

CIMD2IOS Communication Server

for Microsoft Windows
and InTouch Applications

**User Manual
Ver 1.x Rev 1.0
DR 460 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

| | |
|--|----|
| Overview..... | 1 |
| Communication Protocols..... | 1 |
| Accessing Remote Items via the Server..... | 2 |
| Installing the CIMD2IOS Server | 3 |
| Installing the Server | 3 |
| Installing the I/O Server Infrastructure | 4 |
| Configuring the CIMD2IOS Server | 5 |
| Server Settings Command..... | 5 |
| Socket Settings Command | 7 |
| Saving CIMD2IOS Configuration File | 8 |
| Configuration File Location | 8 |
| Topic Definition Command | 9 |
| Item Names | 10 |
| Examples | 11 |
| Built-in I/O items | 14 |
| Monitoring Communication with SMSC | 15 |
| Using the CIMD2IOS Server with InTouch | 15 |
| Defining the Access Names..... | 15 |
| Defining the Tag names..... | 17 |
| Monitoring the Status of Communication with InTouch..... | 19 |
| Notes on Using Microsoft Excel..... | 20 |
| Reading Values into Excel Spreadsheets..... | 20 |
| Writing Values to CIMD2IOS Points | 20 |
| Troubleshooting..... | 22 |
| WIN.INI entries | 22 |
| Troubleshooting menu | 22 |

CIMD2IOS Communication Server

Overview

The **CIMD2IOS Communication Server** (hereafter referred to as the “CIMD2IOS Server” or “CIMD2IOS” or “Server”) is a Microsoft Windows 32-bit application program that acts as a communication protocol Server and allows other Windows application programs to exchange data with remote GSM devices by sending and receiving SMS messages through Short Messages Service Centers (SMSCs) by using the CIMD2 (Computer Interface to Message Distribution) protocol. The CIMD2IOS I/O Server is primarily intended for use with **Wonderware InTouch**, but any Microsoft Windows program that is capable of acting as a DDE, FastDDE or SuiteLink *Client* may use the CIMD2IOS I/O Server.

The CIMD2IOS I/O Server requires an Ethernet card and TCP/IP protocol (supporting Windows Sockets interface) to be installed on the computer to communicate with SMSC over TCP/IP network. The CIMD2IOS I/O Server supports freely configurable data formatting and extracting from SMS messages - all configuration information related to SMS messages (including I/O item naming) is stored in manually prepared text file (config.txt), which is read at CIMD2IOS I/O Server startup.

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 CIMD2IOS 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 CIMD2IOS 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:

To obtain the data from *Server*, the *client* program (e.g. InTouch or MS Excel) must open a channel to the *Server* by specifying three things: the *Server application name* (**CIMD2IOS**), the *Topic name* (unique name, e.g. **Topic1**) and the specific *item name* (e.g. **RL001_002_pres01**).

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 by sending and receiving SMS messages through SMSC by using the CIMD2 protocol, the application portion of the address is **CIMD2IOS**.

topic name

Meaningful names are configured in the Server to identify specific group of items. These names are then used as the topic name in all conversations to that device. For example, **Topic1**.

Note! You can define multiple topic names for the SMSC.

item name

A specific data element within the specified topic. For example, **RL001_002_pres01**. 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 CIMD2IOS Server

Installing the Server

The CIMD2IOS Server installation package can be supplied:

1. As a self-extracting archive 46010xxx.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 CIMD2IOS Server from the self-extracting archive, run the 46010xxx.EXE and proceed as directed by the CIMD2IOS Server Setup program.

To **install** the CIMD2IOS 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 CIMD2IOS 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 "CIMD2IOS SL and DDE Server" and click on "Setup...".
5. Proceed as directed by the CIMD2IOS Server Setup program.

