

# **ABBCS31 Communication Server**

for Microsoft Windows  
and InTouch Applications

**User Manual  
Ver 1.x Rev 1.9  
DR 010 06  
DR 010 07**

KLINKMANN AUTOMATION  
P.O. Box 38  
FIN-00371 Helsinki Finland  
tel. int. + 358 9 5404940  
fax int. + 358 9 5413541  
[www.klinkmann.com](http://www.klinkmann.com)

# Table Of Contents

Overview.....	1
Communication Protocols.....	1
Accessing a Remote Item from ABBCS31 .....	3
The Connection Cable.....	3
Installing the ABBCS31 Server.....	4
Installing the Server .....	4
Installing the I/O Server Infrastructure.....	5
Configuring the ABBCS31 Server .....	7
Configure Server Settings Command.....	7
Configure Communication Port Command .....	9
Saving ABBCS31 Configuration File .....	10
Configuration File Location .....	10
Topic Definition Command.....	11
Item (Point) Naming.....	14
ABB CS31 07 KT93 Controllers .....	14
ABB CS31 07 KT94 Controllers .....	15
Common features of all Controller's Item (Point) Naming .....	16
Monitoring and Controlling Communication with a PLC .....	17
Using the ABBCS31 Server with OPC Clients.....	19
Configuring DCOM.....	19
Running ABBCS31 "OPC & DDE" version as Windows NT Service.....	21
Using ABBCS31 with OPCLink Server.....	22
OPCLink Topic Definition .....	22
Accessing ABBCS31 Items via the OPCLink Server.....	23
Using the ABBCS31 Server with Suite Link and DDE Clients .....	24
Using the ABBCS31 Server with InTouch .....	24
Defining the Access names.....	24
Defining the Tag names .....	26
Monitoring the Status of Communication with InTouch .....	28
Notes on Using Microsoft Excel.....	29
Reading Values into Excel Spreadsheets .....	29
Writing Values to ABBCS31 Points.....	29
Troubleshooting.....	30
WIN.INI entries.....	30
Troubleshooting menu .....	32

# ABBCS31 Communication Server

## Overview

The **ABBCS31 Communication Server** (hereafter referred to as the "ABBCS31 Server" or "ABBCS31" or "Server") is a Microsoft Windows NT 32-bit application program that acts as a communication protocol Server and allows other Windows application programs to access to data from the ABB Procontic CS31 automation system via serial port. The Server has been tested with ABB CS31 07 KR91, 07 KT93 and 07 KT94 Controllers.

The *Server* may be used by any Microsoft Windows program that is capable of acting as a **DDE, FastDDE, SuiteLink** or **OPC Client**.

There are two different ABBCS31 Server versions described in this manual:

- Server version (ordering number DR 010 06), supporting SuiteLink, FastDDE and DDE protocols; this version hereafter is referred to as the "**Suite Link & DDE**" version.
- Server version (ordering number DR 010 07), supporting OPC and DDE protocols; this version hereafter is referred to as the "**OPC & DDE**" version;

The separate installation package is supplied for each version of the Server. In all cases the name of Server executable file is **ABBCS31.EXE**. All further information in this manual is same for all versions of the Server, with the exception of few points where communication protocol specific features are explained.

## Communication Protocols

**Dynamic Data Exchange (DDE)** is a communication protocol developed by Microsoft to allow applications in the Windows environment to send/receive data and instructions to/from each other. It implements a client-server relationship between two concurrently running applications. The server application provides the data and accepts requests from any other application interested in its data. Requesting applications are called clients. Some applications such as Wonderware InTouch and Microsoft Excel can simultaneously be both a client and a server.

**FastDDE** provides a means of packing many proprietary Wonderware DDE messages into a single Microsoft DDE message. This packing improves efficiency and performance by reducing the total number of DDE transactions required between a client and a server. Although Wonderware's FastDDE has extended the usefulness of DDE, this extension is being pushed to its performance constraints in distributed environments.

The ABBCS31 Server supports the FastDDE Version 3 - an extension to Wonderware's proprietary FastDDE Version 2. This extension supports the transfer of Value Time Quality (VTQ) information. The original DDE and FastDDE Version 2 formats are still supported, providing full backward compatibility with older DDE clients. FastDDE Version 3 works on Windows 9x systems as well as Windows NT systems.

**NetDDE** extends the standard Windows DDE functionality to include communication over local area networks and through serial ports. Network extensions are available to allow DDE links between applications running on different computers connected via networks or modems. For example, NetDDE supports DDE between applications running on IBM compatible computers connected via LAN or modem and DDE-aware applications running on non-PC based platforms under operating environments such as VMS and UNIX.

**SuiteLink** uses a TCP/IP based protocol and is designed by Wonderware specifically to meet industrial needs such as data integrity, high-throughput, and easier diagnostics. This protocol standard is only supported on Microsoft Windows NT 4.0 or higher. SuiteLink is not a replacement for DDE, FastDDE, or NetDDE. The protocol used between a client and a server depends on your network connections and configurations. SuiteLink was designed to be the industrial data network distribution standard and provides the following features:

- Value Time Quality (VTQ) places a time stamp and quality indicator on all data values delivered to VTQ-aware clients.
- Extensive diagnostics of the data throughput, server loading, computer resource consumption, and network transport are made accessible through the Microsoft Windows NT operating system Performance Monitor. This feature is critical for the scheme and maintenance of distributed industrial networks.
- Consistent high data volumes can be maintained between applications regardless if the applications are on a single node or distributed over a large node count.
- The network transport protocol is TCP/IP using Microsoft's standard WinSock interface.

**OPC** (OLE for Process Control) is an open interface standard to provide data from a data source and communicate the data to any client application in a common standard way. The OPC is based on Microsoft OLE, COM and DCOM technologies and enables simple and standardized data interchange between the industrial or office sector and the production sector. From general point of view many aspects of OPC are similar to DDE, but main difference is in the implementation by using Microsoft's COM (Component Object Model) technology. It enables fast exchange with process automation data and OPC open interface allows access to data from OPC Server in same standard way from OPC client applications supplied by different developers.

For more information on the basics of OPC, please refer to the **OPC Specification**. The OPC Data Access Custom Interface Specification is maintained by **OPC Foundation**, the current specification is 2.04 dated September 2000.

The OPC support for ABBCS31 Server "OPC & DDE" version is implemented based on **FactorySoft OPC Server Development Toolkit** and it conforms to OPC Data Access Custom Interface Specification 2.04. The ABBCS31 Server "OPC & DDE" version is tested for compliance and is compatible with OPC Foundation OPC Data Access Compliance Test Tool.

The Suite Link, FastDDE (Version 3) and DDE support for ABBCS31 Server "Suite Link & DDE" version is implemented by **Wonderware I/O Server Toolkit** ver. 7.0 (060).

The FastDDE (Version 2) and DDE support for ABBCS31 Server "OPC & DDE" version is implemented by *Wonderware I/O Server Toolkit* ver. 5.0 (008).

## Accessing a Remote Item from ABBCS31

The communication protocol addresses an element of data in a conversation that uses a three-part naming convention that includes the *application name*, *topic name* and *item name*. The following briefly describes each portion of this naming convention:

### *application name*

The name of the Windows program (server) that will be accessing the data element. In the case of data coming from or going to ABB Procontic CS31 hardware via this Server, the application portion of the address is **ABBCS31**.

### *topic name*

Meaningful names are configured in the Server to identify specific devices (PLCs). These names are then used as the topic name in all conversations to that PLC. The **ABBCS31** Server considers each ABB Procontic PLC to be a separate topic. For example, **PLC1**.

**Note!** You can define multiple topic names for the same PLC to poll different points at different rates and/or through different FA networks.

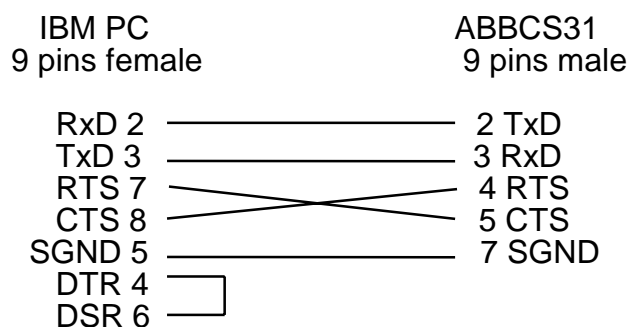
**Note:** In some cases, the term "node" is used interchangeably with the term "topic".

### *item name*

A specific data element within the specified topic. When using the ABBCS31 Server, an item is an operand (variable): BINARY input, BINARY output, BINARY flag, BINARY step, BINARY constant, WORD input, WORD output, etc. The term "point" is used interchangeably with the term "item" in this User Manual. For more information on item/point names, see the *Item (Point) Naming* section later in this manual.

