ABB T200 Communication Server

for Microsoft Windows
and InTouch Applications

User Manual
Ver 1.x Rev 1.2
DR 370 10

KLINKMANN AUTOMATION
P.O. Box 38
FIN-00371 Helsinki Finland
tel. int. + 358 9 5404940
fax int. + 358 9 5413541
www.klinkmann.com

Table Of Contents

ABB T200 Communication Server

Overview

The **ABB T200 Communication Server** (hereafter referred to as the “ABBT200 Server” or “ABBT200” or “Server”) is a Microsoft Windows 32-bit application program that acts as a communication protocol *Server* and allows other Windows application programs access to data from ABB Procontic T200 Programmable Logic Controllers (PLCs) using the RS-232C serial interface. The communication is possible with ABB T200 CPU Programming Port or with RS-232C serial port on 07 KP 60 Communication Processor. Any Microsoft Windows program that is capable of acting as a **DDE**, **FastDDE** or **SuiteLink** *Client* may use the ABBT200 Server.

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 for our industry, this extension is being pushed to its performance constraints in distributed environments. The ABBT200 Server “Suite Link & DDE version” 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.

The Suite Link, FastDDE (Version 3) and DDE support for ABBT200 Communication Server is implemented by ***Wonderware I/O Server Toolkit*** ver. 7.0 (060).

Accessing Remote Items via the 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 ABB T200 PLC via this Server, the application portion of the address is **ABBT200**.

topic name
Meaningful names are configured in the Server to identify specific devices (ABB T200 PLCs). These names are then used as the topic name in all conversations to that device. For example, **PLC1**.
Note! You can define multiple topic names for the same device (PLC) to poll different items at different rates.

item name
A specific data element within the specified topic. For example, when using this Server, items can be individual operands (bits, words and double words) in the ABB T200 PLC. The term "point" is used interchangeably with the term "item" in this User Manual. For more information on item names, see the ***Item Names*** section later in this manual.

Installing the ABBT200 Server

Installing the Server

The ABBT200 Server installation package can be supplied:

1. As a self-extracting archive 37010xxx.EXE 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 ABBT200 Server from the self-extracting archive, run the 37010xxx.EXE and proceed as directed by the ABBT200 Server Setup program.

To **install** the ABBT200 Server from CD or distribution disks, on MS Windows (NT, 2000, XP or 9x):

1. Insert the CD with Klinkmann Software into CD drive or insert the ABBT200 Server 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 "ABBT200 SL and DDE Server" and click on "Setup…".
5. Proceed as directed by the ABBT200 Server Setup program.

When installation is finished, the subdirectory specified as a folder where to install the ABBT200 Server files will contain the following files:

ABBT200.EXE	The ABBT200 Server Program. This is a Microsoft Windows 32-bit application program.
ABBT200.HLP	The ABBT200 Server Help file.
ABBT200.CFG	An example configuration file.
LICENSE.TXT	Klinkmann Automation software license file.

To **uninstall** the ABBT200 Server, start Control Panel, select "Add/Remove Programs" and select the "ABBT200 SL 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 ABBT200 Server 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 ABBT200 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 ABBT200 Server 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 ABBT200 Server and these infrastructure files must be install prior to executing the ABBT200 Server.*
3. *The HASP key is needed for full time running of ABBT200 Server. The HASP Driver setup is performed during the Server setup. Without HASP Driver installed the ABBT200 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:

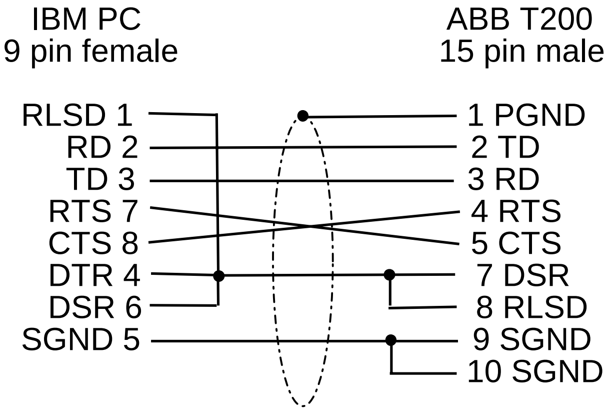
1. Insert the I/O Server Infrastructure disk into a floppy drive A: or B:.
2. Select the **Run** command under the **Start** menu.
3. Type "A:SETUP" or "B:SETUP".
4. Click on **OK**.
5. Proceed as directed by the I/O Server Infrastructure Setup program.

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.*

Connection Cable

To connect the ABB T200 PLC to your computer, you can either use some manufactured cable (for example, the 07 SK 61 R1 or 07 SK 62 R1 type interface cables used for connection to ABB Procontic T200 907 PC 32 programming system), or yourself made connection cable. The following wiring diagram can be used to connect computer's RS-232 serial port and PG port on ABB T200 CPU (e.g. 07 ZE 61) or RS-232C port on 07 KP 60 Communication Processor:



Configuring the ABB T200 PLC

Before starting the ABBT200 Server, the ABB T200 PLC side must be ready to communicate with the Server. Depending on which type of connection (CPU Programming Port or RS-232C port on 07 KP 60) is used, the following must be done:

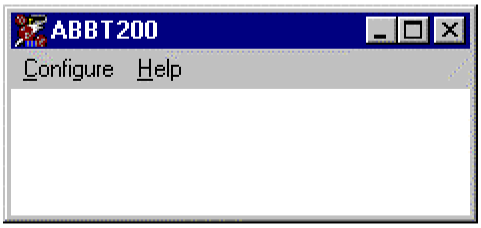
- If Server will communicate with CPU Programming Port (PG) then serial interface parameters for PG must be checked (same parameters must be configured for ABBT200 Server, see *Com Port Settings Command* section); it can be done by “907 PC 332 ABB Procontic Programming and Test Software” at “Programming interface configuration”; usually the default parameters (baud rate 19200 bps, 7 data bits, even parity, 1 stop bit, with checksum) are most commonly used and recommended.
- If Server will communicate with RS-232C port on 07 KP 60 then following settings are needed on 07 KP 60: the MODE switch (operating mode rotary switch) must be set to “1” or “2” (depending on application) and RS-232C Parameter DIL switch settings must match with settings configured for ABBT200 Server, see *Com Port Settings Command* section); usually the default parameters (baud rate 19200 bps, 7 data bits, even parity, 1 stop bit, with checksum - the corresponding positions of RS-232C Parameter switch are: switches 1,7 – OFF, switches 2,3,4,5,6,8 - ON) are most commonly used and recommended.

The *One-way Activation Control Procedures* are used between the ABBT200 Server and ABB T200 PLC, i.e. each of modules in ABB Procontic T200 PLC is activated by an entry of a task code from the ABBT200 Server.

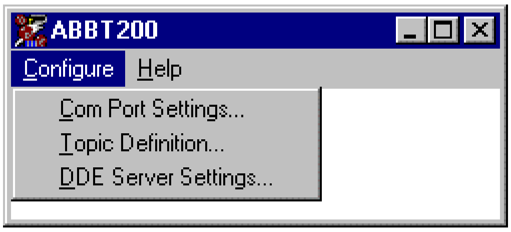
Configuring the ABBT200 Server

After the ABBT200 Server is initially installed, a little of configuration is required. Configuring the Server automatically creates an **ABBT200.CFG** file that holds all of the topics (nodes) definitions entered, as well as the communication port configurations. This file will be placed automatically in the same directory in which **ABBT200** is located unless the path where the configuration file will be placed is specified via the */Configure/Server Settings...* command.

To perform the required configurations, start up the ABBT200 program. If the Server starts up as an icon, double-click on the icon to open the server's window. The following will appear:



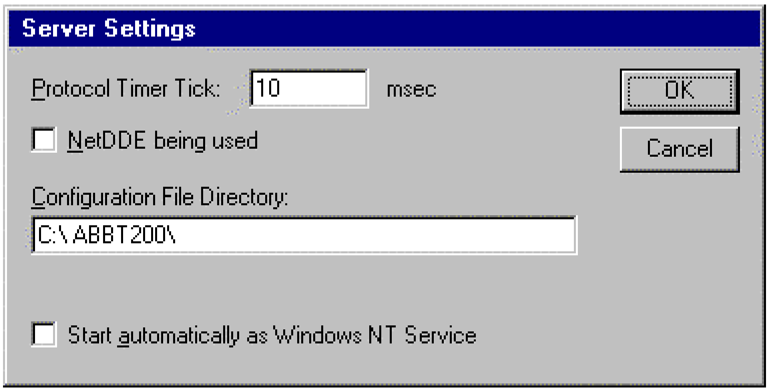
To access the commands used for the various configurations, open the */Configure* menu:



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 *Configure/Server Settings...* command. The "Server Settings" dialog box will appear:


Server Settings

Protocol Timer Tick: 10 msec

☐ NetDDE being used

Configuration File Directory:
C:\ABBT200\

☐ Start automatically as Windows NT Service

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 request command to PLC and receive one reply from PLC). Usually three activations of the Server are needed to process one full data obtaining cycle: at first tick the request with task code is sent to PLC, on next tick the ACK is received from PLC and response request (inquiry) is sent to PLC, and on third tick the response with data is received from PLC. If some step of send/response cycle is taking too long time then more than one activation of Server is needed to process it. If computer is very busy or some other MS Windows application is taking over the computer then Server is activated rarely than setting in the **Protocol Timer Tick**.

Note: *The default and minimum value of* ***Protocol Timer Tick*** *is 10 milliseconds. With this value the maximum performance of Server is ensured, but in this case also the overload of computer is possible. This minimum* ***Protocol Timer Tick*** *value can be increased to some higher value (e.g. to 50 milliseconds) to achieve Server performance and computer load optimum correlation.*

NetDDE being used
Select this option if you are networking using NetDDE.

Configuration File Directory
The first field is used to specify the path (disk drive and directory) in which ABBT200 will save its current configuration file. ABBT200 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* ***ABBT200.CFG.***
Note: *There is no limit to the number of configuration files created, although each must be in a separate directory. When using the ABBT200 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 ABBT200 Server to start as a Windows NT service.