When installation is finished, the subdirectory specified as a folder where to install the CIMD2IOS Server files will contain the following files:

| | |
|---------------------|---|
| CIMD2IOS.EXE | The CIMD2IOS Server Program. This is a Microsoft Windows 32-bit application program. |
| CIMD2IOS.CHM | The CIMD2IOS Server Help file. |
| CIMD2IOS.CFG | An example configuration file. |
| CONFIG.TXT | An example input file with all information about data formatting and extracting from SMS messages and I/O item names supported. |
| LICENSE.TXT | Klinkmann Automation software license file. |

To **uninstall** the CIMD2IOS Server, start Control Panel, select “Add/Remove Programs” and select the “CIMD2IOS 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 CIMD2IOS 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 CIMD2IOS 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 CIMD2IOS 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 CIMD2IOS Server and these infrastructure files must be install prior to executing the CIMD2IOS Server.
3. The HASP key is needed for full time running of CIMD2IOS Server. The HASP Driver setup is performed during the Server setup. Without HASP Driver installed and HASP key plugged into PC parallel port, the CIMD2IOS Server will run only 1 hour (with all features enabled).

Installing the I/O Server Infrastructure

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

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

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

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

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

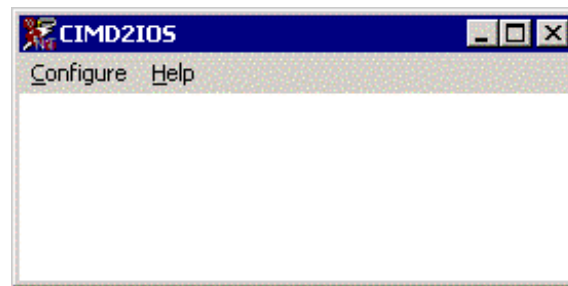
To **uninstall** the I/O Server Infrastructure, start Control Panel, select “Add/Remove Programs” and select the “IO Server Infrastructure” from the list of available software products. Click on “Add/Remove...” and proceed as directed by the UnInstallShield program.

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

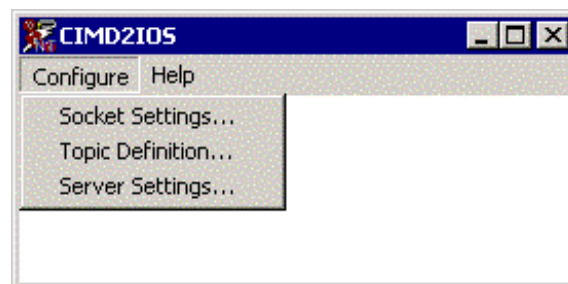
Configuring the CIMD2IOS Server

After the CIMD2IOS Server is initially installed, a little of configuration is required. Configuring the Server automatically creates a **CIMD2IOS.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 **CIMD2IOS** 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 CIMD2IOS program. If the Server starts up as an icon, double-click on the icon to open the server's window. The following will appear:



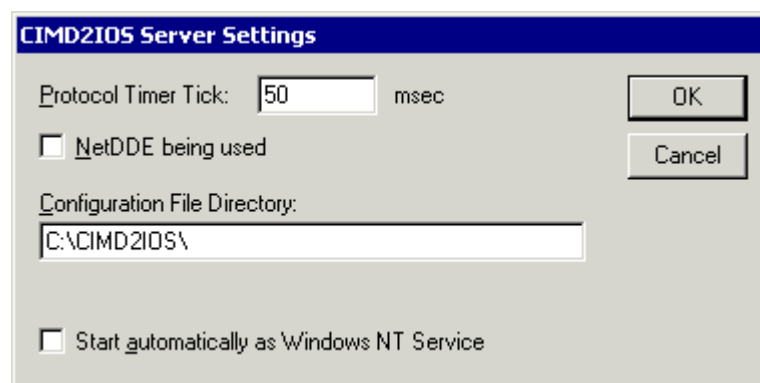
To access the commands used for the various configurations, open the */Configure* menu:



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:



The following describes each field in this dialog box:

Protocol Timer Tick

This field is used to change the frequency at which the Server is continuously activated (the Server checks for work to do). At this frequency the Server checks if there is something received or if it necessary to send something out. If the send/response cycle is too long then more than one activation of Server is necessary to process it. If computer is very busy or some other MS Windows application is taking over the computer then the Server is activated rarely than setting in the **Protocol Timer Tick**.