## The Connection Cable

For connecting the ABB Procontic CS31 automation system to the IBM PC or compatible personal computer the following cable diagram must be used:



# Installing the ABBCS31 Server

## Installing the Server

The ABBCS31 Server installation package can be supplied:

1. As a self-extracting archive (01006xxx.EXE for "Suite Link & DDE" version or 01007xxx.EXE for "OPC & DDE" version) if downloaded from Klinkmann's web site (the xxx is the current (latest) version of the Server).
2. From installation on CD.
3. On two or three distribution disks (floppies).

To **install** the ABBCS31 Server from the self-extracting archive, run the 01006xxx.EXE or 01007xxx.EXE and proceed as directed by the ABBCS31 Server Setup program.

To **install** the ABBCS31 Server from CD or distribution disks (floppies), on MS Windows (NT, 2000, XP or 95 (98)):

1. Insert the CD with Klinkmann Software into CD drive or insert ABBCS31 Disk1 into a floppy drive A: or B:.
2. Select the **Run** command under the **Start** menu.
3. Run STARTUP.EXE if installing from CD or SETUP.EXE if installing from distribution disks (floppies).
4. If installing from CD: select "Protocol Servers (DDE, SuiteLink, OPC)", find "ABBCS31 SL and DDE Server" or "OPC and DDE Server" and click on "Setup...".
5. Proceed as directed by the ABBCS31 Server Setup program.

### Notes:

1. *If installing the ABBCS31 Server "OPC & DDE" version on Windows 95 then DCOM (Distributed COM) must be installed on Windows 95 before installing the ABBCS31 Server. The DCOM update can be obtained, for example, from the Microsoft's web site <http://www.microsoft.com/com/dcom95/> or from Microsoft Visual C++ 5.0 and 6.0 CDs. Before starting the ABBCS31 Server the DCOM must be configured (both on NT and Win9x) for right "Access Permissions" and "Location" (on NT) or "Properties" (on Win 9x). See **Configuring DCOM** section in this manual for DCOM configuration details.*
2. *All MS Windows (both NT and 9x) applications using Microsoft's shared DLLs (e.g. MFC42.DLL and MSVCRT.DLL) must be closed before installing the ABBCS31 Server "OPC & DDE" version. Otherwise there can be problems with ABBCS31 Server registration as OPC server. If during the ABBCS31 Server "OPC & DDE" version installation some warning messages about shared DLLs are displayed, then it is quite possible the ABBCS31 Server registration as OPC server failed. In this case after system reboot the ABBCS31 Server registration as OPC server can be done by starting the ABBCS31 Server manually with special command line parameter added: "ABBCS31.EXE /Regserver".*

When installation is finished, the subdirectory specified as a folder where to install the ABBCS31 Server files will contain the following files:

**ABBCS31.EXE**            The ABBCS31 Server Program. This is a Microsoft Windows

	32-bit application program.
<b>ABBCS31.HLP</b>	The ABBCS31 Server Help file.
<b>ABBCS31.CFG</b>	An example configuration file.
<b>LICENSE.TXT</b>	Klinkmann Automation software license file.
<b>KLSERVER.DLL</b>	Dynamic Link Library necessary for “OPC & DDE”version of the Server.
<b>WWDLG32.DLL</b>	Dynamic Link Library necessary only for “OPC & DDE”version of the Server.

To **uninstall** the ABBCS31 Server, start Control Panel, select “Add/Remove Programs” and select the “ABBCS31 SL and DDE Server” or “ABBCS31 OPC and DDE Server” from the list of available software products. Click on “Add/Remove...” and proceed as directed by the UnInstallShield program.

**Notes:**

1. The ABBCS31 Server “Suite Link & DDE” version is developed with Wonderware I/O Server Toolkit (ver 7.0) and needs the **Wonderware FS2000 Common Components** to be installed on computer where the ABBCS31 Server is running. The Wonderware FS2000 Common Components are installed automatically when any of Wonderware FS2000 Components (e.g. InTouch or some Wonderware I/O server) is installed.
2. If ABBCS31 Server “Suite Link & DDE” version will run on PC where Wonderware FS2000 Common Components are not installed then a special **I/O Server Infrastructure installation package** can be obtained from Klinkmann Automation (see **Installing the I/O Server Infrastructure** section below). This I/O Server Infrastructure installation package contains the minimum set of software needed to run the ABBCS31 Server “Suite Link & DDE” version and these infrastructure files must be install prior to executing the ABBCS31 Server.
3. The HASP key is needed for full time running of ABBCS31 Server. The HASP Driver setup is performed during the Server setup. Without HASP Driver installed the ABBCS31 Server will run only 1 hour (with all features enabled).

## Installing the I/O Server Infrastructure

The I/O Server Infrastructure installation package can be supplied:

1. As a self-extracting archive (IOServerInfrastructure.exe) if downloaded from Klinkmann’s web site.
2. On one distribution disk (floppy).

To **install** the I/O Server Infrastructure from the self-extracting archive, run the IOServerInfrastructure.exe and proceed as directed by the I/O Server Infrastructure Setup program.

To **install** the I/O Server Infrastructure from the distribution disk, on MS Windows NT:

1. Insert the I/O Server Infrastructure disk into a floppy drive A: or B:.
2. Select the **Run** command under the **Start** menu.

3. Type "A:SETUP" or "B:SETUP".
4. Click on **OK**.
5. Proceed as directed by the I/O Server Infrastructure Setup program.

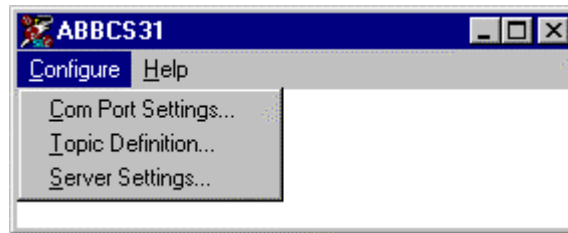
To **uninstall** the I/O Server Infrastructure, start Control Panel, select "Add/Remove Programs" and select the "IO Server Infrastructure" from the list of available software products. Click on "Add/Remove..." and proceed as directed by the UnInstallShield program.

**Note:** *The I/O Server Infrastructure installation will be rejected if Wonderware FS2000 Common Components are already installed on same computer.*

## Configuring the ABBCS31 Server

After the ABBCS31 Server is initially installed, a small amount of configuration is required. Configuring the Server automatically creates an **ABBCS31.CFG** file which holds all of the topic (PLC device) definitions entered, as well as the communication port configurations. This file will automatically be placed in the same directory in which **ABBCS31** is located unless the path where the configuration file will be placed is specified via the `/Configure/DDE Server Settings...` command.

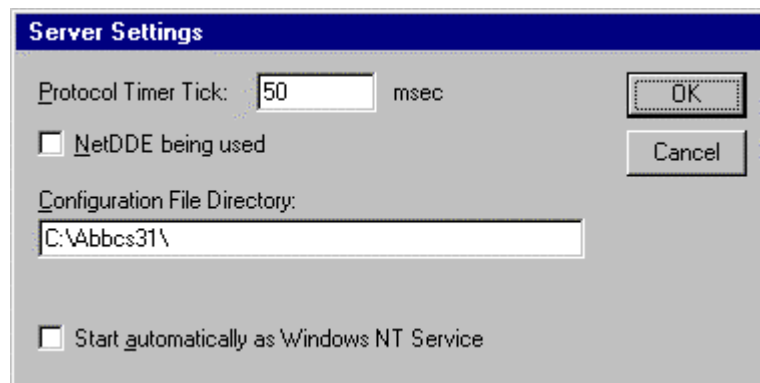
To perform the required configurations, start up the ABBCS31 program. If the Server starts up as an icon, double-click on the icon to open the server's window. To access the commands used for the various configurations, open the `/Configure` menu:



## Configure Server Settings Command

A number of parameters that control the internal operation of the Server can be set. In most cases, the default settings for these parameters provide good performance and do not require changing. However, they can be changed to fine-tune the Server for a specific environment.

To change the Server's internal parameters, invoke the **DDE Server Settings...** command. The "DDE Server Settings" dialog box will appear:



The following describes each field in this dialog box:

### Protocol Timer Tick

This field is used to change the frequency at which the Server checks for work to do (at this frequency the Server tries to send one data request to PLC and receive one reply from PLC. If the send/response cycle is too long then more than one activation of Server is necessary to process it. If computer is very busy or some other MS Windows application is taking over the computer then the Server is activated rarely than setting in the **Protocol Timer Tick**.

**Note:** The default value is 50 milliseconds. The minimum value is 10 milliseconds.

**NetDDE being used**

Select this option if you are in network using NetDDE.

**Configuration File Directory**

The first field is used to specify the path (disk drive and directory) in which ABBCS31 will save its current configuration file. ABBCS31 will use this path to load the configuration file the next time it is started.

