'xxxx'	S3	2	The measuring range has been over ranged for the listed channel or the cable is damaged there.	Ensure that values remain within the permissible measuring range and/or check the process signal cable.
Overflow/broken wire at I: 'xxxx'	S3	2	The measuring range has been over ranged for the listed input channel or the cable is damaged there.	Ensure that values remain within the permissible measuring range and/or check the process signal cable.
Overflow/broken wire at O: 'xxxx'	S3	2	The measuring range has been over ranged for the listed output channel or the cable is damaged there.	Ensure that values remain within the permissible measuring range and/or check the process signal cable.

Message text	Pr	MR	Cause of message	Remedy
Overload, first adjustment not possible	S1	2	The CPU module is operating with an overload, with the result that redundant operation cannot be initiated. The redundancy has been completely deactivated.	Rectify the overload by altering the task interval times or splitting up the program.
Power fail of power supply 1 [to 2]	S1	2	The power supply in question has failed.	Check cable connection and the supply itself.
Prim/Sec toggle by fieldbus communication	S1	1	Failure of the fieldbus communication has triggered a redundancy toggle.	Check the connection and configuration of the fieldbus devices.
Prim/Sec toggle by function	S1	1	Redundancy toggle caused by user function call PRIM/SEC in the normal program.	Normal operation.
Prim/Sec toggle by HW module failure	S1	1	Redundancy toggle through failure of a module of the process station.	Replace the defective module or check for other causes.
Prim/Sec toggle by network error	S1	1	Redundancy toggle through detection of an error in the system bus network connection.	Check the system bus connection of the CPU modules, in particular that of the current secondary.
Prim/Sec toggle by primary failure	S1	1	Redundancy toggle caused by failure of the primary CPU module.	Replace faulty CPU, or investigate other causes.
Prim/Sec toggle by toggle button	S1	1	Redundancy toggle by operating the toggle button in Freelance Engineering.	Normal operation.
Prim/Sec toggle by toggle switch	S1	1	Redundancy toggle by operating the toggle button on the primary CPU-module.	Normal operation.
Process bus 1 [to 3] data lost: Reception overload	S2	1	The CPU-module was unable to process the received data quickly enough.	Check the terminating resistors and the cable lengths or contact your authorized service engineer.

Message text	Pr	MR	Cause of message	Remedy
Process bus 1 [to 3] data lost: Sending overload	S2	1	The CPU-module was unable to send any data for a certain length of time.	Contact your authorized service engineer.
Process bus data lost while receiving	S2	1	The module was unable to process the received data quickly enough.	Check the terminating resistors and the cable lengths or contact your authorized service engineer.
Process bus data lost while sending	S2	1	For a certain period of time the module was unable to send any data.	Check the terminating resistors and the cable lengths or contact your authorized service engineer.
Process bus data lost / sending (Distance)	S2	1	For a certain period of time the module was unable to send any data. It is likely that the connection distance to the supplementary racks is too great.	Reduce the distance or contact your authorized service engineer.
PROFIBUS: Diagnosis buffer overflow	S2	2	The PROFIBUS slaves have given an overflow of diagnosis data. Probably 'flutter' signals.	Check the PROFIBUS configu-ration and, if possible, in-crease the number of the PROFIBUS diagnosis buffers of the master. Check the Profi-bus slaves or contact the slave manufacturer's service technician.
PROFIBUS: DP Master Bus Cycle Timeout	S2	2	The maximum bus cycle time of the PROFIBUS communication was exceeded. The PROFIBUS master was reset.	Check the configuration and connections of all PROFIBUS devices or contact your service technician.
PROFIBUS: DP Master in state CLEAR	S2	2	PROFIBUS DP Master is in the CLEAR state.	Check the configuration, connection and status of all PROFIBUS devices.
PROFIBUS DP Master in state ERROR [IP1/2]	S2	2	PROFIBUS DP Master is disturbed, all devices fail.	Check the configuration, connection and status of all PROFIBUS devices.

Message text	Pr	MR	Cause of message	Remedy
PROFIBUS: DP Master in state OFFLINE	S2	2	The PROFIBUS DP Master is in the OFFLINE state.	Check the configuration and connection to the PROFIBUS master and the status of the PROFIBUS module.
PROFIBUS: DP Master in state STOP	S2	2	PROFIBUS DP Master is in the STOP state.	Check the configuration, connection and status of all PROFIBUS devices.
PROFIBUS: Duplicate master addr detected	S2	2	The PROFIBUS master software has detected that another PROFIBUS master with an identical address is active on the same PROFIBUS line.	Check the configuration of the PROFIBUS line.
PROFIBUS: Firmware API malfunction	S2	2	The PROFIBUS master software has detected an error in the protocol stack. Depending on the nature of the error, a reset of the protocol stack is initiated and PROFIBUS communication is resumed.	
PROFIBUS: Firmware API timeout	S2	2	The PROFIBUS master software has detected a timeout in the protocol stack. Depending on the nature of the error, a reset of the protocol stack is initiated and PROFIBUS communication is resumed.	
PROFIBUS: Firmware error (xxx)	S2	2	An internal error has occurred in the communications software of the PROFIBUS Master.	Contact your service technician.
PROFIBUS: Firmware version error [IP1/2]	S2	2	The Firmware version of the PROFIBUS module is outdated.	Load a new Firmware version into the module, change the module or contact your service technician.