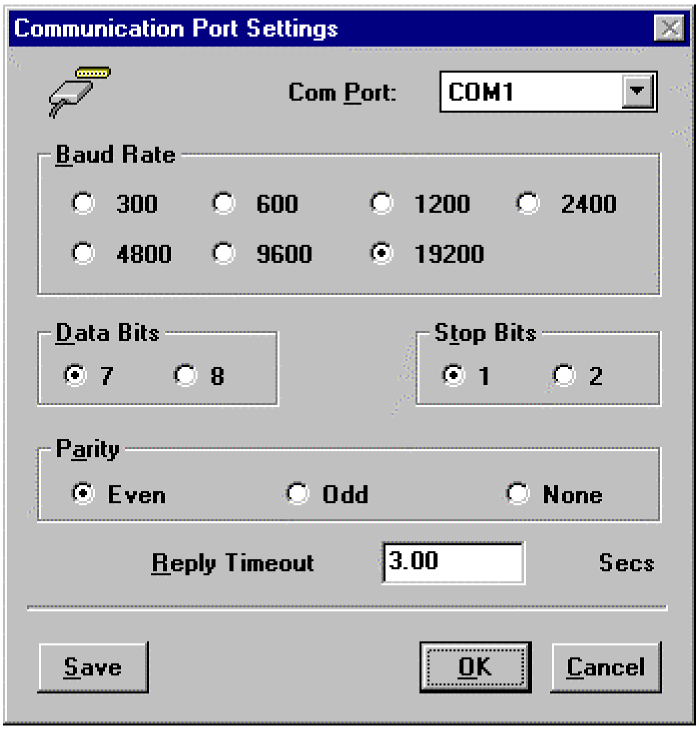
Windows NT offers the capability of running applications even when a user is not logged on to the system. This is valuable when systems must operate in an unattended mode. Enabling this option and rebooting the system will cause the Server to run as a Windows NT service. However, to view configuration 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.

Note: *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* ***ABBT200_IOServer*** *is selected). If* ***Allow Service to Interact with Desktop*** *is not selected then ABBT200 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**.

Com Port Settings Command

This command is used to configure the communication port that will be used to communicate with ABB T200 PLCs. Invoke the */Configure/Com Port Settings...* command. The "Communication Port Settings" dialog box will appear.



Communication Port Settings

Com Port: COM1

Baud Rate

○ 300 ○ 600 ○ 1200 ○ 2400

○ 4800 ○ 9600 ● 19200

Data Bits

● 7 ○ 8

Stop Bits

● 1 ○ 2

Parity

● Even ○ Odd ○ None

Reply Timeout 3.00 Secs

Select the **Com Port** and examine the characteristics of the selected Port. To select a necessary Com Port, click on the combo box button and make your choice from the list box.

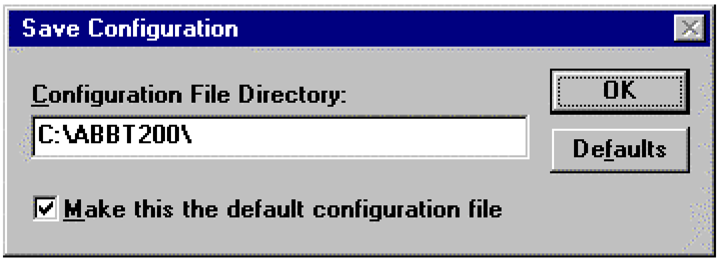
The **Reply Timeout** field is used to enter the amount of time (in seconds) the PLC using the selected communication port will be given to reply to commands from the Server. ***Note.*** *The default value of 3 seconds should be sufficient for most configurations.*

The following are default and recommended communication parameters: *Baud Rate - 19200, 7 Data Bits, 1 Stop Bit, Even Parity*. If necessary, these parameters can be changed. The communication parameters entered here must match with settings used on ABB T200 PLC side (on CPU Programming Port or on RS-232C port on 07 KP 60).

Once communication parameters and Reply Timeout are entered, select **Save** to save the configuration for the communication port. Once all entries have been made, click on **OK.**

Saving ABBT200 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:


Save Configuration

Configuration File Directory:

C:\ABBT200\

☑ Make this the default configuration file

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 file"** option. Doing so will allow the ABBT200 Server to find the configuration file automatically each time it is started.

Configuration File Location

When the ABBT200 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 and enter the following:

ABBT200 /d:c:\directoryname

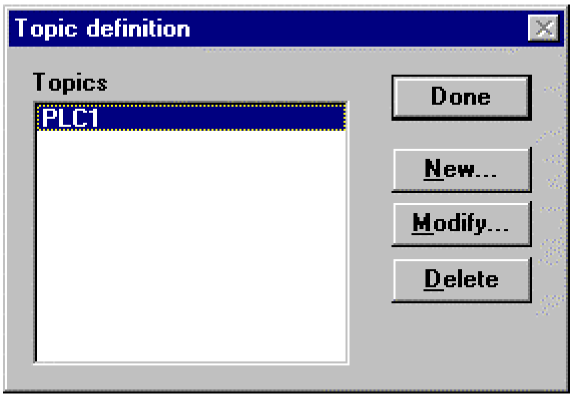
Note: *There is no limit to the number of configuration files that may be created, although each must be in a separate directory.*

Topic Definition Command

The user provides each PLC with an arbitrary name that is used as the Topic Name for all references to the PLC.

The following steps are taken to define the Topic attached to the ABB T200 PLC:

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

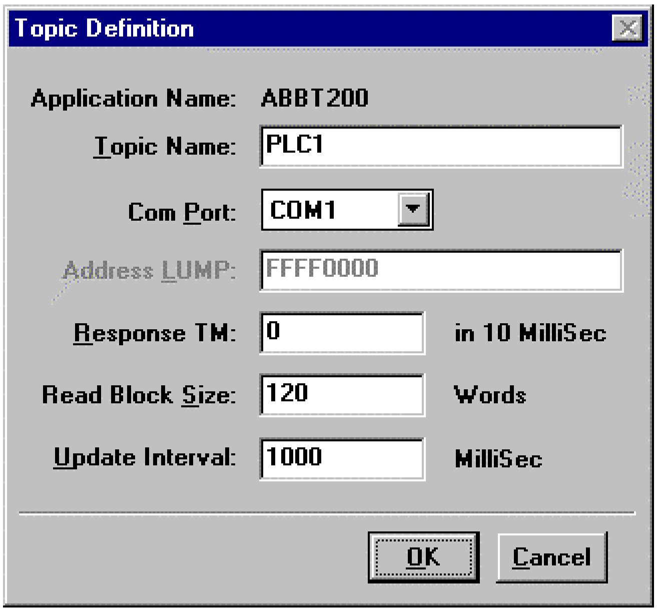


Topic definition

Topics

PLC1

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



Topic Definition

Application Name: ABBT200

Topic Name: PLC1

Com Port: COM1

Address LUMP: FFFF0000

Response TM: 0 in 10 MilliSec

Read Block Size: 120 Words

Update Interval: 1000 MilliSec

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 ABBT200 Server with InTouch*** *section.*

4. Click on the **Com Port** button to associate a topic with the communication port.
   Note: *Additional topics may be associated with the same communication port later.*

5. Enter the value in the **Response TM** field. This value (in units of 10 milliseconds) determines the response waiting time in transmission of control characters (ACK, EOT or NAK) from PLC to the Server. This value can be from 0 to 15 (from 0 to 150 milliseconds). This value should be adjusted depending on real application needs. ***Note:*** *The default* ***Response TM*** *is 0 (no waiting time).*

6. Enter the value in the **Read Block Size** field. This value determines the maximum number of consecutive words, which can be read from PLC in one read command (communication protocol 40H task). The greater value is configured the faster data update rate can be achieved. Therefore reduce the default **Read Block Size** value (120 words) only if PLC has timing problems to process great amount of data in a short time period.

7. Set the **Update Interval** field to indicate the frequency the items/points on this topic will be read (polled). This is requested update rate **-** at this frequency the values of all this topic active items must be updated. In real conditions (when large amount of data is requested from PLC) the real update rate can be longer **-** it will be automatically adjusted by the Server for maximum possible performance. ***Note:*** *The default* ***Update Interval*** *is 1000 milliseconds.*

There is also the “Address LUMP” field specified in this dialog box. This field is reserved for future use for the possibility to enter the LUMP (network address). In the current version of ABBT200 Server the default unchangeable value ‘FFFF0000” (direct communication) is used.

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

Select **Done** in the "Topic definition" dialog box when configuration for all Topics has been performed.

Item Names

Within the ABBT200 Server, item/point naming depends on ABB T200 PLC operand naming conventions. The Server supports the fixed set of item names, each of them generally may be described as:

Oa.bc,xxd

where **O -** operand identifier, e.g. **E** or **MW**;
a. - basic subrack number, used only for inputs and outputs; can be 0 or number of remote I/O line;
bc - group number or number of expansion subrack or number of substation and slot number for inputs and outputs;
, - separator;
xx - channel number;
d - optionally used suffix, can be the following:
:b - the address of bit in word or double word operands, for example, **MW5,12:07** is the bit 7 in flag word MW5,12; for word operands possible bit addresses are 00…15, for double word operands possible bit addresses are 00…31; if **:b** addressing is used then item type is Discrete (binary) Read Only;
u – indicates that word or double word value is interpreted as unsigned integer (value range 0...65535 for word operands and 0…21474836447 for double word operands); corresponding item without suffix **u** is interpreted as signed integer (value limits -32767...32767 for word operands and -21474836447…21474836447 for double word operands).

Notes:

*1. Both uppercase and lowercase characters can be used in item names. For example, both **MW5,12u** and **mw5,12U** are valid item names.*

*2. Leading zeroes can be excluded when addressing operands and bits. For example, both **MW5,12:07** and **MW005,12:7** are valid item names.*

*3. It is highly recommended that **consecutive addressing** (for example, MW1,1, MW1,2, MW1,3, etc.) is used for items accessed via the ABBT200 Server. This will greatly increase the performance of ABBT200 Server.*

*4. **Do not operate with items representing physically non-existing operands (e.g. inputs or outputs) in the PLC!** It will lead to communication failures, because the Server recognizes item name error only if item name goes out of range shown in the tables of supported item names (see below).*

The following tables list the supported item names for ABB T200 PLCs:

Flag operands

Operand type	Operand identifier in item name	Operand identifier in PLC program	Item type	Address range	Value Range (*)
Bit and special flag area	M	M	Discrete	000,00... 127,15	0, 1
Bit/word flag area(**)	MM	M’	Discrete	000,00... 383,31	0,1
Word flag area(***)	MW	MW	Integer	000,00... 4127,15	-32767... 32767 (0...32767, 32769…65535)
Double word flag area(***)	MD	MD	Integer	000,00... 3135,14	-2147483647… 2147483647 (0…2147483647)
Word/bit flag area(**)	MMW	MW’	Integer	000,00... 383,01	-32767... 32767 (0...32767, 32769…65535)
Double word/bit flag area(**)	MMD	MD’	Integer	000,00... 383,00	-2147483647… 2147483647 (0…2147483647)
Step chains (Read Only)	S	S	Discrete	000,00... 255,15	0,1

(*) Value ranges -32767... 32767 and -2147483647…2147483647 - for signed integers (item/point without suffix **u**);
value ranges 0...32767,32769…65535 and 0…2147483647 - for unsigned integers (item/point with suffix **u**).