**Note:** Only the "path" may be modified with this field. The configuration file is always named **ABBCS31.CFG**.

**Note:** There is no limit to the number of configuration files created, although each must be in a separate directory. When using the ABBCS31 Server with **InTouch**, it is good practice to place the configuration file in the application directory.

**Start automatically as Windows NT Service**

Enabling this option will cause the server to start as a Windows NT service.

Windows NT offers the capability of running applications even when a user is not logged on to the system. This is valuable when systems must operate in an unattended mode. Enabling this option and rebooting the system will cause the Server to run as a Windows NT service. However, to view configuration information or to reconfigure the Server, the user must log on to the system. Any Server related problems that may arise such as missing adapter cards, licensing failures or device drivers not loading will not be visible to the user until a log on is performed. Disabling this option and rebooting the system will cause the Server to run as a Windows NT application program once again.

**Notes:**

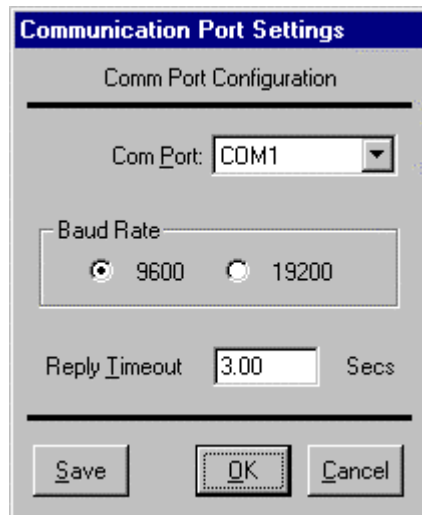
1. The **Start automatically as Windows NT Service** feature can be activated only with ABBCS31 Server "Suite Link & DDE" version. To start the ABBCS31 Server "OPC & DDE" version as Windows NT Service, refer to **Running ABBCS31 "OPC & DDE" version as Windows NT Service** section of this manual.
2. The Service Startup configuration can be changed by MS Windows NT **Control Panel/Services** configuration dialogs. The **Allow Service to Interact with Desktop** checkbox in "Service" dialog box must be checked (the "Service" dialog box can be invoked by pressing the "Startup" button on "Services" dialog box when Service **ABBCS31\_IOServer** is selected). If **Allow Service to Interact with Desktop** is not selected then ABBCS31 Server full functionality is not ensured (e.g. the Server configuration can not be changed, no message boxes will be displayed, etc.).

Once all entries have been made, click on **OK**.

## Configure Communication Port Command

This command is used to configure the communication port that will be used to communicate with the ABB Procontic CS31. Invoke the **Comm Port Configuration...** command.

The "Communication Port Settings" dialog box will appear:



The following describes each field in this dialogue box:

### Com Port

This field is used to select a communication port.

**Note:** The default Communication port is COM1.

### Baud Rate

Select the Baud Rate that matches the setting chosen for the hardware.

**Note:** The default Baud Rate is 9600.

### Reply Timeout

This field is used to enter the amount of time (in seconds) ABB Procontic CS31 Controller using the selected communication port will be given to reply to commands from the Server.

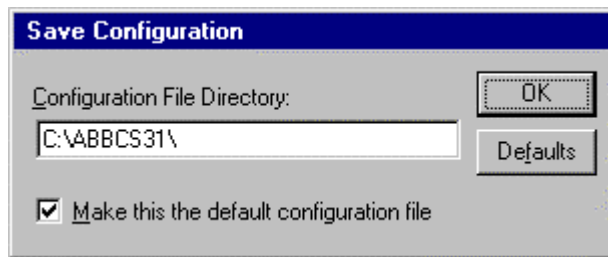
**Note:** The default value of 3 seconds should be sufficient for most configurations.

The ABBCS31 Server supports the following other communication parameters : no parity and one stopbit.

When all entries have been made, click on **OK** to process the configuration for the communication port.

## Saving ABBCS31 Configuration File

If the configuration file does not currently exist, or a new configuration path has been specified, the Server will display the "Save Configuration" dialog box:



This dialog box displays the path where the Server is going to save the current configuration file. The path may be changed if necessary. Also, the path can optionally be recorded in the **WIN.INI** file by selecting the "**Make this the default configuration**" option. Doing so will allow the ABBCS31 Server to find the configuration file automatically each time it is started.

## Configuration File Location

When the ABBCS31 Server starts up, it first attempts to locate its configuration file by, first checking the **WIN.INI** file for a path that was previously specified. If the path is not present in the **WIN.INI** file, the Server will assume that the current working directory is to be used.

To start the Server from an application directory configuration file other than the default configuration file a special switch (**/d:**) is used. For example, invoke the **File/Run** command from the **File Manager** or **Program Manager** and enter the following:

**ABBCS31 /d:c:\directoryname**

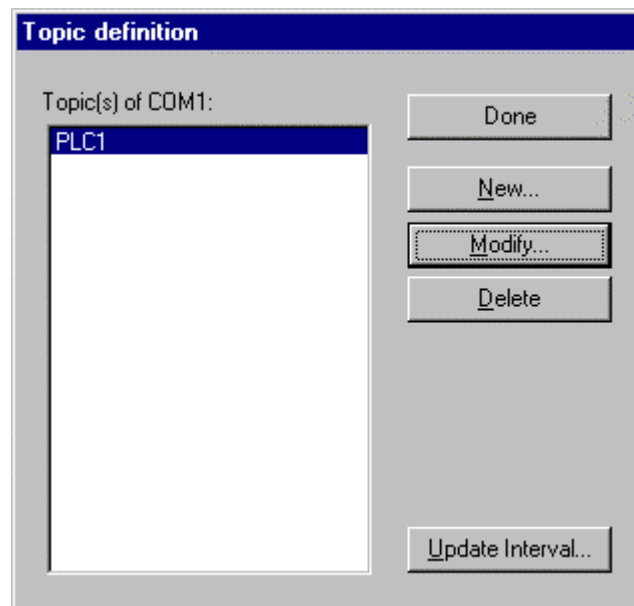
**Note:** *There is no limit to the number of configuration files that may be created, although each must be in a separate directory.*

## Topic Definition Command

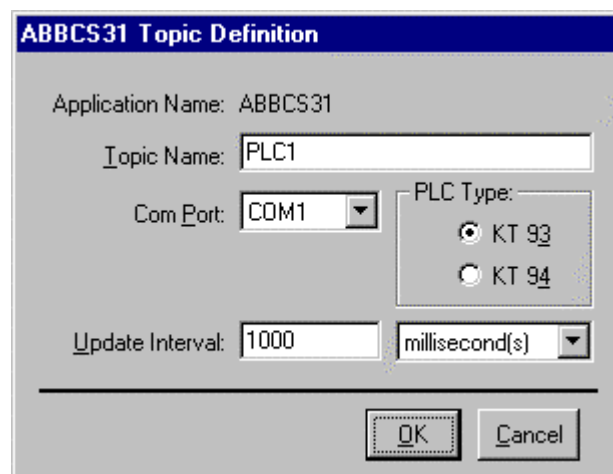
The user provides the PLC with an arbitrary name which is used as the topic for all references to the PLC.

The following steps are taken to define the Topics (PLCs) attached to the ABB Procontic CS31 Control System:

1. Invoke the **Topic Definition...** command. The "Topic definition" dialog box will appear:



2. To modify an existing topic, select the topic name and click on **Modify**. To define a new topic, click on **New**. The "ABBCS31 Topic Definition" dialog box will appear:

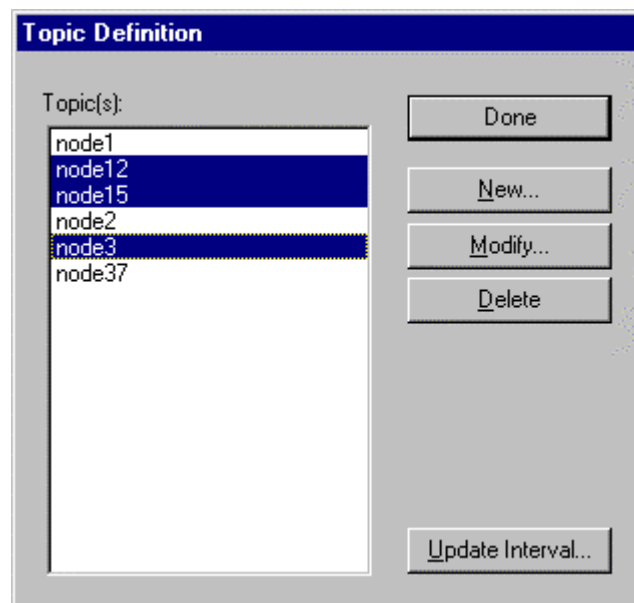


3. Enter the **Topic Name**.  
**Note:** If using *InTouch* the same Topic Name is to be entered in the "Add Access Name" dialog box described in the *Using the ABBCS31 Server with InTouch* section.
4. Click on the **Comm Port** button to associate a communication port with the topic.  
**Note:** Additional topics may be associated with the same communication port later.

