ABBCS31
ARCNET
DDE Server

for Microsoft Windows
and InTouch Applications

User Manual
Ver 2.x Rev 2.5
DR 010 15
DR 010 16

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 CS31
ARCNET DDE Server

The ABB CS31 ARCNET DDE Server (ABB_ARCN) is a Microsoft Windows program that acts as a DDE (Dynamic Data Exchange) *Server* and allows other Windows programs to access to data from ABB CS31 07KT93 Controllers. The ABB CS31 ARCNET DDE *Server* requires an ARCNET adapter card (e.g. ARCNET-PC130E Network Controller Board based on the SMC ARCNET chip or SOHARD SH ARC PCI ARCNET card) to communicate with ABB CS31 07KT93 Controllers. Through this high speed communication network the *Server* can access up to 8 ABB CS31 07KT93 Controllers. Program, running on the ABB CS31 07KT93 Controller, must be developed so, that it ensures communication with the host through ARCNET bus.

There are two different ABB_ARCN Server versions described in this manual:
- 32-bit version for ISA ARCNET cards (ordering number DR 010 15), can be used on Microsoft Windows NT; this version of Server uses WinRT kernel driver to access ISA ARCNET adapter card.
- 32-bit version for SOHARD SH ARC PCI ARCNET card (ordering number DR 010 16), can be used on Microsoft Windows NT and 2000; this version of Server uses corresponding SOHARD ARCNET Raw Drivers (included in the installation package).

Also two different Server Distribution Disks are supplied. All further information in this manual is same both for 16-bit and 32-bit versions with the exception of several chapters, where NT, WinRT and PCI ARCNET card specific items are described.

The *Server* is primarily intended for use with **Wonderware InTouch** (version 3.01 and later), but it may be used by any Microsoft Windows (NT or 2000) program that is capable of acting as a DDE, FastDDE or SuiteLink *Client.*

What is DDE?

DDE is a complete communication protocol designed 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 **InTouch** and Excel can simultaneously be both a *client* and *server.*
To obtain data from another application the *client* program opens a channel to the *server* application by specifying three things: the *server* ***application name***, the ***topic name*** and the specific ***item name***. For example, in the case of Excel, the application name is "Excel", the topic name is the name of the specific spreadsheet that contains the data and the item name is the specific cell on the spreadsheet. With **InTouch** the application name is "View", the topic name is the word "Tag name" when reading/writing to an **InTouch** tag name and the item name is a specific tag name in the **InTouch** Data Dictionary.
When a *client* application sets up a link to another DDE program, it requests the *server* application to *advise* the *client* whenever a specific item's value changes. These data links will remain active until either the *client* or *server* program terminates the link or the conversation. They are a very efficient means of exchanging data because once the link has been established no communication occurs until the specified data value changes.

InTouch uses DDE to communicate with I/O device drivers and other DDE application programs.

Accessing a Remote DDE Item from ABB_ARCN

The DDE protocol identifies an element of data by using a three-part address, including: ***Application***, ***Topic*** and ***Item.***

Application refers to the name of the Windows program (server) that knows how to access the data element. In the case of data coming from or going to ABB CS31 07KT93 hardware, the application portion of the DDE address is **ABB_ARCN**.

Topic is an application-specific sub-group of data elements. The ABB_ARCN Server considers each ABB CS31 07KT93 Controller to be a separate topic. The user creates meaningful name for each Controller and uses this name as the topic name for DDE references.
Note: *In some cases, the term "node" is used interchangeably with the term "topic".*

Item indicates a specific data element within the specified topic. With the ABB_ARCN Server, an item is an individual point in the ABB CS31 07KT93 Controller memory. (The item/point names are fixed by the ABB_ARCN Server as described in the ***Item (Point) Naming*** section.)
Note: *In some cases, the term "point" is used interchangeably with the term "item".*

Installing the ABB_ARCN DDE Server

Installing the Server

The ABB_ARCN Server installation package can be supplied:

1. As a self-extracting archive (11015xxx.EXE for ISA version or 11016xxx.EXE for PCI 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 ABB_ARCN Server from the self-extracting archive, run the 11015xxx.EXE or 11016xxx.EXE and proceed as directed by the Server Setup program.

To **install** the ABB_ARCN Server from CD or distribution disks, on MS Windows (NT or 2000):

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

ABB_ARCN DDE Server for **Windows NT** must be installed from Administrator account or from other privileged account with Administrator rights.

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

ABB_ARCN.EXE	The ABB_ARCN Server Program. This is a Microsoft Windows 32-bit application program.
ABB_ARCN.HLP	The ABB_ARCN Server Help file.
ABB_ARCN.CFG	An example configuration file.
LICENSE.TXT	Klinkmann Automation software license file.
WNT_RAW_DRIVER_V0517.zip	SoHard ARCNET Raw Driver v5.17 installation package for MS Windows NT, needed only for PCI version.
W2K_RAW_DRIVER_V0517.zip	SoHard ARCNET Raw Driver v5.17 installation package for MS Windows 2000, needed only for PCI version.
manual_SH_ARC_PCI_e.pdf	SoHard ARCNET PCI Card User Manual.
PROJEKT.ZIP	Packed file containing ABB CS31 07KT93 example program with no response to write commands.
PROJEKT2.ZIP	Packed file containing ABB CS31 07KT93 example program processing response to write commands.

After the Server is installed, the following additional steps are necessary to enable Server interface with SOHARD SH ARC PCI ARCNET card:

1) Add the following two rows to WIN.INI file (usually located in Windows main directory, e.g. C:\WINNT):

[ABB_ARCN]
PCICard=1

2) Install tha SOHARD SH ARC PCI ARCNET and driver:
- Unpack the driver installation (**WNT_RAW_DRIVER_V0517.zip** or **W2K_RAW_DRIVER_V0517.zip**) to some temporary location.
- Switch off PC and insert the ARCNET card in any free PCI slot.
- Turn on the PC.
- When the "Found New Hardware Wizard" is started, check "Search for... (recommended)".
- Leave the checked state "Specify a location" only.
- Enter a full path where installation of SH RAW driver is located.
- In the next dialog box the Wizard must found the device "Network Controller".
- At the end, click on "Finish".

For more information how to install the SOHARD SH ARC PCI ARCNET and driver refer to readme.html included in the **WNT_RAW_DRIVER_V0517.zip** or **W2K_RAW_DRIVER_V0517.zip.**

Note!
Always check if there is newer driver version supplied together with SOHARD SH ARC PCI ARCNET card – in this case use the newest version supplied with card.

3) Start the ABB_ARCN Server. Ignore (press OK) message displayed regarding WINRT settings. Check if Board I/O Address and Board Memory Segment entries are disabled (grayed) in the "ARCNET Adapter Card Settings" dialog box (see *Board Configuration Command* section below). Do not reboot if message is displayed to activate some changes.

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

Before the ABB_ARCN DDE Server is used, all connected ABB CS31 07KT93 Controllers must be initialized for the communication by loading the PLC application program to each Controller. The example programs for communication with ABB_ARCN DDE Server are contained in the \PROJEKT and \PROJEKT2 subdirectories on the ABB_ARCN DDE Server distribution disk. (see chapters *Installing the ABB CS31 Example Program* and *ABB CS31 07KT93 Program Description*).