Message text	Pr	MR	Cause of message	Remedy
PROFIBUS: Line A/B no data reception [IP1/2]	S2	2	The redundancy of the PROFIBUS line is disturbed.	Check the configuration and the devices of the PROFIBUS.
PROFIBUS: Master bus cycle timeout	S2	2	The redundancy of the Profi-bus line is disturbed.	Check the configuration and the devices of the PROFIBUS.
PROFIBUS: No connection to any slave [(IP1/2)]	S2	2	The PROFIBUS master software has detected a physical error on the line, and it is no longer possible to perform any-communication.	Check the configuration and the devices of the PROFIBUS.
PROFIBUS: Physical layer malfunction	S2	2	The PROFIBUS master software has detected a physical error on the line, and it is no longer possible to perform error-free communication.	Check the configuration of the PROFIBUS line.
Readback fault in output channel	S2	2	An error has occurred during signal readback on an output channel.	Contact your authorized service engineer.
Red startup without current process values	S1	1	After redundancy toggle, the new primary CPU is not receiving any current input data within the configured waiting time.	Check the configuration of the processing station, in particular, the parameter “Toggle timeout for field bus inputs”.
Redundant fieldbus comm. not available	S1	1	The redundant communication of a fieldbus has failed.	Check the connections and configuration of the connected field busses.
RS-Wire break of channel ’xxxx’	S3	2	The cable connection for the input signal RS (Run/Stop) is faulty.	Check the relevant process signal cable.

Message text	Pr	MR	Cause of message	Remedy
RUN / STOP mismatch P-RUN <-> S-STOP	S1	2	Mismatched positions of RUN/STOP switches on CPU-modules, with current positions displayed.	Adjust switch positions.
RUN / STOP mismatch P-STOP <-> S-RUN	S1	2	Mismatched positions of RUN/STOP switches on CPU-modules, with current positions displayed.	Adjust switch positions.
RUN/STOP mismatch switch <-> state	S1	2	The positions of the RUN / STOP-switches of the Primary and Secondary are different. After redundancy toggling, the switch position and state (RUN / STOP) on the new primary do not correspond.	Match the switch positions.
SD card removed [(IP1/2)]	S2	2	SD card could not be detected in the controller.	Insert a valid SD card in the controller.
Sec boot cause: CPU fault detected	S1	1	A fatal software error has occurred on the module.	Contact your authorized service engineer.
Sec boot cause: Critical error	S1	1	A critical software error has triggered a new synchronization by the primary CPU module.	Contact your authorized service engineer.
Sec boot cause: Power fail	S1	1	A new synchronization has been triggered on the CPU-module by the reset button. Reset button pressed for longer than 5 seconds.	Normal operation, check your power supply if necessary.
Sec boot cause: Reset switch	S1	1	A new synchronization has been triggered on the CPU-module by the reset button. Reset button pressed for longer than 5 seconds.	Normal operation.

Message text	Pr	MR	Cause of message	Remedy
Sec boot cause: Software error	S1	1	A fatal software error has occurred on the module.	Contact your authorized service engineer.
Sec boot cause: Watchdog	S1	1	The CPU-module watchdog has triggered a new synchronization by the primary CPU-module.	Contact your authorized service engineer.
Secondary failure	S1	2	In the course of synchronous operation, the secondary has failed.	Check that the CPU module is fully functional. Also occurs when there is a fault in the RED link connection.
Secondary has an invalid operating system	S1	1	The station cannot accept the ‘sync’ state, because the operating system of the secondary does not correspond to that of the primary.	Load the operating system of the secondary using the Freelance Settings tool.
Secondary not found	S1	2	The redundancy partner (secondary) cannot be found, and consequently cannot be loaded.	Check whether there is a secondary CPU, and whether the RED link is connected correctly.
Secondary not reachable after boot	S1	2	After the operating system has been loaded, the secondary cannot be accessed.	Check whether the secondary was unable to terminate the boot-up of its operating system.
Short circuit at C: ’xxxx’	S3	2	A short circuit has been detected for one or more digital channels; all channels are listed if enough space in text.	Check the relevant process signal cable.
Short circuit at I: ’xxxx’	S3	2	A short circuit has been detected for one or more input channels; all channels are listed if enough space in text.	Check the relevant process signal cable.