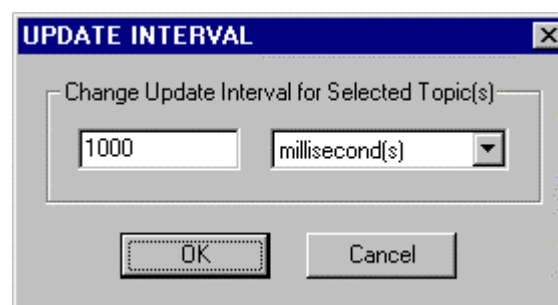
5. Select the PLC type.  
**Note:** Default setting KT93 is usefull when access data from 07 KR91 PLC.
6. Set the Update **Interval** field to indicate the frequency the items/points on this topic will be read (polled). Default value is 1000 milliseconds.
7. Click on the **Ok** button.  
**Note:** If this is the first time a communication port has been configured, the user will be prompted to save it in an existing directory.

### Update Interval for Multiple Topics

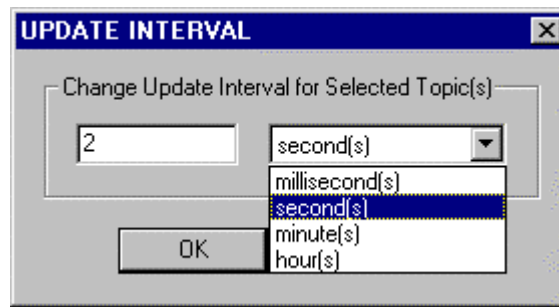
The ABBCS31 Server supports the possibility to enter the new **Update Interval** value for multiple Topics in a single operation. At first these multiple Topics must be selected in the "Topic Definition" dialog box by holding the CTRL key while clicking on the selected ones:



Then click on **Update Interval...** . The "UPDATE INTERVAL" dialog box will appear:



Enter the new value in the **Update Interval** field and select the appropriate time measuring units (milliseconds, seconds, minutes or hours):



Click on **OK** to change the **Update Interval** value for all selected Topics or click on **Cancel**. The "Topic Definition" dialog box will appear again.

The **Update Interval** can be changed also at run-time (for each Topic separately, not for several ones at once) after Topic is activated by client application (see description of **UPDATEINTERVAL** item in *Item (Point) Naming* section).

Select **Done** when configuration for all Topics has been performed.

## Item (Point) Naming

Within the ABBCS31 Communication Server, item/point naming depends on ABB Procontic CS31 system naming conventions for operands (variables). All item/point names (except special Items, e.g. STATUS, FORCR) generally may be described as:

### FNg,c:bs

where **F** - optionally used prefix (character **F** or **f**) for forced item indication;  
**N** - one or two characters identifying the name of item;  
**g** - group number;  
**c** - channel number;  
**b** - bit number for items representing Bit of Word (Double Word) Items;  
**s** - optionally used suffix (character **U** or **u**) for unsigned Word indication; this suffix can be used only for EW, AW, MW and KW Word items.

The tables below list the item/point names supported by the ABBCS31 Communication Server.

## ABB CS31 07 KT93 Controllers

### BINARY points

Item name	Description	Tag Type	Range
<b>E</b>	BINARY input	Discrete	E00,00...E63,15
<b>A</b>	BINARY output	Discrete	A00,00...A63,15
<b>M</b>	BINARY flag	Discrete	M00,00...M255,15
<b>S</b>	BINARY step	Discrete	S00,00...S127,15
<b>K</b>	BINARY Indirect constant	Discrete	K00,00...K00,01

### WORD&DOUBLE points

Item name	Description	Tag Type	Range
<b>EW</b>	WORD input	Integer	EW00,00...EW07,15
<b>AW</b>	WORD output	Integer	AW00,00...AW6,15
<b>MW</b>	WORD flag	Integer	MW00,00...MW255,15
<b>KW</b>	WORD Indirect constant	Integer	KW00,00...KW39,15
<b>MD</b>	DOUBLE WORD flag	Integer	MD00,00...MD31,15
<b>KD</b>	DOUBLE WORD indirect constant	Integer	KD00,00...KD07,15

In most commonly used cases item/point names are without prefixes. Prefix "F" may be used for BINARY Inputs, WORD Inputs, BINARY Outputs and WORD Outputs and means that the point is forced. Forced points are only written to the PLC and no polling is performed on them. These forced values apply until the ABBCS31 Server is started next time.

**FORCED points**

Item name	Description	Tag Type	Range
<b>FE</b>	FORCED BINARY input	Discrete	E00,00...E63,15
<b>FA</b>	FORCED BINARY output	Discrete	A00,00...A63,15
<b>FEW</b>	FORCED WORD input	Integer	EW00,00...EW07,15
<b>FAW</b>	FORCED WORD output	Integer	AW00,00...AW6,15

**ABB CS31 07 KT94 Controllers****BINARY points**

Item name	Description	Tag Type	Range
<b>K</b>	BINARY indirect constant	Discrete	K00,00...K00,01
<b>E</b>	BINARY input	Discrete	E00,00...E163,15 E200,00...E263,15
<b>A</b>	BINARY output	Discrete	A00,00...A163,15 A200,00...A263,15
<b>M</b>	BINARY flag	Discrete	M000,00...M511,15
<b>S</b>	BINARY step	Discrete	S00,00...S255,15

**WORD points**

Item name	Description	Tag Type	Range
<b>KW</b>	WORD indirect constant	Integer	KW00,00...KW89,15
<b>EW</b>	WORD input	Integer	EW00,00...EW34,15 EW100,00...EW107,15 EW200,00...EW207,15
<b>AW</b>	WORD output	Integer	AW00,00...AW34,15 AW100,00...AW107,15 AW200,00...AW207,15
<b>MW</b>	WORD flag	Integer	MW000,00...MW511,15

**DOUBLE WORD points**

Item name	Description	Tag Type	Range
<b>KD</b>	DOUBLE WORD indirect constant	Integer	KD00,00...KD23,15
<b>MD</b>	DOUBLE WORD flag	Integer	MD00,00...MD63,15

In most commonly used cases item/point names are without prefixes. Prefix "F" may be used for BINARY Inputs, WORD Inputs, BINARY Outputs and WORD Outputs and means that the point is forced. Forced points are only written to the PLC and no polling is performed on them. These forced values apply until the ABBCS31 Server is started next time.

## FORCED points

Item name	Description	Tag Type	Range
<b>FE</b>	FORCED BINARY input	Discrete	E00,00...E163,15 E200,00...E263,15
<b>FA</b>	FORCED BINARY output	Discrete	A00,00...A163,15 A200,00...A263,15
<b>FEW</b>	FORCED WORD input	Integer	EW00,00...EW34,15 EW100,00...EW107,15 EW200,00...EW207,15
<b>FAW</b>	FORCED WORD output	Integer	AW00,00...AW34,15 AW100,00...AW107,15 AW200,00...AW207,15

## Common features of all Controller's Item (Point) Naming

The forced values apply until client application sets value of Discrete item **FORCR** to 1.

The maximum number of I/O Items to be forced into one controller are the following:

Binary inputs - 64;

Word inputs - 16;

Binary outputs - 64;

Word outputs - 16.

If the limit is exceeded by client application, then it must change the value of **FORCR** Item to 1 before performing next forced writes. It is recommended to write value 1 into FORCR item for every controller after Server's startup, because the Server can not recognize the number of previously forced items (previously forced by some other means) and will try to force more items than it is allowed.

### Notes:

1. Items/points **E**, **S** and **EW** are Read Only.

2. When **writing** or **forcing** new values to outputs, the outputs of the PLC central unit will be physically accessed (state or value changed) only if corresponding operand presents in the PLC program's instructions or block calls (not enough if presents only in Variables List).

3. It is **highly recommended** to use **consecutive addressing** for item/point naming (for example, MW1,1, MW1,2, MW1,3, etc.). This will greatly increase the performance of ABBCS31 Server.

4. Range of Bit number in items representing Bits: from 0 to 15 – in Words, from 0 to 31 in Double Words.

## Value range

The range of values is the following:

binary points 0,1;

words -32768 ... +32767;

double words -2147483648 ... +2147483647.

## Signed and Unsigned Words

Values of Word items (EW, AW, MW and KW) are processed by the Server as signed integers in range -32767 ... 32767 (8001 ... 7FFF hex ). The value -32768 (8000 hex) is inadmissible value. The same range presents in Server - PLC communication. Therefore

the Server transfers the value received from client application to PLC without converting. Also, the value received from PLC is transferred to client application without converting.