Notes:
1. *The ABB_ARCN 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 ABB_ARCN 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 ABB_ARCN Server 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 ABB_ARCN Server and these infrastructure files must be install prior to executing the ABB_ARCN Server.*
3. *The HASP key is needed for full time running of ABB_ARCN Server. The HASP Driver setup is performed during the Server setup. Without HASP Driver installed and HASP key plugged into PC parallel port, the ABB_ARCN Server will run only 1 hour (with all features enabled).*
4. *The ABB_ARCN Server PCI version uninstallation will remove only the Server; to remove also the SOHARD ARCNET Raw Drivers, follow instructions from corresponding installation packages (**WNT_RAW_DRIVER_V0517.zip** or **W2K_RAW_DRIVER_V0517.zip**).*

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 CD or 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 CD or distribution disk, on MS Windows (NT or 2000):

1. Insert the CD with Klinkmann Software into CD drive or insert the I/O Server Infrastructure disk 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 disk (floppy).
4. If installing from CD: select "Protocol Servers (DDE, SuiteLink, OPC)", find "I/O Server Infrastructure" and click on "Setup…".
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.*

Installing the ABB CS31 Example Program

The ABB CS31 example programs for communication with ABB_ARCN DDE Server are not installed by Installation program. Each ABB CS31 example program is installed by copying all 15 files contained in the \PROJEKT (or \PROJEKT2) subdirectory on the ABB_ARCN DDE Server distribution disk to a subdirectory on your hard disk, where the Project Files for your ABB Procontic CS31 Programming System are located (defaults: \ABB-SPS\CS31\PROJEKT or \ABB-SPS\CS31\PROJEKT2). Before installing ABB CS31 example program you must install the ABB Procontic Programming System according to the *Operating Manual.*

The following steps are taken to install the ABB CS31 example program:

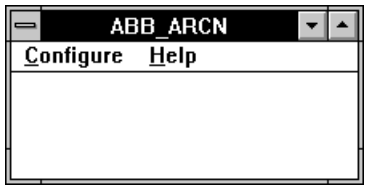
1. Insert the ABB_ARCN DDE Server disk into a floppy drive.
2. At the DOS prompt type:
   COPY A:\PROJEKT*.* <path to project file directory>
   and press Enter (default path: C:\ABB-SPS\CS31\PROJEKT).

Before the ABB_ARCN DDE Server is started, all connected ABB CS31 07KT93 Controllers must be initialized for the communication by loading the PLC application program to each Controller. (see chapter *ABB CS31 07KT93 Program Description*).

Configuring the ABB_ARCN Server

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

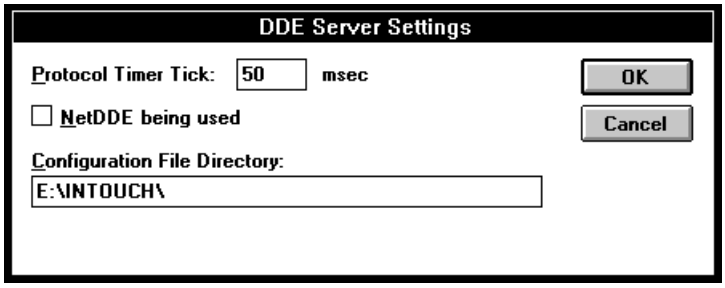
To perform the required configurations, start up the ABB_ARCN program. If the Server starts up as an icon, double-click on the icon to open the server's window. The following will appear:



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

DDE 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:

Configuration File Directory
The first field is used to specify the path (disk drive and directory) in which ABB_ARCN will save its current configuration file. ABB_ARCN 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* ***ABB_ARCN.CFG****.*
Note: *There is no limit to the number of configuration files created, although each must be in a separate directory. When using the ABB_ARCN Server with* ***InTouch****, it is good practice to place the configuration file in the application directory.*

Protocol Timer Tick
This field is used to change the frequency at which the Server checks for data to process. This should be approximately 2 to 4 times the fastest rate desired to update data from the ABB CS31 07KT93 hardware.
PLC program's cycle time should also be taken into account when configure this parameter. For example, PLC program's cycle time is 100 milliseconds. If data from PLC is accessed by one PC then **Protocol Timer Tick** should be 100 milliseconds or a bit greater. If two PCs access data from PLC simultaneously then **Protocol Timer Tick** should be 200 milliseconds or a bit greater on each PC.

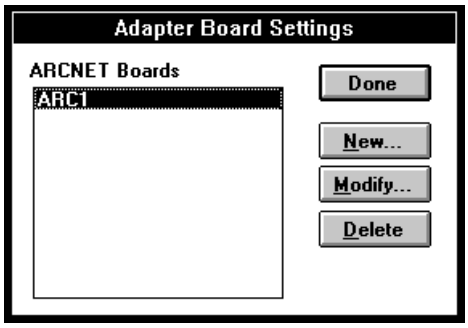
NetDDE being used
Select this option if you are networking using NetDDE.

When all entries have been made, click on **OK**. To reset the default values, click on **Defaults**.

Board Configuration Command

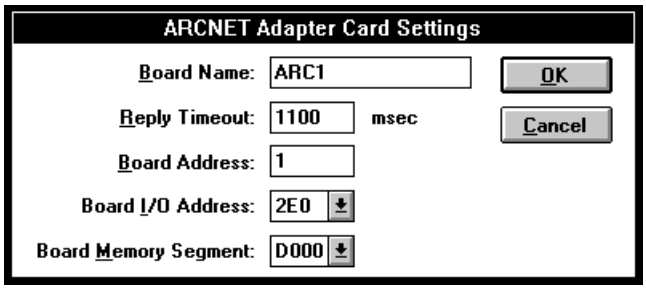
ARCNET Network Board configuration for **Windows NT** version of ABB_ARCN DDE Server must be done from Administrator account or from other privileged account with Administrator rights.

To configure the ARCNET Network Board, invoke the **Board Configuration...** command. The "Adapter Board Settings" dialog box will appear:



Select the ARCNET Network Board and click on **Modify** to examine the characteristics of the selected Board.

The "ARCNET Adapter Card Settings" dialog box will appear:



The following describes each dialog field in this dialog box:

Board Name
This field is used to enter the ARCNET Board name.

Board Address
This field is used to enter the Board address consistent with the Board Station address selected on the ARCNET adapter card.

Reply Timeout
This field is used to enter the amount of time (in milliseconds) all Controllers using the selected communication port will be given to reply to commands from the Server.
Note: *The default value of 1100 milliseconds should be sufficient for most configurations*.