(**) This is a common flag area, which can be addressed both in terms of bits, words and double words:

Double word:	MMDxxx,00					
Words:	MMWxxx,00			MMWxxx,01		
Bits:	MMxxx,00	…	MMxxx,15	MMxxx,16	…	MMxxx,31

(***) This is a common flag area, which can be addressed both in terms of words and double words:

Double word:	MDxxx,00	
Words:	MWxxx,00	MWxxx,01

Operands for inputs and outputs

The location addressing of inputs and outputs (I/Os) is the following:

Oa.bc,xx

where **O** - operand identifier (E, A, EW, AW, ED, AD);
a. - basic subrack: 0 or number of remote I/O line (1…4);
b - number of expansion subrack (0…5) or number of substation (0…9);
c - slot number (0…9);
, - separator;
xx - channel number: 00…31 for binary I/Os, 0…7 for word I/Os and 0…6 for double word I/Os.

Operand type	Operand identifier in item name	Operand identifier in PLC program	Item type	Address range	Value Range (*)
Binary inputs	E	E	Discrete	0.00,00... 4.99,31	0, 1
Binary outputs	A	A	Discrete	0.00,00... 4.99,31	0, 1
Word inputs	EW	EW	Integer	0.00,00... 4.99,07	-32767... 32767 (0...32767, 32769…65535)
Double word inputs	ED	ED	Integer	0.00,00... 4.99,06	-2147483647… 2147483647 (0…2147483647)
Word outputs	AW	AW	Integer	0.00,00... 4.99,07	-32767... 32767 (0...32767, 32769…65535)
Double word outputs	AD	AD	Integer	0.00,00... 4.99,06	-2147483647… 2147483647 (0…2147483647)

(*) Value ranges -32767... 32767 and -2147483647…2147483647 - for signed integers (item/point without suffix **u**);
value ranges 0...32767,32769…65535 and 0…2147483647 - for unsigned integers (item/point with suffix **u**).

Both inputs and outputs areas can be can be addressed both in terms of bits, words and double words:

Double word:	EDx.xx,00					
Words:	EWx.xx,00			EWx.xx,01		
Bits:	Ex.xx,00	…	Ex.xx,15	Ex.xx,16	…	Ex.xx,31

Operands for timers and counters

Operand type	Operand identifier in item name	Operand identifier in PLC program	Item type	Address range	Value Range
Timers	T	T	Integer	00,00... 15,15	0…32767
16-bit counters	Z	Z	Integer	00,00... 15,15	0…65535

All timer and counter items are Read Only.

Operands for coupler areas

Operand type	Operand identifier in item name	Operand identifier in PLC program	Item type	Address range	Value Range (*)
Inputs in the coupler area	EE	E’	Discrete	0000,00... 2047,15	0,1
Outputs in the coupler area	AA	A’	Discrete	0000,00... 2047,15	0,1
Word inputs, coupler area	EEW	EW’	Integer	0000,00... 2047,15	-32767... 32767 (0...32767, 32769…65535)
Word outputs, coupler area	AAW	AW’	Integer	0000,00... 2047,15	-32767... 32767 (0...32767, 32769…65535)

(*) Value ranges -32767... 32767 and -2147483647…2147483647 - for signed integers (item/point without suffix **u**);
value ranges 0...32767,32769…65535 and 0…2147483647 - for unsigned integers (item/point with suffix **u**).

Both inputs and outputs in the coupler areas can be can be addressed both in terms of bits and words:

Word:	EEWxxxx,00		
Bits:	EExxxx,00	…	EExxxx,15

Item/Point Naming Examples

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

M1,15 - bit in **Bit flag area**, group **1**, channel **15**, item type **Discrete**, R/W

MM6,2 - bit in **Bit/word flag area**, group **6**, channel **2**, item type **Discrete**, R/W

MW10,0 - word in **Word flag area**, group **10**, channel **0**, item type **Integer**, R/W

MW8,0:12 - **12th** bit in **Word flag area** word, group **8**, channel **0**, item type **Discrete**, R/O

MD13,14 - double word in **Double word flag area**, group **13**, channel **14**, item type **Integer**, R/W

MMW0,7u - word in **Word/bit flag area**, group **10**, channel **7**, item type **Integer** (unsigned), R/W

MMD003,3 - double word in **Double word/bit flag area**, group **3**, channel **3**, item type **Integer**, R/W

S0,08 - bit in **Step chains area**, group **0**, channel **8**, item type **Discrete**, R/O

E2.48,31 - the **binary input**, **2nd** remote I/O line, **4th** substation, **8th** slot, channel **31**, item type **Discrete**, R/O

AW0.03,00 - the **word output**, basic subrack, **3rd** slot, channel **3** (16-bit channel), item type **Integer**, R/W

AD2.15,1 - the **double word output**, **2nd** remote I/O line, **1st** substation, **5th** slot, channel **1** (32-bit channel), item type **Integer**, R/W

t0,5 - **timer**, group **0**, channel **5**, item type **Integer**, R/O

Ee0005,12 - bit in **binary inputs coupler area**, group **5**, channel **12**, item type **Discrete**, R/W

aaw21,04 - word in **word outputs, coupler area**, group **21**, channel **4**, item type **Integer**, R/W

Monitoring Communication with a PLC

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:

=ABBT200|topic!STATUS

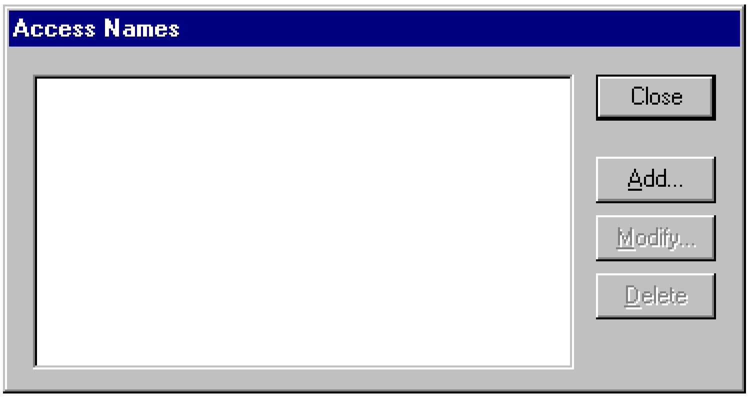
Using the ABBT200 Server with InTouch

To access items/points on ABB T200 PLC from **InTouch**, the Access Names and Tag names should be defined in **WindowMaker**.