If item names with suffix u (U) are used then client application can process values in range 0 ... 65535 ( 0 ... FFFF hex). The value 32768 (8000 hex) is inadmissible value. When Server receives a value from client application, it converts the received value to range -32767 ... 32767 (8001 ... 7FFF hex ) and transfers to the PLC the converted value. For example, if client application assigns the value **32769** to item **MW100,04U**, then value 32769 (8001 hex) is transferred to the Server. The Server converts it and transfers to the PLC the value -32767 (8001 hex) - the PLC does not accept the value 32769. On the other hand, when polling the item **MW100,04U** and if value **-32767** is received from PLC, this received value will be converted by the Server to the value 32769. The converted value (32769) is transferred to the client application.

## Item/Point Naming Examples

The following examples show the correct format for item/point names:

E01,15 - BINARY Input (CS31 bus), Group number 1, Channel number 15.  
E62,00 - Direct BINARY Input, Group number 62, Channel number 0.  
AW6,02 - WORD Output, Group number 6, Channel number 2.  
MW9,8:6 - The 6th bit in the WORD Flag, Group number 9, Channel number 8.  
MW9,8u - WORD Flag in value range 0 ... 65535, Group number 9, Channel number 8.  
MD31,10 - DOUBLE WORD flag, Group number 31, Channel number 10.  
KD7,8 - DOUBLE WORD indirect constant, Group number 7, Channel number 8.  
FE62,03 - forced BINARY Input, Group number 62, Channel number 3.

## Monitoring and Controlling Communication with a PLC

For each topic, there are following additional items offered by ABBCS31 Communication Server to monitor and control the communication with PLC.

### STATUS

For each topic, there is a built-in discrete item that indicates the state of communication with PLC. The discrete item (**STATUS**) is set to **0** when communication fails and set to **1** when communication is successful.

From **InTouch** the state of communication may be read by defining an I/O Discrete tagname and associating it with the topic configured for the PLC and using **STATUS** as the item name.

From **Excel**, the status of the communication may be read by entering the following formula in a cell:

**=ABBCS31|topic!STATUS**

### UPDATEINTERVAL

The **UPDATEINTERVAL** item is an Integer type Read/Write item used to access the currently set Update Interval (see Topic Definition Command). It indicates the current requested update interval (in milliseconds). The value of this item can be read through

DDE or Suite Link. Client can poke new values to this item. The range of valid values is from 10 to 2147483647 milliseconds. The value of zero indicates that no items on this topic are updated. The write commands are still executed (new values written to PLC) if **UPDATEINTERVAL** value is 0.

***Note:** By poking a value of zero to the **UPDATEINTERVAL** item, a client can stop all update activities on the corresponding topic without having to deactivate the items.*

### **MAXINTERVAL**

The **MAXINTERVAL** item is an Integer type Read Only item used to access the measured maximum update interval (in milliseconds) of all items for the corresponding topic for the last completed poll cycle. The range of valid values is from 0 to 2147483647 milliseconds.

The **UPDATEINTERVAL** and **MAXINTERVAL** items can be used to tune the performance of communication.

### **ITEMCOUNT**

The **ITEMCOUNT** item is an Integer type Read Only item used to access the number of active items in the corresponding topic. The range of valid values is from 0 to 2147483647.

### **ERRORCOUNT**

The **ERRORCOUNT** item is an Integer type Read Only item used to access the number of active items with errors in the corresponding topic. The range of valid values is from 0 to 2147483647.

### **ERRORITEMS**

The **ERRORITEMS** item is an Integer type Read/Write Only (unique for each topic) used to access the total number of items with invalid item names (these items are rejected by Server). The **ERRORITEMS** value can be reset by writing 0 to this item. The range of valid values is from 0 to 2147483647.

### **WRITECOUNT**

The **WRITECOUNT** item is an Integer type Read Only item used to access the number of write commands (messages) waiting for execution. The range of valid values is from 0 to 2147483647.

For example, in following way the **WRITECOUNT** item can be used to avoid the increasing of memory occupied by not executed write commands:

- activate the hot link with **WRITECOUNT** item and start to monitor it;
- activate new write command (by poking new value) only if value of **WRITECOUNT** becomes equal to 0, e.g. all previous write commands are executed and memory occupied by them is freed.

### **SUSPEND**

Special Read/Write Discrete Item **SUSPEND** may be used to control the communication with a separate topic. If application changes **SUSPEND** value from 0 to 1 then communication with topic is suspended. If **SUSPEND** value is changed back to 0 then communication with this topic is resumed.

**Note:** If topic is suspended by setting **SUSPEND** value to 1, then Server rejects all new write values to this topic, i.e. no new write messages are created after **SUSPEND** value has changed from 0 to 1.

## Using the ABBCS31 Server with OPC Clients

The "OPC & DDE" version of ABBCS31 Server is accessible from OPC Clients.

There are the following general steps needed to access an OPC item from ABBCS31 Server:

1. Run OPC Client application and select the "ABBCS31 OPC and DDE Server" from the list of available OPC Servers. If ABBCS31 Server currently is not running, it will start automatically.
2. Create a new group (or topic if Wonderware OPCLink application is used).
3. If OPC Client supports the validating of items, validate the item before adding it.
4. Add the item. Depending on OPC Client it can be done in several different ways, for example:
  - a) By entering separately the access path to topic name (valid topic name configured in ABBCS31 Topic definition) and separately the item name.
  - b) By entering the full path to item name in the format **TopicName.ItemName** where **TopicName** is the valid topic name configured in ABBCS31 Topic definition.
  - c) By browsing the server address space.

By default the ABBCS31 Server is installed and used as a local OPC Server - both OPC Server and OPC Client reside on same computer. The ABBCS31 Server can run also as a remote OPC Server – in this case OPC Server and OPC Client are located on separate computers. Accessing the remote OPC Server is same as for local OPC Server, but some DCOM (Distributed COM) configuration is required before accessing the remote OPC Server. The DCOM configuration must be done both on OPC Server and OPC Client computers.

### Configuring DCOM

To access ABBCS31 Server as a remote OPC Server, it is necessary to install ABBCS31 Server on both (OPC Server and OPC Client) computers. Also the DCOM must be configured on both computers.

After Server installation the System Administrator must configure DCOM by **Dcomcnfg.exe** (or Dcomcnfg32.exe for Win9x) system tool. This utility is located in the Windows system directory – e.g. in \WinNT\system32\ or \Win9x\system\.

Below is a simple example how to configure DCOM on NT Workstations for OPC Server computer (computer name *JohnB*) and on OPC Client computer (computer name *SteveL*).

Action	Remote OPC Server Computer name – <i>JohnB</i>	OPC Client Computer name – <i>SteveL</i>
Install the OPC Server.	✓	✓
Run Dcomcnfg.exe	✓	✓
Invoke <b>Default Properties</b> and set <b>Default Authentication Level</b> to <i>(None)</i> , <b>Default Impersonation Level</b> to <i>Impersonate</i> .	✓	✓
Select OPC Server from <b>Applications</b> list and click on the <b>Properties...</b> button. Click on the <b>Location</b> tab, uncheck <b>Run application on this computer</b> , check <b>Run application on the following computer</b> and browse the remote computer <i>JohnB</i> .		✓
Edit <b>Security</b> settings: 1) Set the following <b>custom access permissions</b> : NETWORK, SYSTEM, <i>SteveL</i> ; 2) Set the following <b>custom launch permissions</b> : INTERACTIVE, SYSTEM, NETWORK, <i>SteveL</i> ; 3) be sure the <b>Default Configuration Permissions</b> contain SYSTEM.	✓	
Click on the <b>Identity</b> tab and select <b>The interactive user</b> .	✓	

Before starting a remote OPC Server, be sure the OPC Server computer and OPC Client computer can access each other on the network. The "Remote Procedure Call" Service should be started on OPC Server computer.

Now remote OPC Server is accessible for OPC Client on computer *SteveL*. To allow the access to more OPC Clients, configure DCOM on each OPC Client computer.

The following table contains most frequent errors when configuring DCOM.