Board I/O Address
This field is used to enter the I/O base address, in hexadecimal, of the ARCNET interface. The Board I/O Address must be consistent with the base address selected on the ARCNET adapter card. To select a necessary Board I/O Address click on the combo

box button and make your choice from list box. Care should be taken to select a base address which will not conflict with other devices in the system.
Note: *The default Board I/O Address is 2E0.*

Board Memory Segment
This field is used to enter the segment address, in hexadecimal, for the ARCNET chip buffers. The Board Memory Segment address must be consistent with the base memory (RAM) buffer address selected on the ARCNET adapter card. To select a necessary Board Memory Segment click on the combo box button and make your choice from list box.
Note: *The default Board Memory Segment address is D000.*

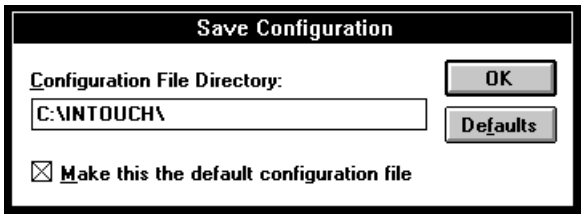
When all entries have been made, select **OK** to process the configuration for the ARCNET board.

Select **DONE** in the "Adapter Board Settings" dialog box when all ARCNET boards have been configured.

Notes:
*1. For **Windows NT PCI version** of ABB_ARCN DDE Server the **Board I/O Address** and **Board Memory Segment** entries are disabled (grayed).*
*2. For **Windows NT ISA version** of ABB_ARCN DDE Server to add new ARCNET boards, remove existing ones, or change "Board I/O Address" or "Board Memory Segment" for existing ARCNET boards configuration must be done from Administrator account or from other privileged account with Administrator rights.*
*3. For **Windows NT ISA version** of ABB_ARCN DDE Server if you added new ARCNET boards, removed existing ones, or changed "Board I/O Address" or "Board Memory Segment" for existing ARCNET boards to activate board settings WinRT kernel device must be restarted. If ABB_ARCN DDE Server fails to restart WinRT kernel device you will need to reboot the computer to activate board settings.*

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

Configuration File Location

When the ABB_ARCN Server starts up, it first attempts to locate its configuration file by, first checking the **WIN.INI** file for a path which 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:

ABB_ARCN /d:c:\directoryname

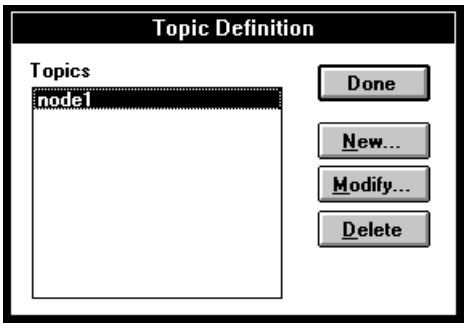
Note: *There is no limit to the number of configuration files that may be created, although each must be in a separate directory.*

Topic Definition Command

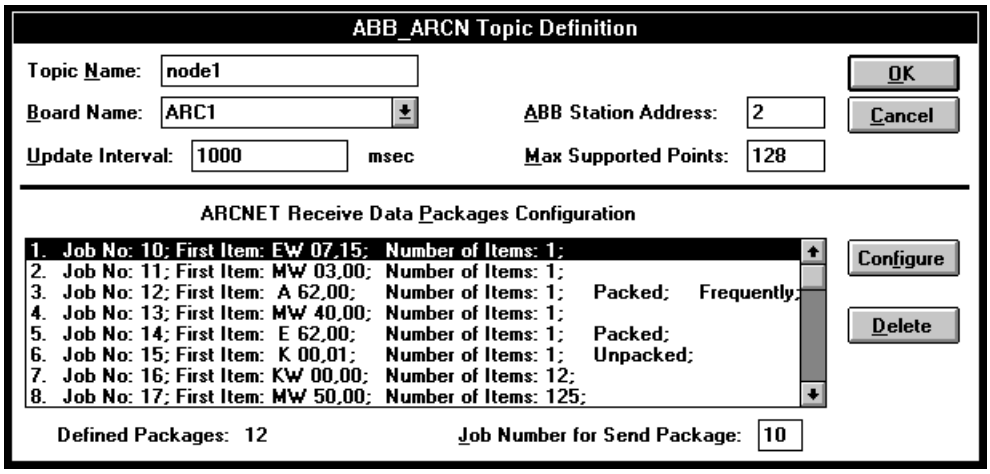
The user provides each Controller with an arbitrary name which is used as the DDE topic for all references to the Controller.

The following steps are taken to define the Topics (Controllers) attached to the network:

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 "ABB_ARCN DDE Topic Description" dialog box will appear:



3. Enter the **Topic Name** which corresponds to the DDE Topic Name. (The DDE Topic Name is entered in the "DDE Access Name Definition" dialog box described in the **Using the ABB_ARCN Server with InTouch** section.)

4. Enter the value in the **ABB Station Address** field. (Refer to the ABB CS31 Operational Manual for the correct address setting.)
   Note: *Each Controller must have an unique non-zero address.*

5. Set the **Max Supported Points** for the Controller. This field is used to enter the maximum number of DDE items supported by this topic.
   Note: *Default value is 128.*

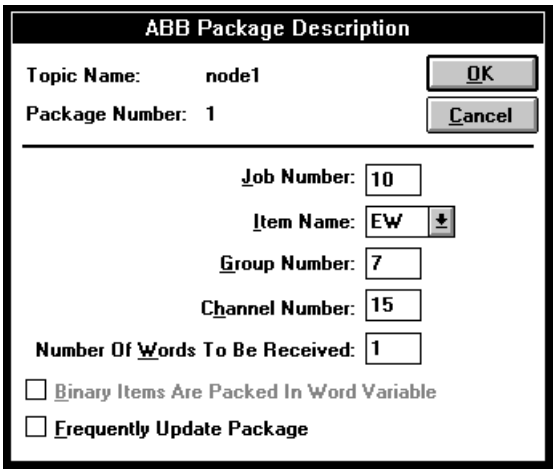
6. Click on the **Board Name** combo box to associate an ARCNET Network Controller Board with the topic. (Additional topics may be associated with this same ARCNET Board at a later time.)
   Note: *If this is the first time an ARCNET Board has been configured, the user will be prompted to save it to an existing directory.*

7. Set the **Update Interval** field to indicate the frequency the items/points on this topic will be read (polled).

8. Fill **ARCNET Receive Data Packages Configuration** list box; enter information about send and all receive data packages used in communication between the PLC and the ABB_ARCN Server. Data in this list box must correspond to ABB CS31 program. For example, your ABB CS31 program contains $ASEND block with Configuration Order Number (CON) 1 (see chapter *ABB CS31 07KT93 Program Description*) and the following parameters:

   - #JOB #12
   - EN0 M001,00 ;CON 1
   - #n0 #1
   - N0 KW01,01
   - #D0 #127
   - #J0 #10
   - #L0 #1
   - MW0 EW07,15
   - ST0 M00,00
   -
   - EN11 M001,11
   - #n11 #1
   - N11 KW01,01
   - #D11 #127
   - #J11 #21
   - #L11 #2
   - MW11 MW010,01
   - ST11 M00,11

To enter information about the $ASEND block with Configuration Order Number 1 choose the row number 1 in the list box (for any $ASEND block - row number in the list box must be equal to the $ASEND block Configuration Order Number) and press **Configure** button (or mouse left button double-click). The "ABB Package Description" dialog box will appear:



The following describes each dialog field in this dialog box:

Job Number
This field is used to enter the Job Number of the correspondent data package in the $ASEND CE of ABB CS31 program. For ABB CS31 this number must be from 0 to 31. Job Number must be unique.