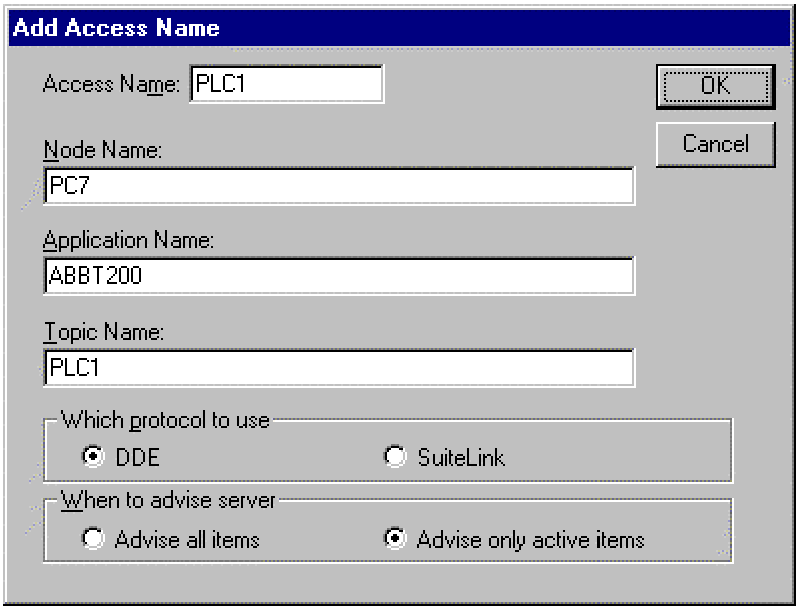
Defining the Access Names

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:



Add Access Name

Access Name: PLC1

Node Name: PC7

Application Name: ABBT200

Topic Name: PLC1

Which protocol to use

● DDE ○ SuiteLink

When to advise server

○ Advise all items ● Advise only active items

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 ABBT200 Server the ABBT200 is used. Do not enter the .exe extension portion of the program name.

Topic Name

Enter the name defined for the topic in the ABBT200 Server to identify the topic the ABBT200 Server will be accessing.
The Topic Name is an application-specific sub-group of data elements. In the case of data coming from ABBT200 Server program, the topic name is the exact same name configured for the topic in the ABBT200 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 ABBT200 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 <u>is not recommended</u>.

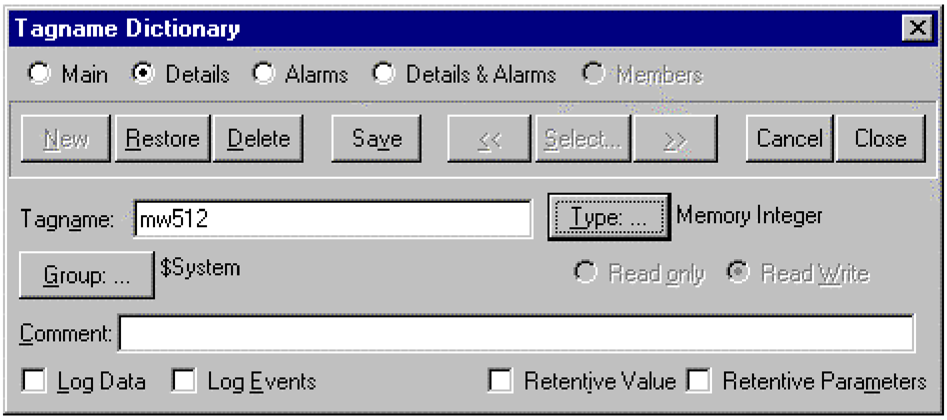
Select **Advise only active items** if you want the Server program to poll only points in visible windows and points that are alarmed, logged, trended or used in any script.

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:



Tagname Dictionary

○ Main ● Details ○ Alarms ○ Details & Alarms ○ Members

Tagname: mw512 Memory Integer

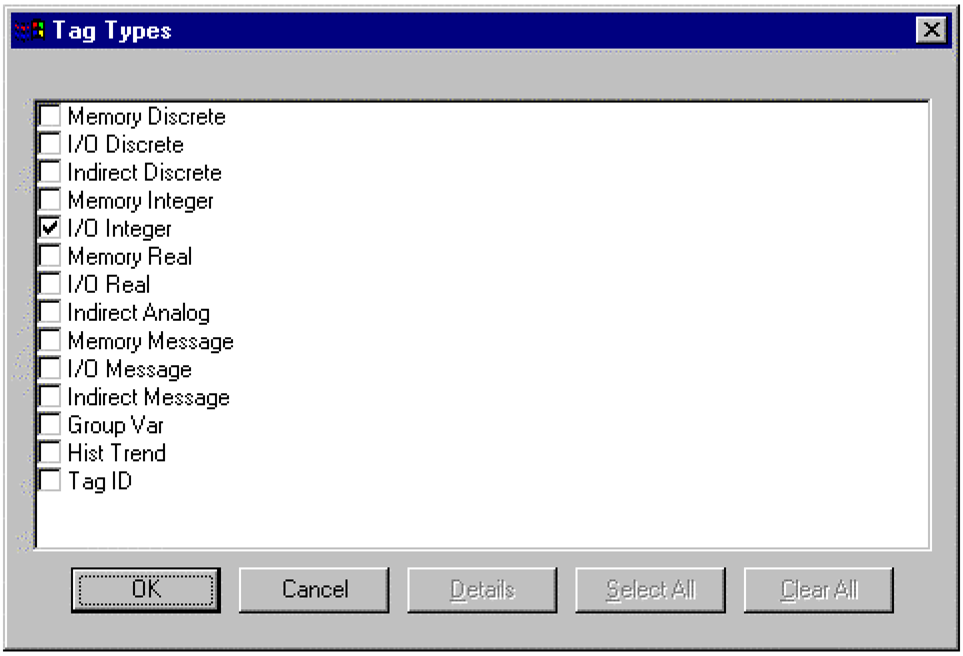
$System ○ Read only ● Read Write

Comment:

□ Log Data □ Log Events □ Retentive Value □ Retentive Parameters

Click on **New** and enter the **Tag Name**. (The tag name defined here is the name **InTouch** will use. The ABBT200 Server does not see this name.)

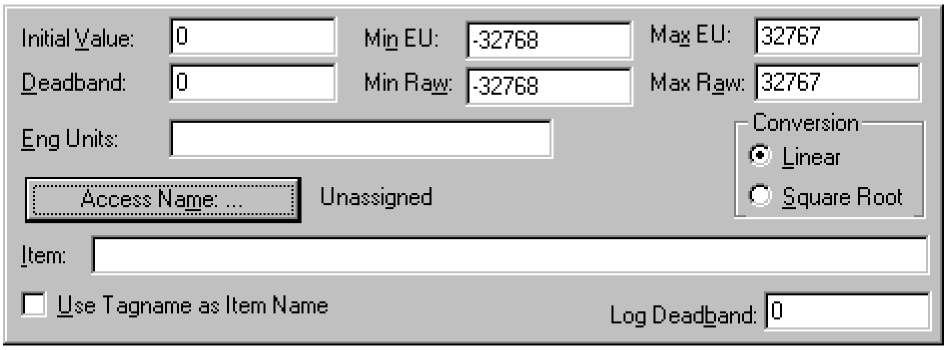
Select the tag type by clicking on the **<u>T</u>ype:...** button. The "Tag Types" dialog box will appear:



☐ Memory Discrete
☐ I/O Discrete
☐ Indirect Discrete
☐ Memory Integer
☑ I/O Integer
☐ Memory Real
☐ I/O Real
☐ Indirect Analog
☐ Memory Message
☐ I/O Message
☐ Indirect Message
☐ Group Var
☐ Hist Trend
☐ Tag ID

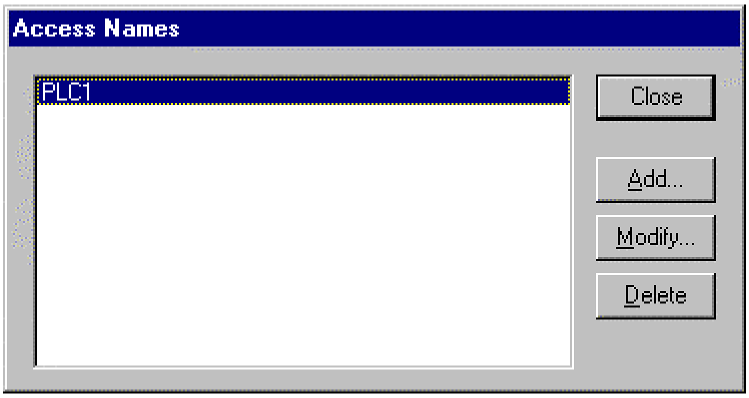
To access ABBT200 items, the type must be **I/O Discrete** or **I/O Integer.** Select the Tag type.

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



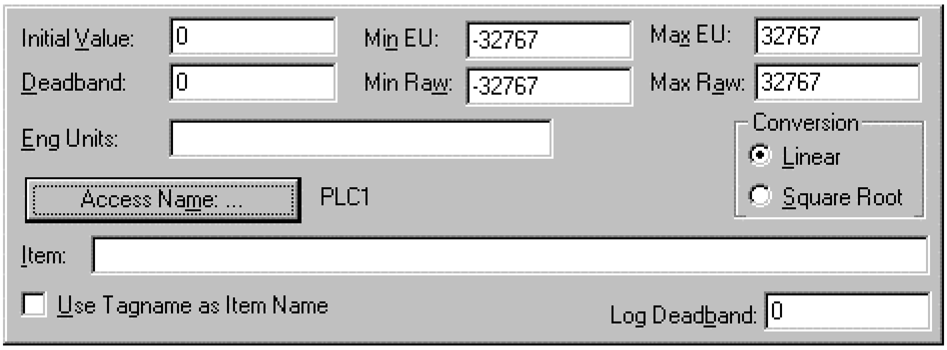
Initial Value: 0
Min EU: -32768
Max EU: 32767
Deadband: 0
Min Raw: -32768
Max Raw: 32767
Eng Units:
Conversion
● Linear
○ Square Root
Unassigned
Item:
☐ Use Tagname as Item Name
Log Deadband: 0