Error message	Possible reason
DCOM not installed	DCOM has not been installed
Server execution failed	<ol style="list-style-type: none"> <li>1) OPC Server can not be started</li> <li>2) <b>Identity</b> for OPC server not properly configured</li> <li>3) OPC Server is not located on a local hard disk</li> <li>4) OPC Server path in registry is too long or uses LFN (Long File Names)</li> <li>5) DCOMCNFG <b>Location</b> is not set to <b>Run on this computer</b>.</li> </ol>
Class not registered	OPC Server has not been registered
RPC server is unavailable	<ol style="list-style-type: none"> <li>1) <b>Remote Procedure Call</b> service is not running on the OPC Server computer</li> <li>2) Invalid computer name for remote OPC Server</li> <li>3) Make sure TCP/IP is installed properly</li> </ol>
Interface not supported	<ol style="list-style-type: none"> <li>1) Permission not granted to OPC Client</li> <li>2) Guest account disabled</li> </ol>
Access is denied	<ol style="list-style-type: none"> <li>1) DCOM security not configured properly</li> <li>2) OPC Server application not located on local hard disk</li> <li>3) SYSTEM account in DCOMCNFG must have <b>Access</b>, <b>Launch</b> and <b>Configure</b> privileges</li> </ol>
Error 80070776	Network error - TCP/IP has not been configured properly
Catastrophic failure	<ol style="list-style-type: none"> <li>1) Trying to access an object before it is created</li> <li>2) Unhandled exception is occurs on the OPC Server</li> </ol>
Not enough storage	SYSTEM account in DCOMCNFG must have <b>Access</b> , <b>Launch</b> and <b>Configure</b> privileges

## Running ABBCS31 "OPC & DDE" version as Windows NT Service

To install ABBCS31 Server "OPC & DDE" version to run as **Windows NT Service**, the ABBCS31 Server must be started with command line parameter `"/Service"`:

### ABBCS31 /Service

After this the "ABBCS31 OPC & DDE Server" NT Service will be installed with Startup type "Manual". The Service Startup configuration can be changed by MS Windows NT *Control Panel/Services* configuration dialog boxes. The **Allow Service to Interact with Desktop** checkbox in "Service" dialog box must be checked (the "Service" dialog box can be invoked by pressing the "Startup" button on "Services" dialog box when Service **ABBCS31 OPC & DDE Server** is selected). If **Allow Service to Interact with Desktop** is not selected then ABBCS31 Server full functionality is not ensured (e.g. the Server configuration can not be changed, no message boxes will be displayed, etc.).

To use ABBCS31 Server "OPC & DDE" version as Windows NT Service you may need to configure DCOM. For details of configuring DCOM refer to *Configuring DCOM* section of

this manual. If "ABBCS31 OPC & DDE Server" NT Service will be accessed only from local OPC clients (i.e. ABBCS31 Server will not be used as a remote OPC Server), then **custom access** and **launch permissions** "NETWORK" are not needed.

To uninstall "ABBCS31 OPC & DDE Server" NT Service, at first the Service must be stopped by *Control Panel/Services/Stop* and then ABBCS31 Server must be started manually with command line parameter `"/DelService"`:

### **ABBCS31 /DelService**

After this the ABBCS31 Server "OPC & DDE" version will be still registered and accessible to OPC clients.

## **Using ABBCS31 with OPCLink Server**

*The Wonderware OPCLink I/O Server (hereafter referred to as "OPCLink") is a Microsoft Windows application program that acts as a communication protocol converter and allows other Windows application programs access to data from local or remote OPC servers. OPCLink connects to OPC servers, converts client commands to OPC protocol and transfers data back to clients using DDE, FastDDE, or Suite Link protocols.*

Please refer to **Wonderware OPCLink Server and OPC Browser User's Guide** for details how to install, start and use the OPCLink Server. The following information in this section covers only the most important points about using "OPC & DDE" version of ABBCS31 Server with OPCLink Server.

### **OPCLink Topic Definition**

The **Topic Definition** option from OPC Link Configure menu is used to create, modify, or delete OPCLink topic definitions. If OPC Link will communicate with ABBCS31 Server then there must be one or more topics defined for ABBCS31 Server. There are following important fields on the "OPCLink Topic Definition" dialog box:

#### **Topic Name**

Enter a unique name (e.g. **PLC1**) for the PLC in this field. If using InTouch then same Topic Name is to be entered in the "Add Access Name" dialog box when defining the Access Names for OPCLink Server in InTouch WindowMaker.

#### **OPC Server Name**

Select the name of the OPC server (**ABBCS31.OPC\_Server**) that will be used by this topic. The list box shows the registered OPC servers in the system.

#### **OPC Path**

Enter the name of the OPC path (e.g. **node2.**) used by this topic. This OPC path is the first part of a full OPC item name string common to all items that will be used in this topic. The available OPC paths for ABBCS31 Server can be obtained by clicking on "Browse" button (this allows to view the ABBCS31 Server's exposed address space).

### Update Interval

Enter the frequency (in milliseconds) that the server will acquire data for the items/points associated with this topic. If 0 (zero) is entered here, OPCLink will not gather data from ABBCS31 Server.

### Browse

Clicking on this button initiates the browsing through exposed address space of ABBCS31 Server. All currently available item/point names (including data area names and range of addresses) and names of pre-defined (additional) items will appear on "Browse OPC items:" window in alphabetical order.

## Accessing ABBCS31 Items via the OPCLink Server

The communication protocol addresses an element of data in a conversation that uses a three-part naming convention that includes the **application name**, **topic name** and **item name**. The following briefly describes each portion of this naming convention:

### **application name**

The name of the Windows program (server) that will be accessing the data element. In the case of data coming from or going to ABBCS31 Server "OPC & DDE" version, the application portion of the address is **OPCLINK**.

### **topic name**

Meaningful names are configured to identify specific devices (PLCs). These names are then used as the topic name in all conversations to that device (PLC). This must be same name as **Topic Name** entered in the "OPCLink Topic Definition" dialog box, for example, **PLC2**.

**Note!** You can define multiple topic names for the same PLC to poll different points at different rates.

### **item name**

A specific data element within the specified topic. The OPCLink Server item syntax follows the following rules. The item names must start with:

- d – discrete value
- i – integer value
- r – real value
- m – message value

The item name added to the OPC path of the topic (without the heading type letter) must give a fully qualified OPC item name for the ABBCS31 Server. Some examples of possible item names acceptable by OPCLink Server/ABBCS31 Server connection:

#### **DE62,02**

Discrete  
BINARY input, group number 62, channel number 2;

#### **iAW10,00**

Integer  
WORD output, group number 10, channel number 00.

## Using the ABBCS31 Server with Suite Link and DDE Clients

The "**Suite Link & DDE**" version of ABBCS31 Server is accessible from Suite Link clients (e.g. InTouch) and DDE clients (e.g. Excel). The "**OPC & DDE**" version of ABBCS31 Server is accessible from DDE clients.

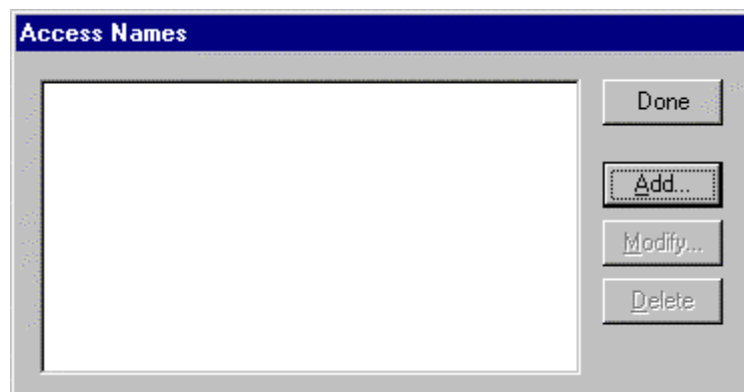
## Using the ABBCS31 Server with InTouch

To access to operands on the ABB Procontic CS31 PLC from **InTouch**, the Access Names and Tag names should be defined in **WindowMaker**.

### Defining the Access names

InTouch uses **Access Names** to reference real-time I/O data. Each Access Name equates to an I/O address, which can contain a **Node**, **Application**, and **Topic**. In a distributed application, I/O references can be set up as global addresses to a network I/O Server or local addresses to a local I/O Server.

To define the Access Names in WindowMaker node invoke the */Special/Access Names...* command. The "Access Names" dialog box will appear.



Click on **Add....** The "Add Access Name" Dialog Box will appear:

The screenshot shows a dialog box titled "Add Access Name". It has the following fields and options:

- Access Name:** A text box containing "PLC1".
- Node Name:** An empty text box.
- Application Name:** A text box containing "ABBCS31".
- Topic Name:** A text box containing "PLC1".
- Which protocol to use:** Two radio buttons: "DDE" (unselected) and "SuiteLink" (selected).
- When to advise server:** Two radio buttons: "Advise all items" (unselected) and "Advise only active items" (selected).
- Buttons:** "OK" and "Cancel" buttons are located in the top right corner.

**Note:** If **Add** is selected, this dialog box will be blank when it initially appears. Data has been entered here to illustrate the entries that are made.

The following fields are required entries when entering an Access Name Definition:

### Access Name

In the Access Name box type the name you want InTouch to use to this Access Name. (For simplicity, use the same name that you will use for the **Topic Name** here.)

### Node Name

If the data resides in a network I/O Server, in the Node Name box, type the remote node's name.

### Application Name

In the Application Name box, type the actual program name for the I/O Server program from which the data values will be acquired. In case the values are coming from the ABBCS31 Server the ABBCS31 is used. Do not enter the .exe extension portion of the program name.

### Topic Name

Enter the name defined for the topic in the ABBCS31 Server to identify the topic the ABBCS31 Server will be accessing.

The Topic Name is an application-specific sub-group of data elements. In the case of data coming from a ABBCS31 Server program, the topic name is the exact same name configured for the topic in the ABBCS31 Server.