Item Name
This field is used to enter the Item Name of the first received item. To select a necessary Item Name click on the combo box button and make your choice from list box. Choices - E, A, M, K, EW, AW, MW and KW (see chapter *Item (Point) Naming*).

Group Number
This field is used to enter the Group Number of the first received item (see chapter *Item (Point) Naming*).

Channel Number
This field is used to enter the Channel Number of the first received item (see chapter Item (Point) Naming).

Number Of Words To Be Received
This field is used to enter the Number Of Words To Be Received from the ABB CS31 in the corresponding data package. Value must be equal to the parameter #L0 of $ASEND block in the ABB CS31 program. For ABB CS31 valid number is from 1 to 125 (words). ***Note:*** *do not configure data package which exceeds the end of the corresponding item (PLC memory) area!*

Binary Item Are Packed In Words
Choose this option, if received binary items are packed in word variable(s) in the ABB CS31 program. This option is possible only for binary items - E, A, M, K (see chapter "Item (Point) Naming").

Frequently Update Package
Choose this option, if package must be received not only in its update period but also together with any received group of packages. This option is useful, for example, for error or communication status detection. ***Note:*** *It is recommended to use this option only if it is*

really necessary, because otherwise it can hardly interrupt normal communication performance!

ABB CS31 07KT93 Program Description

ABB CS31 07KT93 (PLC) ARCNET communication depends on the user developed PLC program. The PLC memory areas are accessible from the ABB_ARCN Server, if a PLC program contains corresponding connection elements (CE), which operate with the same areas. Therefore, ABB_ARCN Server user must develop his PLC application program or modify an existing program according to the following matters:

an application program must contain CE for ARCNET initialization;

an application program must contain CE for data package receive. Four consecutive words in the PLC memory must be reserved for received data storage. (Received 4 words contains instructions for the PLC program);

an application program must contain CE groups for indexed writing of binary and word variables;

an application program must contain 1...32 CE for data sending. Each data send CE can send the portion of data, which parameters (data type, start address and sent word number (1...125)) is set during the PLC program development and are unchangeable during program performance. Any portion of data you need to access, must be represented in one of the data send CEs;

if system is configured to process an answer to write commands then an application program must contain CE for sending response to write command;

if you need to access binary data, necessary number of words must be reserved for binary data packing and corresponding CE must be used to send the packed data to the server;

an application program must contain CE for the bit pattern unpacking to recognize, which data send CE must be executed.

After the PLC program startup it initializes ARCNET communication and waits for instructions from the server. If the change value command is received, the PLC program performs necessary actions but does not send any response to the PLC. If the data request command is received, the PLC program sends requested data to the server.

All these features illustrates a PLC example program. An example program is developed using ABB Procontic Programming System software (version 05 / 94). Program contains a link of CE (full of PLC program text see using ABB Procontic Programming System CE editor) for successful communication between PLC and ABB_ARCN DDE Server. The following CE are important :

CE for initialization of the ARCNET controller. The following parameters are set:

- interrupt after reception of a data package;
- only short package (256 bytes);
- data packages to all stations.

Example :

```
$AINIT
        #0/1        M255,15
        DONE        A62,00
        TO          MW003,00
        ERR         A62,01
        NODE        MW000,00
        STAT        MW000,01
        DIAG        MW000,02
        TOS         A62,02
        TOND        MW000,03
        TOJN        MW000,04
        LEV         MW000,05
        RECO        A62,03
```

Parameter TO - Timeout in ms - waiting time for sending data packages. This parameter value must be less than parameter **Update Interval** (see **Topic Definition** dialog box in chapter "**Topic Definition Command**") and parameter **Replay Timeout** (see **Board Configuration** dialog box in chapter "**Configure Board Command**").

Parameter TOS - timeout has occurred during send operation.

CE for data package receive. Only one data receive package is used to receive data from the ABB_ARCN Server. The received 4 words of package data contains instructions to ABB CS31 program.

The 1st received word identifies the data type the ABB_ARCN Server and the ABB CS31 program operates with at the present moment as well as operation type (read or write) to be performed by the PLC program.

The 2nd received word contains accessing variable index (see IDSm block description in the *"ABB Procontic Programming System Library of Connection Elements ABB Procontic CS31"*). The same type variable with group number 0 and channel number 0 is used as a basic variable.

If the server polls values, the 3rd and 4th received words is a bit pattern to identify the data packages which must be send to the Server. The bit pattern is used in the "CE for bit pattern unpacking in the storage buffer".

If the server pokes data into the ABB CS31, the 3rd received word is poked value (write command is performed only for one Item). The 4th received word is "0". Example (current line comments starts by ";"):

```
$AREC
        #JOB        #00001
        UJOB        A62,04
        N00         MW003,01
        #D00        #00127
        #J00        #00010
        #L00        #00004
        MW00        MW005,00            ;The first received word
        #JR00       A62,05
```

Parameter J00 - Received Job Number. This parameter value must be equal to the parameter **Job Number** for Send Package (see **Topic Definition** dialog box in chapter "**Topic Definition Command**").

Parameter L00 - Number of words of the user data to be received. This parameter value must be equal to 4.

Received Package in case of read (poll) Item EW7,15 is shown in the following example :

Byte No	Content	Description
0	02	Station address of the sender
1	01	Station address of the receiver
2	F5	Continuous pointer (address of the first user data byte)
...	...	Unused
F5	7F	DIN identification (always 7Fh)
F6	0A	Low-byte of Job Number (parameter #J00 of AREC block)
F7	00	High-byte of Job Number (always 00)
F8	04	Low-byte of Point Type (for E-00, A-01, M-02, K-03, EW-04, AW-05, MW-06, KW-07)
F9	00	High-byte of Point Type (always 00)
FA	7F	Low-byte of Index (Index=Group Number*16 + Channel Number)(see chapter “**Item (Point) Naming**”)
FB	00	High-byte of Index
FC	01	The first byte of send package bit pattern
FD	00	The second byte of send package bit pattern
FE	00	The third byte of send package bit pattern
FF	00	The fourth byte of send package bit pattern

Received Package example in case of write (poke) value 50 into Item MW50,01 is shown below.

Byte No	Content	Description
0	02	Station address of the sender
1	01	Station address of the receiver
2	F5	Continuous pointer (address of the first user data byte)
...	...	Unimportant information
F5	7F	DIN identification (always 7Fh)
F6	0A	Low-byte of Job Number (parameter #J00 of AREC block)
F7	00	High-byte of Job Number (always 00)
F8	6A	Low-byte of Point Type (for A-65h(101), M-66h(102), AW-69h(105), MW-6Ah(106))
F9	00	High-byte of Point Type (always 00)
FA	21	Low-byte of Index (Index=Group Number*16 + Channel Number)(see chapter “**Item (Point) Naming**”)
FB	03	High-byte of Index
FC	32	Low-byte of write value.(50 = 32h)
FD	00	High-byte of write value.
FE	00	Not used
FF	00	Not used

CE for the first word (the 3rd word of the Received Package**) of bit pattern unpacking** into binary variables to enable correspondent $ASEND blocks. Example (current line comments starts by ";") :

```
$UNPACK
        WORT        MW005,02
        #n          #00016
        BI00        M001,00         ;CON 1
        BI01        M001,01         ;CON 2
        BI02        M001,02         ;CON 3
        BI03        M001,03         ;CON 4
        BI04        M001,04         ;CON 5
        BI05        M001,05         ;CON 6
        BI06        M001,06         ;CON 7
        BI07        M001,07         ;CON 8
        BI08        M001,08         ;CON 9
        BI09        M001,09         ;CON 10
        BI10        M001,10         ;CON 11
        BI11        M001,11         ;CON 12
        BI12        M001,12         ;CON 13
        BI13        M001,13         ;CON 14
        BI14        M001,14         ;CON 15
        BI15        M001,15         ;CON 16
```

Parameter WORT - Word variable to be unpacked. This parameter is the 3rd data word of the received package.