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

NetDDE being used

This option has to be selected if CIMD2IOS I/O Server will be accessed from remote I/O Client(s) by using NetDDE.

Configuration File Directory

This field is used to specify the path (disk drive and directory) in which CIMD2IOS I/O Server will save its current configuration file. CIMD2IOS I/O Server will use this path to load the configuration file the next time it is started.

Start automatically as Windows NT Service

Enabling this option will cause the CIMD2IOS I/O Server to start as a Windows NT service.

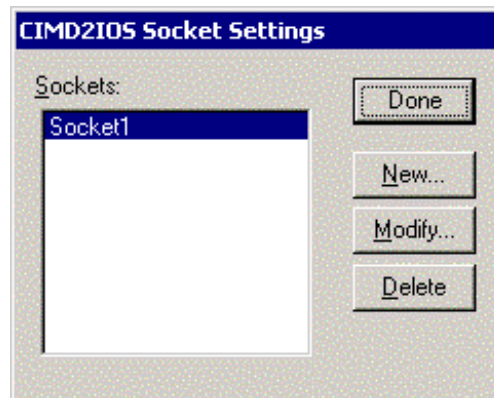
Windows NT offers the capability of running applications even when a user is not logged on to the system. This is valuable when systems must operate in an unattended mode. Enabling this option and rebooting the system will cause the Server to run as a Windows NT service. However, to view configuration 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 **CIMD2IOS_IOServer** is selected). If **Allow Service to Interact with Desktop** is not selected then CIMD2IOS Server full functionality is not ensured (e.g. the Server configuration can not be changed, no message boxes will be displayed, etc.).*

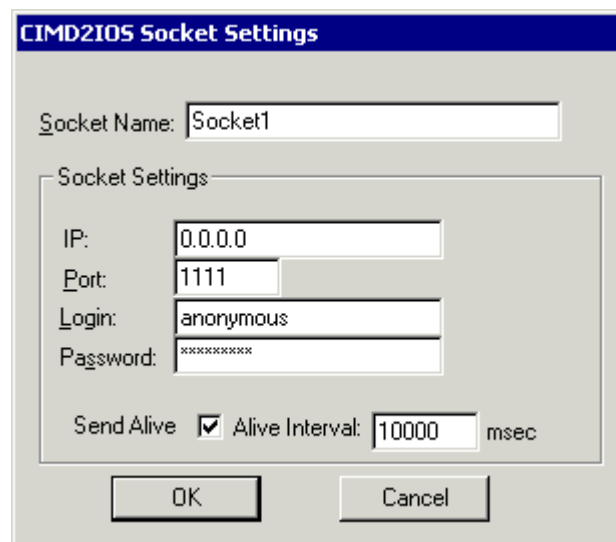
Once all entries have been made, click on **OK**.

Socket Settings Command

To configure the Socket used for communication with SMSC, invoke the `/Configure/Socket Settings...` command. The "CIMD2IOS Socket Settings" first dialog box will appear:



To modify or examine an existing Socket, select the topic name and click on **Modify**. To define a new Socket, click on **New**. The "CIMD2IOS Socket Settings" second dialog box will appear:



The following describes each dialog field in this dialog box - for each connection with SMSC (in general the connection can be established with several SMSCs), the following Socket settings are possible to set:

Socket Name

Enter the Socket name and later use it in *Topic Definition*. Only one Socket with same **IP** (Internet Address) and **Port** (Port Number) can be defined.

IP

The Internet Address of SMSC where to connect.

Port

The Port Number of SMSC where to connect.

Login and Password

The Login and Password used in *Login* operation when connecting to SMSC.

Send Alive and Alive Interval

If **Send Alive** is checked then CIMD2IOS I/O Server at **Alive Interval** (default 10 seconds) will execute the *Alive* operation - will check whether the link between this Socket and the SMSC is still alive.