**Note:** This will usually be the same as the "Access Name", although, if desired, they may be different. However, it must be the same name used when the topics were configured in section **Configuring the ABBCS31 Server**.

### Which protocol to use

Select the protocol (DDE or Suite Link) that you are using.

### When to advise server

Select **Advise all items** if you want the Server program to poll for all data whether or not it is in visible windows, alarmed, logged, trended or used in a script. Selecting this option will impact performance, therefore its use is not recommended.

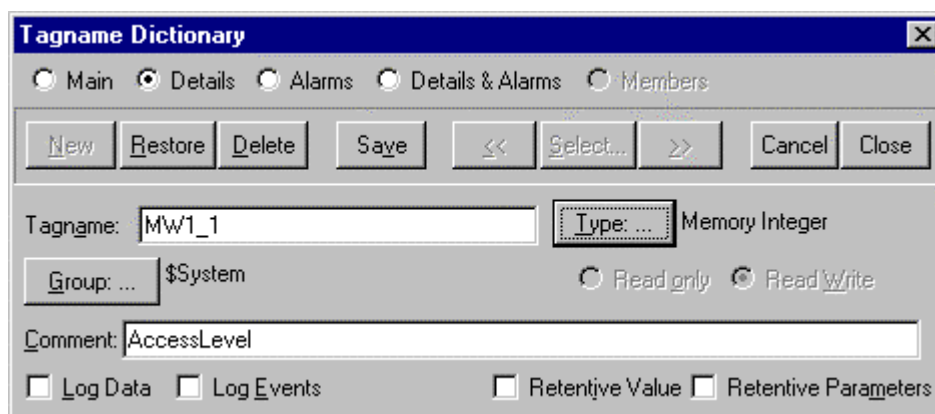
Select **Advise only active items** if you want the Server program to poll only points in visible windows and points that are alarmed, logged, trended or used in any script.

Click **OK** to accept the new Access Name and close the "Add Access Name" dialog box. The "Access Names" dialog box will reappear displaying the new Access Name selected in the list.

Click **Close** to close the "Access Names" dialog box.

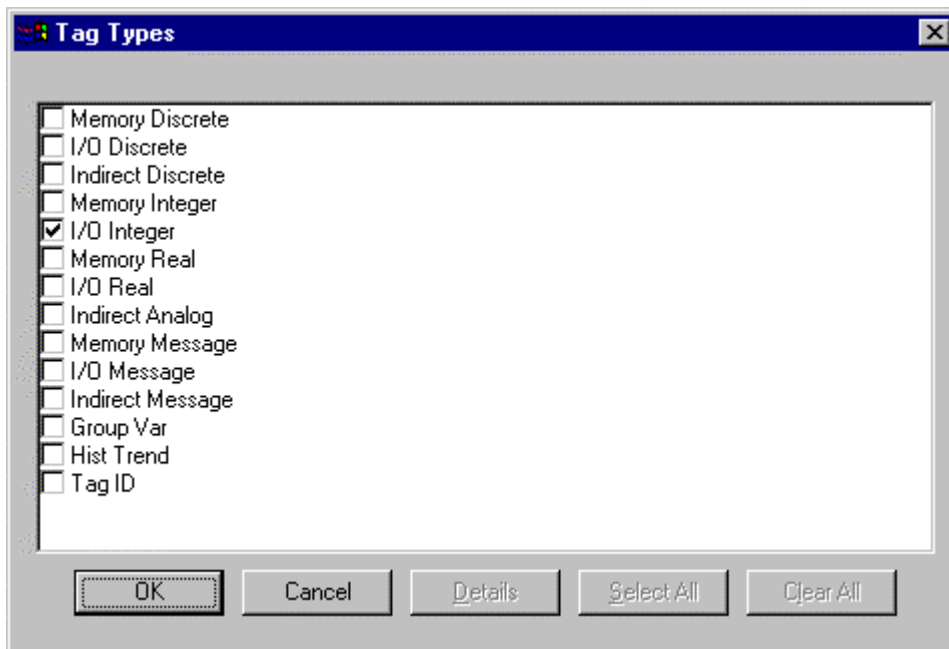
### Defining the Tag names

To define the Tag names associated with the new "Access Name", invoke the */Special/Tagname Dictionary...* command (in **WindowMaker**). The "Tagname Dictionary" dialog box will appear:



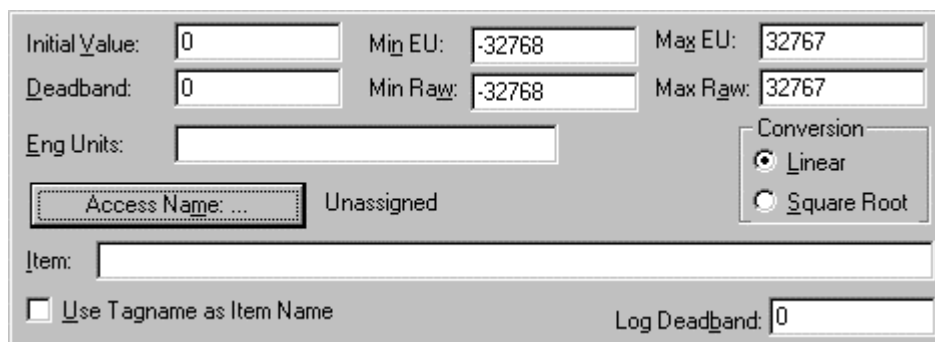
Click on **New** and enter the **Tagname**. (The tagname defined here is the name **InTouch** will use. The ABBCS31 Server does not see this name.)

Select the tag type by clicking on the **Type:...** button. The "Tag Types" dialog box will appear:

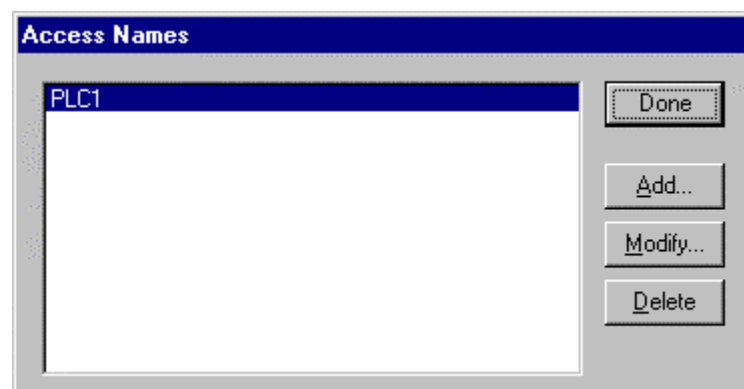


To access ABBCS31 items, the type must be **I/O Discrete** or **I/O Integer**. Select the Tag type.

The "Details" dialog box for the tag name will appear:



Select the Access name for ABBCS31 Server by clicking on the **Access Name:...** button. The "Access Names" dialog box will appear:



Select the appropriate Access Name and click on **Close**. (If the Access Name has not been defined as previously described, click on **Add** and define the Access Name now.)

The "Details" dialog box will appear displaying the selected Access Name:

For integers fill in the **Min EU**, **Max EU**, **Min Raw** and **Max Raw** fields. These fields control the range of values that will be accepted from the Server and how the values are scaled. If no scaling is desired, **Min EU** should be equal to **Min Raw** and **Max EU** equal to **Max Raw**.

Enter the ABBCS31 item name to be associated with this tagname in the **Item**: field in the "Details" box:

(Refer to the **Item (Point) Naming** section below for complete details.)

Where applicable, the **Use Tagname as Item Name** option may be selected to automatically enter the tag name in this field. **Note:** *The tag name can only be used if it follows the conventions listed in the **Item (Point) Naming** section.*

Once all entries have been made, click on the **Save** button (in the top dialog box) to accept the new tag name. To define additional tagnames click on the **New** button. To return to the **WindowMaker** main screen, select **Close**.

## Monitoring the Status of Communication with InTouch

**InTouch** supports built-in topic names called **DDEStatus** and **IOStatus** that are used to monitor the status of communications between the Server and InTouch. For more information on the built-in topic names DDEStatus and IOStatus, see your online "InTouch User's Guide".

The status of communication between the Server and InTouch can be read into **Excel** by entering the following DDE reference formula in a cell on a spreadsheet (in following examples **plc01** is the Topic Name configured for ABBCS31 Server):

```
=view|DDEStatus!plc01
```

or

```
=view|IOStatus!plc01
```

## Notes on Using Microsoft Excel

Data from the ABBCS31 topic (PLC) may be accessed from Excel spreadsheets. To do so, enter a formula like the following into a cell on the spreadsheet.

```
=ABBCS31|topic!item
```

Sometimes, Excel requires the **topic** and/or **item** to be surrounded by apostrophes.

In the formula, **topic** must be replaced with one of the valid topic names defined during the Server configuration process. Replace **item** with one of the valid item names described in the *Item (Point) Naming* section.