Parameters BI00...BI15 - binary variables used for the $ASEND block enabling. When you develop a PLC program, you must give an unique (1-32) **Configuration Order Number** (CON) to every send block. This CON does not depend on actual block sequence in the PLC program jet an every package send job enable variable must be set in the $UNPACK block according to the CON. For example, if the send block is given CON number 3, its job enable variable must be set in the $UNPACK in the position marked BI02. Given CON is used during **Topic** configuration (see chapter **Topic Definition Command**). Remember given **Configuration Order Numbers** and use them correctly ! If you disarrange CONs, correct PLC and ABB_ARCN Server performance will be impossible !

Perform this CE only in case of read (poll) package receiving for CONs 1-16.

CE for the second word (the 4th word of the Received Package) **of the bit pattern unpacking** into binary variables to enable correspondent $ASEND blocks. Example (current line comments starts by ";") :
Example :

```
$UNPACK
        WORT        MW005,03
        #n          #00016
        BI00        M002,00         ;CON 17
        BI01        M002,01         ;CON 18
        BI02        M002,02         ;CON 19
        BI03        M002,03         ;CON 20
        BI04        M002,04         ;CON 21
        BI05        M002,05         ;CON 22
        BI06        M002,06         ;CON 23
        BI07        M002,07         ;CON 24
        BI08        M002,08         ;CON 25
        BI09        M002,09         ;CON 26
        BI10        M002,10         ;CON 27
```

```
        BI11        M002,11         ;CON 28
        BI12        M002,12         ;CON 29
        BI13        M002,13         ;CON 30
        BI14        M002,14         ;CON 31
        BI15        M002,15         ;CON 32
```

For detailed description see "**CE for the first word of the bit pattern unpacking** into binary variables". Perform this CE only in case of read (poll) package receiving for CONs 17-32.

CE of indexed writing of binary variables. Use these CE only after write (poke) package receiving. The following CE are used for binary output (Item name - A) writing :

```
$=?         ;Compare Point Type in the received package and constant value
(MW003,11)
            ;If compared values are equal, set 1 into the Block enable parameter
        Z1=?        MW005,00        ;Point Type (1st receive word)
        Z2          MW003,11        ;Constant point type = 101
        Q           M000,13         ;Block enable

$SPBM       ;If a Block enable parameter M000,13 is 1, go to MRK 101
        E           M000,13
        MRK         MRK 101

$MRK
        MRK         MRK 101

$IDSB                               ;perform indexed writing
        FREI        M000,13         ;Block enable
        QUEL        M000,15
        INDX        MW005,01
        BASI        A00,00

$SPBM
        E           M000,13
        MRK         MRK 999
```

Parameter QUEL - source variable (value). This value is unpacked (before the current CE execution) from the 3rd data word of the received package. (In our case - from variable MW005,02).

Parameter INDX - index. This variable is the 2nd data word of the received package. (Index value is calculated in the ABB_ARCN DDE Server.)

Parameter BASI - basic variable.

The target variable for write results from the INDX and the basic variables. In address calculation the same type variable with group number 0 and channel number 0 (A00,00) is used as a basic variable.

Use the alike CE for binary flags (Item Name M) writing also.

CE of indexed writing of word variables. Use this CE only after write (poke) package receiving. The following CE is used for word flag (Item name - MW) writing :

```
$=?         ;Compare Point Type in the received package and constant value
(MW003,14)
```

```
                ;If compared values are equal, set 1 into the Block enable parameter
        Z1=?        MW005,00        ;Point Type (1st receive word)
        Z2          MW003,14        ;Constant point type = 106
        Q           M000,13         ;Block enable

$SPBM       ;If a Block enable parameter M000,13 is 1, go to MRK 101
        E           M000,13
        MRK         MRK 106

$IDSm
        FREI        M000,13
        QUEL        MW005,02
        INDX        MW005,01
        BASI        MW000,00
```

Parameter QUEL - source variable (value). This variable is the 3rd data word of the received package.

Parameter INDX - index. This variable is the 2nd data word of the received package. (Value is calculated in the ABB_ARCN DDE Server).

Parameter BASI - basic variable. The target variable results from the INDX and the basic variables. In address calculation the same type variable with group number 0 and channel number 0 (MW000,00) is used as a basic variable.

Use the same CE for word outputs (Item Name AW) writing also.

CE for data package send. The following group of CE is developed for 12 data packages sending to the ABB_ARCN DDE Server. The first one is a CE with Configuration Order Number (CON) 1:

```
$ASEND
        #JOB        #00012
        EN00        M001,00             ;CON 1
        #n00        #00001
        N00         MW003,02
        #D00        #00127
        #J00        #00010
        #L00        #00001
        ST00        A62,08
        MW00        EW07,15
```

Parameter EN00 - enables the send job. Variable in this parameter (here - M001,00) must be equal to the parameter BI00 in $UNPACK block (the 3rd word of the Received Package unpacking).

Parameter J00 - Job Number.

Parameter L00 - Number of words of the user data to be sent. The represented CE sends 1 data word - value of variable EW07,15.

CE with Configuration Order Number (CON) 2:

```
$ASEND+
        EN01        M001,01             ;CON 2
        #n01        #00001
        N01         MW003,02
        #D01        #00127
```

```
#J01        #00011
#L01        #00001
ST01        A62,09
MW01        MW003,00
```

Parameter EN01 - variable in this parameter (here - M001,01) must be equal to the parameter BI01 in the $UNPACK block (the 3rd word of the Received Package unpacking).

Parameter J01 - Job Number of the 2nd Send Package.

Parameter L01 - Number of words of the user data to be sent. The represented CE sends 1 data word - value of variable MW03,00.

CE with Configuration Order Number (CON) 8:

```
$ASEND+
        EN07        M001,07                 ;CON 8
        #n07        #00001
        N07         MW003,02
        #D07        #00127
        #J07        #00017
        #L07        #00125
        ST07        A62,15
        MW07        MW050,00
```

Parameter EN07 - variable in this parameter (here - M001,07) must be equal to the parameter BI07 in the $UNPACK block (the 3rd word of the Received Package unpacking).

Parameter J07 - Job Number of the Send Package.

Parameter L07 - Number of words of the user data to be sent. The represented CE sends 125 consecutive data words.