Message text	Pr	MR	Cause of message	Remedy
Short circuit at O: ’xxxx’	S3	2	A short circuit has been detected for one or more output channels; all channels are listed if enough space in text.	Check the relevant process signal cable.
Shortcut of channel ’xxxx’	S3	2	The cable connection for the analog input has a short circuit.	Check the relevant process signal cable.
Slave deactivated due to protocol error	S2	2	Errors occurred on the communication to the PROFIBUS slave.	Check the related PROFIBUS device or contact slave manufacturer's service technician.
Slave was parameterized by another Master	S2	2	The PROFIBUS device was configured from another master. A different master has parameterized the slave.	Check the configuration of the other PROFIBUS master.
Stat. boot cause: Bootstrap by operator	S1	1	An operator action (in Freelance Engineering) has caused Bootstrap on the CPU-module.	Normal operation.
Stat. boot cause: Cold start by operator	S1	1	An operator action (in Freelance Engineering) has caused a cold start on the process station.	Normal operation.
Stat. boot cause: Cold start by program	S1	1	The application program has caused a cold start on the process station.	Normal operation.
Stat. boot cause: Cold start by reset switch	S1	1	A cold start has been triggered on the CPU-MODULE by the reset button. Reset button pressed for longer than 5 sec's.	Normal operation.
Stat. boot cause: Cold start/Safety state	S1	1	After reaching the safety state a cold start on the process station has caused.	Check reason for safety state using the boot cause information dialog.
Stat. boot cause: CPU fault detected	S1	1	A fatal software error has occurred on the CPU-module.	Contact your authorized service engineer.

Message text	Pr	MR	Cause of message	Remedy
Stat. boot cause: Default configuration only	S1	1	No boot configuration is present in the process station. The process station has been initialized since it does not contain a boot configuration. The standard (default) configuration has been reverted to.	Contact your authorized service engineer.
Stat. boot cause: Enter recover mode	S1	1	A warm start has been activated on a process station.	Contact your authorized service engineer.
Stat.boot cause Initialization / SD card	S1	1	The controller has been initia-lized with the data from the SD card.	Normal operation
Stat. boot cause: Initialization by operator	S1	1	An operator action (in Freelance Engineering) has caused an Initialize on the process station.	Normal operation.
Stat. boot cause: Redundancy boot	S1	1	CPU starts after a redundancy toggle.	
Stat. boot cause: Software error	S1	1	A fatal software error has occurred on the CPU-module.	Contact your authorized service engineer.
Stat. boot cause: Warm start by operator	S1	1	An operator action (in Freelance Engineering) has caused a warm start on the CPU-module.	Normal operation.
Stat. boot cause: Warm start by reset switch	S1	1	A warm start has been triggered on the CPU-module by the reset button. Reset but-ton pressed for less than 5 seconds.	Normal operation.
Stat. boot cause: Warm start / critical error	S1	1	A critical software error has triggered a warm start on the CPU-module.	Contact your authorized service engineer.
Stat. boot cause: Watchdog	S1	1	The CPU-module watchdog has triggered a boot.	Contact your authorized service engineer.

Message text	Pr	MR	Cause of message	Remedy
Station Ethernet overload of interface xxx	S1	1	The communication load on interface xxx is too high. The Ethernet controller was temporarily switched off.	Check the configuration and the associated Ethernet devices.
Station FPGA image error xxx	S1	1	An error has occurred in the FPGA of the controller's CPU.	Contact your authorized service engineer.
Station radio clock failed	S1	1	A radio clock linked to the process station has failed.	Check radio clock and/or connection.
Station software error ’xxxx’	S1	1	A fatal error has occurred on the CPU-module, ’xxxx’ contains its ID.	Contact your authorized service engineer.
Station stopped	S1	1	Station resource has been stopped.	Normal operation, start the resource.
Station system time not set	S1	2	The system time of the re-source has never been set. The message is displayed with a time stamp that is calculated with a system time DT#2099-12-31 00:00:000 and the actual time zone. The controller performs a cold start on power fail.	Set the system time in your plant (Freelance Engineering, Commissioning, Options, Set system time).
Station task break point activated	S2	1	A user task in the station has reached a break point.	In commissioning mode, debugger: deactivate or remove the break point. Allow task to continue running.
Station task ready	S2	1	The user task in question is ready to be started.	Normal operation.
Station task stopped	S2	1	The user task in question has been stopped.	Normal operation.