Select the Access Name for ABBT200 Server by clicking on the **Access Name:...** button. The "Access Names" dialog box will appear:



PLC1

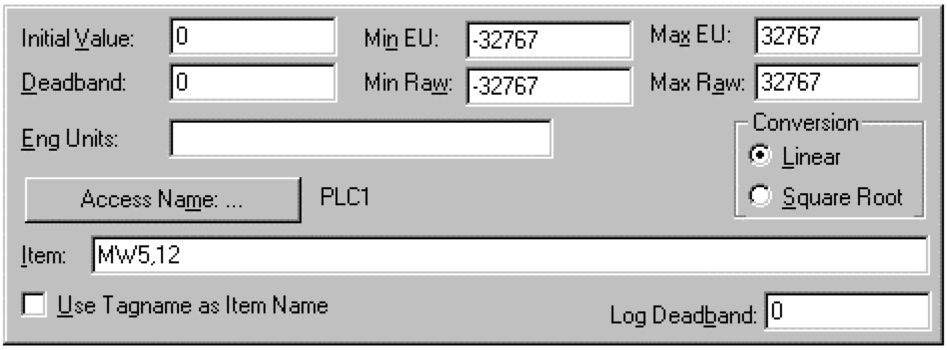
Select the appropriate Access Name and click on **Close**. (If the Access Name has not been defined as previously described, click on **Add** and define the Access Name now.) The "Details" dialog box will appear displaying the selected Access Name:



Initial Value: 0
Min EU: -32767
Max EU: 32767
Deadband: 0
Min Raw: -32767
Max Raw: 32767
Eng Units:
Conversion: ● Linear ○ Square Root
Access Name: ... PLC1
Item:
□ Use Tagname as Item Name
Log Deadband: 0

For integers fill in the **Min EU**, **Max EU**, **Min Raw** and **Max Raw** fields. These fields control the range of values, which 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 ABBT200 item name to be associated with this tagname in the **Item:** field in the "Details" box:



Initial Value: 0
Min EU: -32767
Max EU: 32767
Deadband: 0
Min Raw: -32767
Max Raw: 32767
Eng Units:
Conversion: ● Linear ○ Square Root
Access Name: ... PLC1
Item: MW5,12
□ Use Tagname as Item Name
Log Deadband: 0

(Refer to the **Item Names** 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 Names*** *section.*

Once all entries have been made, click on the **Save** button (in the top dialog box) to accept the new tagname. 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**, which are used to monitor the status of communication 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 **PLC1** is the Topic Name configured for ABBT200 Server):

=view|DDEStatus! PLC1
or
=view|IOStatus! PLC1

Notes on Using Microsoft Excel

Data from ABBT200 topics (nodes) may be accessed from Excel spreadsheets. To do so, enter a formula like the following into a cell on the spreadsheet.

=ABBT200|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 Names** 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:

=ABBT200|'PLC1'!'MW5,12'
=ABBT200|'ABBT200'!'e0.03,00'
=ABBT200|'plc'!'m010,5'

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

Writing Values to ABBT200 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("ABBT200","topicname")
=POKE(channel,"itemname", Data_Reference)
=TERMINATE (channel)
=RETURN()

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

channel=INITIATE("ABBT200 ","topicname")
Opens a channel to a specific topic name (defined in the Server) in an application with name ABBT200 (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 T200 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 specified like described in the ***Item Names*** 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 **MW5,12**:

```
PokeMacro -Ctrl a
=INITIATE("ABBT200","PLC1")
=POKE(A2,"MW5,12",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 ABBT200 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 or C:\Win95). 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 ABBT200 Server:

```
 [ABBT200]
WinIconic=0
WinFullScreen=0
WinTop=112
WinLeft=0
WinWidth=200
WinHeight=168
```

There are following additional WIN.INI entries available for ABBT200 Server:

SlowPollRetries and SlowPollInterval

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 5 consecutive unsuccessful retries:

```
SlowPollRetries=5
SlowPollInterval=120
```

Entering the slow poll mode is reported to WWLogger by following string:

"Set slow poll mode - poll after each <SlowPollInterval * 1000> msecs. Stop error logging to Topic <TOPICNAME>."

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

"Port <PORTNAME>; return to normal communication to Topic <TOPICNAME>."

ConsecutiveWriteErrors

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 default **ConsecutiveWriteErrors** value is 5 (used if WIN.INI file does not contain this entry).

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 window):

Suspend Protocol / Resume Protocol - these choices permit you to turn protocol processing on and off, what means that you can suspend access to the PLC.
Show Send - if checked then all outgoing user data is displayed.
Show Receive - if checked then all incoming user data is displayed.
Show Errors - if checked then all information about errors is displayed.
Verbose - if checked then additional debugging information is displayed. This option is useful for getting additional information about error situations:
Dump - displays all information about ports, active topics and data items.
DumpScreen - if checked information about active topics and messages is displayed in ABBT200 main window.

All debugging information (except **DumpScreen** option) is displayed via the Wonderware Logger, which must be active for these commands to work.
Warning: if you check **Show Send** and/or **Show Receive** debug output grows very fast.

KLINKMANN AUTOMATION
ABBT200 Communication Server
Revision History

Mar 99	Rev 1.0	First Release
May 99	Rev 1.1	Minor changes in the “Installing the ABBT200 Server” section
Mar 2002	Rev 1.2	Installation from CD information added.