Parameter MW07 - The first word variable of the sent package (here - MW050,00).

CE with Configuration Order Number (CON) 12:

```
$ASEND+
        EN11        M001,11                 ;CON 12
        #n11        #00001
        N11         MW003,02
        #D11        #00127
        #J11        #00021
        #L11        #00002
        ST11        M000,11
        MW11        MW010,01
```

Parameter EN11 - variable in this parameter (here - M001,11) must be equal to the parameter BI11 in the $UNPACK block (the 3rd word of the Received Package unpacking).

Parameter J11 - Job Number of the Send Package.

Parameter L11 - Number of words of the user data to be sent. The represented CE sends 2 consecutive data words.

Parameter MW11 - The first word variable of the sent package (here - MW010,01). In this case - binary variables are previous packed by CE $PACK into the sent package word variables (MW010,01 and MW010,02).

Sent Package example in case of read Item MW050,01 is shown below:

Byte No	Content	Description
0	01	Station address of the sender
1	02	Station address of the receiver
2	03	Continuous pointer (address of the first user data byte)
3	7F	DIN identification (7Fh)
4	11	Low-byte of Job Number (parameter #J07 of $ASEND block)
5	00	High-byte of Job Number (always 00)
6	00	Low-byte of Item MW050,00 (the 1st variable of the user data words)
7	00	High-byte of Item MW050,00
8	32	Low-byte of Item MW050,01 (the 2nd variable of the user data words)
9	00	High-byte of Item MW050,01
...	...	Items from MW050,02 to MW057,12
FE	00	Low-byte of Item MW057,13 (the 125th variable of the user data words)
FF	00	High-byte of Item MW057,13

CE to initialize memory area in the operand memory with zero. We must set to zero word flags of received package after each processing of theirs:

```
$INITS
        0-1         A62,05
        #n          #00004
        VAR         MW005,00
```

All described CEs location and dependencies see in the full example text (\PROJEKT or \PROJEKT2).

For more detailed CE parameters explanation see *"ABB Procontic Programming System Library of Connection Elements ABB Procontic CS31".*

Notes on Write Command Processing

To process more reliable communication a response to write command must be transferred by PLC Application program to the Server after write command is received and successfully processed. If the system is configured to process responses to write commands then the Server can recognize whether write command is processed correctly or is it necessary to retry the write command.
To configure the Server to process response to write command an item **WriteRetryCount** (see chapter *Troubleshooting*) must be included into WIN.INI file and non-zero value should be assigned to it. Value assigned to **WriteRetryCount** item is number of retries to process write command in case of fault. Job Number for Send Package must be used for response to write command and can not be used for response to read commands.

A response package should contain word value 999 (3E7 hex). In WW Logger file such response is similar to the following message:
... ABB_ARCN R: 02 01 FB .. 7F 1F 00 E7 03

To create such response package the PLC application program Projekt2 contains the following additional CEs:
- $INITV assigning constant value 999 to MW003,15;

- $ >= to assign 1 to $ASEND enabling bit M003,00 if User First Word in received package indicates data write command;

```
$ >=
        Z1>=  MW005,00          USER FIRST WORD
        Z2    MW003,11          const. = 101 (A write)
        Q     M003,00
```

- $ASEND to store job #31 as data package in storage buffer. Write command response data package contains constant value 999;

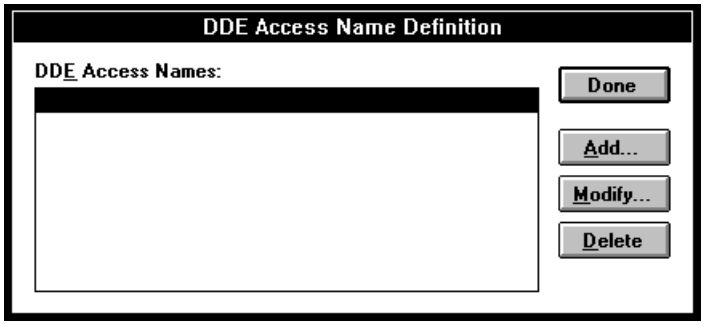
```
$ASEND+
        EN    M003,00
        #n    #00001
        N     MW003,02
        #D    #00127
        #J    #00031
        #L    #00001
        ST    M000,12
        MW    MW003,15          ;CONSTANT VALUE 999
```

Note: *To access data from PLC running example program Projekt2\ABB_ARCN the Server should be started with configuration file ABB_ARCN.CFG (supplied with installation disks) and Topic configuration parameter* ***Job Number for Send Package*** *should be changed from 10 to 31.*

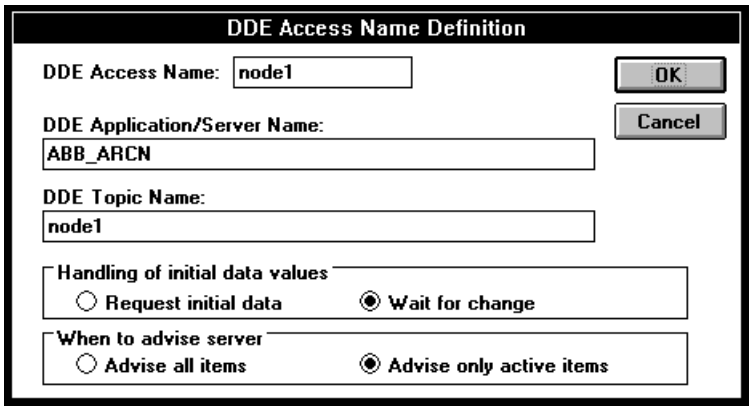
Using the ABB_ARCN Server with InTouch

To access items/points on ABB CS31 07KT93 Controllers from **InTouch**, the following steps (all performed in **WindowMaker**) are required:

To define the DDE Access Names in WindowMaker for each Controller invoke the */Special/DDE Access Name Setup* command. The "Select a DDE Access Name" dialog box will appear.



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



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

The following three fields are required entries when entering a DDE Access Name Definition:

DDE Access Name
Enter an arbitrary name which will be used by **InTouch** to refer to the topic (Controller). It is recommended that the name defined for the topic (Controller) in ABB_ARCN also is used here.)

DDE Application/Server Name
Enter the application name, **ABB_ARCN**, which the DDE Server is used to accessing the Controller.

DDE Topic Name
Enter the name defined for the topic in ABB_ARCN to identify the Controller the ABB_ARCN Server will be accessing.

Note:** This will usually be the same as the "DDE 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 ABB_ARCN Server.

Request Initial Data
This option may be selected if the Server is other than a Wonderware DDE Server **and** the Server does not return data values immediately when a window is displayed. This option is not applicable to the ABB_ARCN DDE Server.