## Reading Values into Excel Spreadsheets

Values may be read directly into Excel spreadsheets by entering a DDE formatted formula into a cell, as shown in the following examples:

```
=ABBCS31|'topic-31!'A62,0'  
=ABBCS31|'rack5!'MW1,1'  
=ABBCS31|'tank-2!'E62,2'
```

**Note:** Refer to the Microsoft Excel manual for complete details on entering Remote Reference formulas for cells.

## Writing Values to ABBCS31 Points

Values may be written to the Server from Microsoft Excel by creating an Excel macro that uses the **POKE** command. The proper command is entered in Excel as follows:

```
channel=INITIATE("ABBCS31","topicname")  
=POKE(channel,"itemname", Data_Reference)  
=TERMINATE (channel)  
=RETURN()
```

The following describes each of the above **POKE** macro statements:

```
channel=INITIATE("ABBCS31","topicname")
```

Opens a channel to a specific topic name (defined in the Server) in an application with name ABBCS31 (the executable name less the .EXE) and assigns the number of that opened channel to **channel**.

**Note:** By using the **channel=INITIATE** statement the word **channel** must be used in the **=POKE** statement instead of the actual cell reference. The "**applicationname**" and "**topicname**" portions of the formula must be enclosed in quotation marks.

#### **=POKE(channel,"itemname", Data\_Reference)**

**POKEs** the value contained in the **Data\_Reference** to the specified item name (actual location in the ABB CS31 PLC) via the **channel** number returned by the previously executed **INITIATE** function. **Data\_Reference** is the row/column ID of the cell containing the data value. For "**itemname**", use some of the valid item names described in the **Item (Point) Naming** section.

#### **=TERMINATE(channel)**

Closes the channel at the end of the macro. Some applications have a limited number of channels. Therefore they should be closed when finished. **Channel** is the channel number returned by the previously executed **INITIATE** function.

#### **=RETURN()**

Marks the end of the macro.

The following is an example of Excel macro used to poke value from cell B2 to topic **PLC1** item **MW1,1**:

```
PokeMacro -Ctrl a
=INITIATE("ABBCS31","PLC1")
=POKE(A2," MW1,1",B2)
=ON.TIME(NOW()+0.01,"TerminateDDEChannel")
=RETURN()
```

```
TerminateDDEChannel
=TERMINATE(A2)
=RETURN()
```

**Note:** Refer to the Microsoft Excel manual for complete details on entering Remote Reference formulas for cells.

## Troubleshooting

### WIN.INI entries

The first time you run the ABBCS31 Server configuration, most of the items in the following list will automatically appear in the WIN.INI file, located in the MS Windows system directory (e.g. C:\WINNT). It is an ASCII file and can be altered manually if you wish with any text editor, e.g., MS Windows Notepad (*do not use a program that formats text, such as MS Word or Write unless the file is saved as a DOS text*). The following is a typical entry for the ABBCS31 Server:

```
[ABBCS31]
WinIconic=0
WinFullScreen=0
WinTop=112
```

**WinLeft=0**  
**WinWidth=200**  
**WinHeight=168**  
**ProtocolTimer=50**  
**ConfigurationFile=C:\ABBCS31\**  
**DumpScreen=1**  
**ShowSend=0**  
**ShowReceive=0**  
**ShowErrors=1**

There are following additional WIN.INI entries available for ABBCS31 Server:

The **SlowPollRetries** entry is used to enter the number of consecutive error retries for one topic (PLC). If after **SlowPollRetries** there is still no successful response from PLC, then this topic is changed to *slow poll mode*. The WIN.INI file **SlowPollInterval** entry is used to enter the slow poll mode update interval (in seconds).

The default values (they are used if WIN.INI file does not contain these entries) are **SlowPollRetries** equal to 3 and **SlowPollInterval** equal to 60 seconds.

For example, the following entries can be used to specify that slow poll mode 2 minutes will start after 3 consecutive unsuccessful retries:

**SlowPollRetries =3**  
**SlowPollInterval=120**

Entering into slow poll mode is reported to Wonderware Logger by following string:

**"Set slow mode - poll after each <SlowPollInterval> msec. Stop error logging to topic <TOPICNAME>"**

Leaving the slow poll mode is reported to Wonderware Logger by following string:

**"Return to normal communication to topic <TOPICNAME>. Error logging reestablished."**

The **WriteRetryIndefinitely** entry is used to disable the deleting of pending write messages when *slow poll mode* is started. As default (no **WriteRetryIndefinitely** entry in WIN.INI or **WriteRetryIndefinitely=0**) all write messages for this topic are deleted when topic enters the slow poll mode. To disable this the following entry can be used:

**WriteRetryIndefinitely=1**

**Note!** Be careful when using this setting if PLC is switched off but client application continues to generate new values to be written to this PLC - it can cause the computer memory overflow with memory allocated for write messages.

The **ConsecutiveWriteErrors** entry is used to enter the number of consecutive unsuccessful write retries for one write command. If after **ConsecutiveWriteErrors** the write command still is not executed then write command is deleted from the list of active

(pending) write messages and write is finally rejected. The unsuccessful bit writing is performed **ConsecutiveWriteErrors** + 1 times. If the total amount of messages in the list of active write messages exceed 300 then every write message is performed only once not taking into account the **ConsecutiveWriteErrors** value.

The default value is:

**ConsecutiveWriteErrors=3**

## Troubleshooting menu

The following debugging choices are appended to the Server's System Menu (the menu that appears when you click on the Server icon in the upper left hand corner of the Server's window):

- Suspend Protocol/Resume Protocol** - these choices permit you to turn protocol processing on and off, what means that you can suspend access to PLC(s).
- ShowEvents** - if checked then Server logs the information about situations when the total number of active (pending) write messages exceeds 300. It may happen when the Server has timing problems when executing write commands. When such warnings appear then it is recommended to modify the Server timing settings or to reduce the total amount of write data in the client application. Otherwise the possibility of losing write data increases.  
Information about suspended and resumed topics also is displayed if ShowEvents is checked.
- Show Send** - if checked then all outgoing user data is logged.
- Show Receive** - if checked then all incoming user data is logged.
- Show Errors** - if checked then all information about errors is logged.
- ShowBadWrites** - if checked then Server displays information about all situations when data write commands are not executed successfully and after **ConsecutiveWriteErrors** retries the write is rejected and write message is deleted from list of active (pending) write messages. This option (if checked) is effective even if other error logging is stopped.
- ShowAllRejectedWrites** - option is useful when communication with a separate topic (topics) is suspended by Item SUSPEND (see **Item (Point) Naming** section) and the Server rejects each write to this topic (topics). If checked then information about each rejected write value is reported to WWLogger or (and) to ABBCS31 Internal Logger. If not checked (default) then Server rejects each write to suspended topic(s) without logging any information.
- Verbose** - if checked then all information about errors (except errors indicated in other choices) is logged. This option is useful for getting additional information about error situations:
- Dump** - all information about ports, topics, messages and data items will be logged once. This can be use to find out how many messages are actually sent to the PLC(s).
- Dump Screen** - if checked then information about active messages are displayed on the ABBCS31 main window. This also can be used to find out how many messages are actually sent to the PLC(s).

**Show Logger** - if checked then ABBCS31 Internal Logger is activated and all debug information is going to ABBCS31 Internal Logger. The ABBCS31 Internal Logger file is named in the format:  
**ABBCS31\_YYYYMMDD.LOGn**  
where **YYYY** is a year, **MM** is a month, **DD** is a day and **n** is a order number of consecutive ABBCS31 Internal Logger file, starting from 1 (the ABBCS31 Internal Logger file maximum size is 16 MB; if there is more information logged then next consecutive file is created, e.g. there can be consecutive files ABBCS31\_19990413.LOG1, ABBCS31\_19990413.LOG2, etc.).

All debugging information (except **DumpScreen** option) is displayed via the Wonderware Logger or (and) to ABBCS31 Internal Logger if Show Logger checked, which must be active for these commands to work.

**Note:** If you check **Show Send** and/or **Show Receive** and/or **Verbose**, the debug output grows very fast and it is possible that computer can become very slow.

KLINKMANN AUTOMATION  
ABBCS31 Communication Server  
Revision History

Jan 94	Rev 1.0	First Release
Aug 94	Rev 1.1	Some grammatical corrections and the cable diagram added
Sep 96	Rev 1.2	NT version added. Dialogue boxes changed.
Aug 97	Rev 1.3	Windows 95 support added. Manual file name changed.
Oct 98	Rev 1.4	Baud Rate settings and Unsigned Word Items added.
Mar 99	Rev 1.5	Item (Point) naming section modified.
Jan 2000	Rev 1.6	SuiteLink and OPC version.
Jan 2001	Rev 1.7	Item (Point) naming section modified.
Jan 2001	Rev 1.8	OPC compliance information added.
Mar 2002	Rev 1.9	Installation from CD information added.