Message text	Pr	MR	Cause of message	Remedy
Station warm start down time > 24 days	S1	1	The CPU-module has been restarted, and has performed a warm start following a power failure. Power down time was higher than 24 days.	Normal operation, check power supply if necessary.
Sys. time set at ’xxxx’	S1	1	The system time in the process or Gateway station has been changed by an operator action (in Freelance Engineering), ’xxxx’ contains the original time (local).	Normal operation.
System network connection failure [IP1/2]	S1	1	The system network connection of the is incorrect.	Check the system net connection of the process station.
Task not executable: DT overflow	S1	1	The program of the user task in question has caused a DT arithmetic error.	Correct problem in user program.
Task not executable: DT underflow	S1	1	The program of the user task in question has caused a DT arithmetic error.	Correct problem in user program.
Task not executable: Illegal array index	S1	1	An illegal array index has been calculated in the user program.	Check and correct the user program.
Task not executable: INT div. by 0	S1	1	The program of the user task in question has caused an integer arithmetic error.	Correct problem in user program.
Task not executable: INT FB error	S1	1	The program of the user task in question has caused an arithmetic error within an H&B function block.	Contact your authorized service engineer.
Task not executable: INT overflow	S1	1	The program of the user task in question has caused an integer arithmetic error.	Correct problem in user program.

Message text	Pr	MR	Cause of message	Remedy
Task not executable: INT overflow (store)	S1	1	The program of the user task in question has caused an integer overflow on saving.	Correct problem in user program.
Task not executable: INT underflow	S1	1	The program of the user task in question has caused an integer arithmetic error.	Correct problem in user program.
Task not executable: Invalid debug command	S1	1	If the resource is halted while a task is at a break point, the task switches to state 'not executable'. The error task is not initiated.	The resource must not be halted whilst one of the tasks is at a break point.
Task not executable: Process image read	S1	1	The program of the user task in question has caused an error on reading in the process values.	The configuration has not been loaded correctly, presumably because of an operator error. Update the project data with “Load whole station” or “Load changed objects”.
Task not executable: Process image write	S1	1	The program of the user task in question has caused an error on outputting the process values.	The configuration has not been loaded correctly, presumably because of an operator error. Update the project data with “Load whole station” or “Load changed objects”.
Task not executable: Program execution abort	S1	1	The program in the user task concerned has been interrupted. A task is running in an endless loop.	Correct problem in user program.
Task not executable: Program execution error	S1	1	The program of the user task in question has caused a program execution error.	Correct problem in user program.

Message text	Pr	MR	Cause of message	Remedy
Task not executable: REAL div. by 0	S1	1	The program of the user task in question has caused a real arithmetic error.	Correct problem in user program.
Task not executable: REAL FB error	S1	1	The program of the user task in question has caused an arithmetic error within an H&B function block.	Contact your authorized service engineer.
Task not executable: REAL no valid float	S1	1	The program of the user task in question is attempting to process an illegal real value.	Correct problem in user program.
Task not executable: REAL overflow	S1	1	The program of the user task in question has caused a real arithmetic error.	Correct problem in user program.
Task not executable: REAL underflow	S1	1	The program of the user task in question has caused a real arithmetic error.	Correct problem in user program.
Task not executable: UINT div. by 0	S1	1	The program of the user task in question has caused an integer arithmetic error.	Correct problem in user program.
Temp. comp. chan. error at I: 'xxxx'	S3	2	Possibly wrong measured value caused by inadmissible temperature of the compensation channel.	Check the temperature compensation channel.
Time stamp is off in module DDI01	S2	2	The “Time-stamp” mode is switched off in DDI01.	To record the sequence of events messages, the "Time stamp" function in DDO 01 must be activated.
Underflow C: 'xxxx'	S3	2	The measuring range has been underrun for the listed channels.	Ensure that values remain within the permissible measuring range.
Underflow I: 'xxxx'	S3	2	The measuring range has been underrun for the listed input channels.	Ensure that values remain within the permissible measuring range.

Message text	Pr	MR	Cause of message	Remedy
Underflow O: ’xxxx’	S3	2	The measuring range has been underrun for the listed output channels.	Ensure that values remain within the permissible measuring range.
Voltage diff. high at I: ’xxxx’	S3	2	Invalid measured value of the channel caused by too high voltage difference.	Check voltage difference; install equalizing conductors if necessary
Wire break of analog input	S3	2	The process signal cable at the analog input is faulty.	Check the process signal cable.
Wire break of channel ’xxxx’	S3	2	The cable connection for the analog input is faulty.	Check the relevant process signal cable.
Wrong module type plugged [IPx]	S2	2	The block type configured is incompatible with the module plugged in. (IPx = IP address of the redundant process station).	Alter the configuration to match the hardware installed.

Index

—

www.abb.com/freelance
www.abb.com/controlsystems

—

We reserve the right to make technical changes to the products or modify the contents of this document without prior notice. With regard to purchase orders, the agreed particulars shall prevail. ABB does not assume any responsibility for any errors or incomplete information in this document.