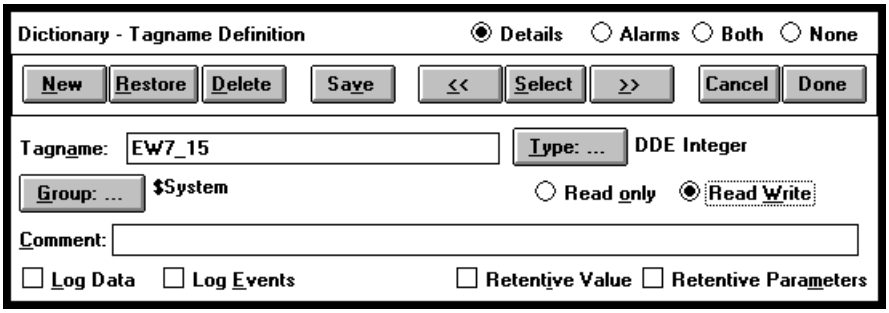
Wait for Change
This option should be selected for the ABB_ARCN DDE Server.

Advise all Items
This option may be selected if the Server is to poll for all data whether or not it is in visible windows, alarmed, logged or trended. Use of the option is not recommended.

Advise only active Items
Selecting this option will cause the ABB_ARCN Server to poll only points in visible windows and points that are alarmed, logged or trended.

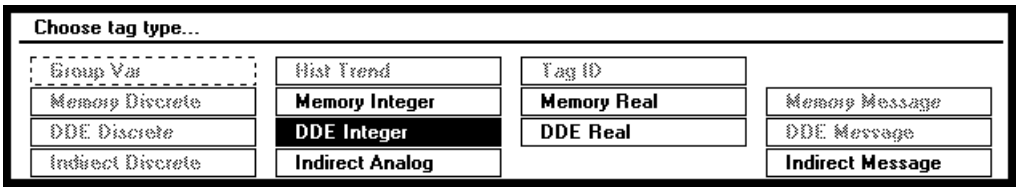
Defining the Tag names

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



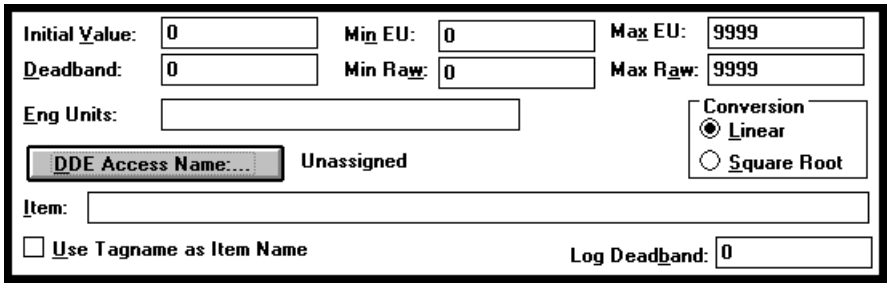
Enter the **Tag Name**. (The tag name defined here is the name **InTouch** will use. The ABB_ARCN Server does not see this name.)

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

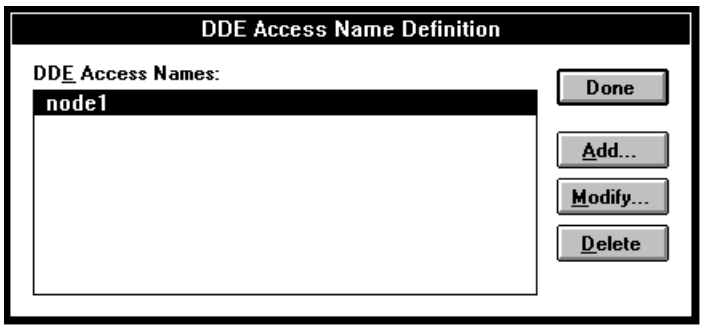


To access ABB_ARCN items , the type must be **DDE Discrete** or **DDE Integer.** Select the DDE type.

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



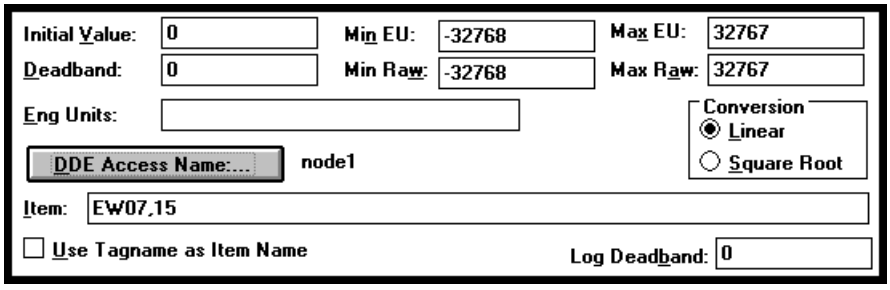
Select the ABB_ARCN topic (Controller) by clicking on the **DDE Access Name**... button. The "Select a DDE Access Name" dialog box will appear:



Select the appropriate topic name and click on **OK**. (If the DDE Access Name has not been defined as previously described, click on **New** and define the DDE topic now.)

For integers and reals, 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** should be equal to **Max Raw**.

Enter the ABB_ARCN item/point name to be associated with this tag name in the **Item** field in the "Details" box:



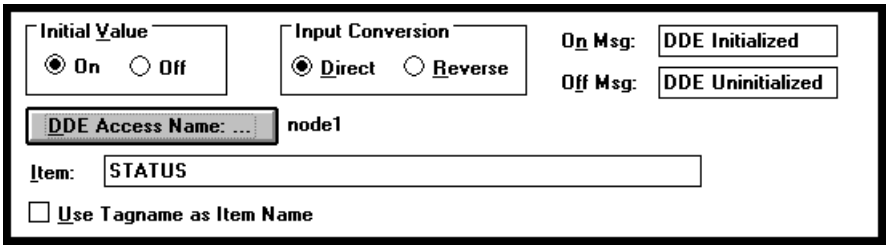
(Refer to the **Item (Point) Naming** section below for complete details.)

Where applicable, the **Use Tag name as Item Name** option may be selected to enter automatically 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.*
When all entries have been made, click on the **Save** button (in the top dialog box) to accept the new tag name. To define additional Tag names click on the **New** button. To return to the **WindowMaker** main screen, select **Done**.

Controller "STATUS" Item

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

From **InTouch** the state of communication with the Controller may be read by defining a DDE Discrete tag name and associating it with the topic configured for the Controller and using **Status** as the *Item* name.



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

=ABB_ARCN|topic!STATUS

Item (Point) Naming

Within the ABB_ARCN Server, item/point naming depends on ABB Procontic CS31 system operand naming conventions. The tables below list the item/point names supported by the ABB_ARCN DDE Server:

BINARY points

Item name	Description	DDE Tag Type	Range
E	BINARY input	Discrete	E00,00...E63,15
A	BINARY output	Discrete	A00,00...A63,15
M	BINARY flag	Discrete	M00,00...M255,15
K	BINARY Indirect constant	Discrete	K00,00...K00,01

WORD points

Item name	Description	DDE Tag Type	Range
EW	WORD input	Integer	EW00,00...EW07,15
AW	WORD output	Integer	AW00,00...AW6,15
MW	WORD flag	Integer	MW000,00...MW255,15
KW	WORD Indirect constant	Integer	KW00,00...KW39,15

It is recommended that you use consecutive addressing for item/point naming (for example, MW001,01, MW001,02, MW001,03, etc.). This will greatly increase the performance of ABB_ARCN DDE Server.

Special write-only DDE Discrete Item **SUSPEND** may be used to control communication with a separate topic. If application changes Item's value from 0 to 1 - then communication with topic is suspended. If Item's **SUSPEND** value is changed back to 0, communication with topic is resumed.
***Note**: When topic is suspended by setting **SUSPEND** item to 1, the Server rejects all write values to this topic.*

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;
- A03,00 BINARY Output, Group number 3, Channel number 0;
- M113,13 BINARY Flag, Group number 113, Channel number 13;
- K00,00 BINARY Constant, Group number 0, Channel number 0;
- EW02,01 Analog WORD Input, Group number 2, Channel number 1;
- AW03,02 Analog WORD Output, Group number 3, Channel number 2;
- MW255,11 WORD Flag, Group number 255, Channel number 11;
- KW38,04 WORD Constant, Group number 38, Channel number 4;

Notes on Using Microsoft Excel

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

=ABB_ARCN|topic!item

Sometimes, Excel requires the topic and/or item/points 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/point 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:

= ABB_ARCN |'node1'!'EW7,15'

Writing Values to ABB_ARCN Points

Values may be written to the ABB_ARCN Server by creating an Excel macro which uses the DDE **"POKE"** Command. The command is entered in Excel as follows:

=INITIATE("ABB_ARCN","Topic Name")
=POKE(Channel_Id,"Item",Data_Reference)
=RETURN()

For Channel_Id, use the cell reference where the **"INITIATE"** statement is found.
For Data_Reference, use the cell identification which contains the **"POKE"** value.
Some applications have a limited number of channels; therefore, they should be closed when finished by using a **"TERMINATE"** statement in the macro. The **"TERMINATE"** macro should have sufficient delay to ensure that the **"POKE"** has been executed. (Refer to the Excel Manual for further details.)

Improving Excel/DDE Server Performance

To improve communication performance between Excel and the DDE Server, the following lines should be added to the **WIN.INI** file under the DDE Server heading:

AllowXLTable=1
DDE Server will accept XLTable format initializations from Excel. This will significantly speed up the initialization of Excel/Server Conversation.

AllowXLTableForPokes=1
DDE Server will accept XLTable format pokes from Excel. This will significantly speed up the poke operations from Excel.

For example:

[ABB_ARCN]
AllowXLTable=1
AllowXLTableForPokes=1

Troubleshooting

WIN.INI entries

The first time you run the ABB_ARCN DDE Server configuration, most of the items in the following list will automatically appear in the WIN.INI file. It is usually in the C:\WINDOWS directory. 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 DOS text*). The following is a typical entry for the ABB_ARCN DDE Server:

[ABB_ARCN]
ProtocolTimer=50
RequestTimer=1000
ValidDataTimeout=60000
DDEBlockSize=4096
WriteRetryIndefinitely=0
ConfigurationFile=C:\ABB_ARCN
WinIconic=0
WinFullScreen=0
WinTop=112
WinLeft=0
WinWidth=200
WinHeight=168
DebugMenu=1
ShowSend=0
ShowReceive=0
ShowErrors=1
Verbose=0
WriteRetryCount=3
ConsecArcnErrorsForLog=1
maxPossiblePackages=4
ShowAllRejectedWrites=4

The **WriteRetryCount** entry is used to configure the Server to wait the responses to write commands. If you want to receive responses to write messages then some non-zero value must be assigned to **WriteRetryCount** item. A good choice is 2 - with this setting the Server performs the second write attempt if the first attempt was unsuccessful. If **WriteRetryCount** value is 3 then three write attempts are performed, etc. The PLC program should be able to send a response to write command (like the ABB CS31 example program PROJEKT2).
If **WriteRetryCount** value is 0 then the response to write messages is not sent (as in ABB CS31 example program PROJEKT). In this case the Server internally is set to perform the second write attempt if write command is not transferred to PLC. The Server can not get information if write command is processed inside PLC therefore write command execution in this case is not reliable enough.

The **ConsecArcnErrorsForLog** entry is used to control number of error messages logged to WW Logger file.

(1) If **ConsecArcnErrorsForLog =0** then each time when ARCNET communication error happens (and **Show Errors** is checked in the Server's System Menu) an error message is logged to WW Logger file.
(2) If **ConsecArcnErrorsForLog =1...4** then ARCNET communication error message is logged to WW Logger file only if consecutive 2...5 ARCNET communication errors happen.
Assigning non-zero value to **ConsecArcnErrorsForLog** item in WIN.INI file you can protect Logger file from huge amount of random communication error logging. Random communication errors may happen but such errors do not affect Server's work, therefore it is not reasonable to log all random errors.
Do not assign values greater then 4 to **ConsecArcnErrorsForLog** item.

The **maxPossiblePackages** entry is used to control number of simultaneously requested data packages. If there is no such item in WIN.INI file than the Server each time tries to request as many packages as possible. It increases amount of communication data but makes WW monitoring slower. Using of **maxPossiblePackages** item allows to access optimal data update rate as well as good monitoring speed.
To find the best value of **maxPossiblePackages** item you can check item **ShowReckPackNumb** in the Server's System Menu (see chapter *Troubleshooting menu*) and run the system. WW Logger file will show number of received packages after each request. Find in Logger file the greatest number of received packages and assign it to **maxPossiblePackages** item in WIN.INI file. Restart the Server. Do not check **ShowReckPackNumb** during normal working process!

The **ShowAllRejectedWrites** option is useful when communication with a separate topic (topics) is suspended by Item SUSPEND (see chapter **Item (Point) Naming**) and the Server rejects each write to this topic (topics). If **ShowAllRejectedWrites** = 1 then information about each rejected write value is displayed by the Server. If **ShowAllRejectedWrites** = 0 then the Server rejects each write to the suspended topics without logging any information.

The default values (they are used if WIN.INI file does not contain these entries) are the following: **WriteRetryCount =0**, **ConsecArcnErrorsForLog =1**, **maxPossiblePackages =8** and **ShowAllRejectedWrites =1**.

Troubleshooting menu

The following debugging choices are appended to the Server's System Menu (the menu that appears when you click the "-" box 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 PLCs.
ShowEvents - if checked then information about suspended and resumed topics are displayed
Show Send - if checked then outgoing data is logged in hexadecimal format.
Show Receive - if checked then incoming data is logged in hexadecimal format.
Show Errors - if checked then errors are logged.

ShowReckPackNumb - if checked then number of received ARCNET packages is logged after each request. Option is useful only for system configuration, not for normal data processing!

Verbose - if checked then all errors are logged.

Dump Screen - if checked then all information about packets is displayed in Server's window.

Dump - displays all information about board, active topics and data items.

All debug information (except **Dump Screen**) 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
ABB_ARCN DDE Server
Revision History

Feb. 95	Rev 1.0	First Release
May 95	Rev 2.0	Windows NT support added. Server configuration dialog boxes changed.
Mar 96	Rev 2.1	Installation Program added.
Aug 97	Rev 2.2	Manual text modified. Manual file name changed.
Nov 97	Rev 2.3	Troubleshooting features changed. SUSPEND item added. Chapter Notes on Write Command Processing added.
Jul 2001	Rev 2.4	Support for SOHARD ARCNET PCI card added.
Dec 2002	Rev 2.5	SuiteLink support added.