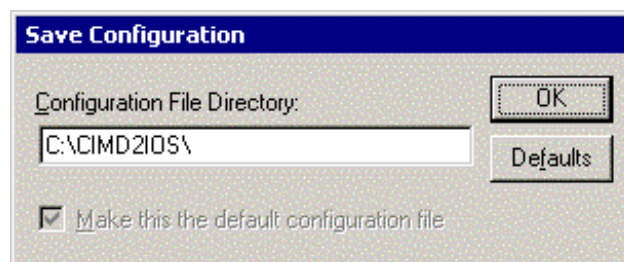
Once all entries have been made, click on **OK** to process the configuration for the Socket. The "CIMD2IOS Socket Settings" first dialog box will appear again.

Click on **Done** when configuration for all Sockets has been performed.

Note: *If this is the first time the Sockets have been configured, the user will be prompted to save configuration to an existing directory.*

Saving CIMD2IOS 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 CIMD2IOS Server to find the configuration file automatically each time it is started.

Configuration File Location

When the CIMD2IOS 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:

CIMD2IOS /d:c:\directoryname

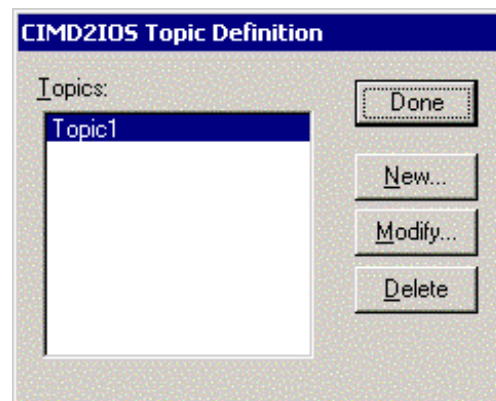
Note: *There is no limit to the number of configuration files that may be created, although each must be in a separate directory.*

Topic Definition Command

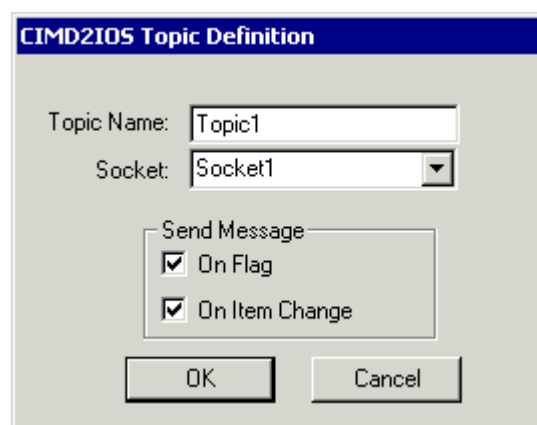
Meaningful names are configured in the Server to identify specific group of items. These names are then used as the topic name in all conversations to that group.

The following steps are taken to define the Topic:

1. Invoke the *Configure/Topic Definition...* command. The "CIMD2IOS Topic Definition" first 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 "CIMD2IOS Topic Definition" second dialog box will appear:



For each Topic it is possible to set the **Topic Name** and **Socket** used by this Topic.

Additionally there is possible to set up the **Send Message** options - the conditions when SMS messages will be sent to SMSC:

- if "Send Message/On Flag" is selected then SMS message will be sent on special built-in integer I/O item SEND value change (see *Send messages* examples in *Item Names* section later in this manual);
- if "Send Message/On Item Change" is selected then SMS message will be sent in case value changes at least of one I/O item linked with SMS message (see *Third Send message* and *Fourth Send message* examples in *Item Names* section later in this manual);

- if both "Send Message/On Flag" and "Send Message/On Item Change" are not selected then any sending of SMS messages to SMSC is disabled.

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

Select **Done** in the "CIMD2IOS Topic Definition" first dialog box when configuration for all Topics has been performed.

Item Names

All information about data formatting and extracting from SMS messages and I/O item names supported by CIMD2IOS Server is stored in manually prepared text file with fixed name **config.txt**. This file is read at CIMD2IOS Server startup.

There are two main parts of config.txt file - one for **Receive** messages and another for **Send** messages. The start of **Receive** messages part is marked by line containing **[RM]** and the start of **Send** messages part is marked by line containing **[SM]**.

The description of each message can contain the following parts:

- the **number of message** (e.g. **2.**); messages have to be numbered starting from **1**. (dot must present);
- the **phone number** or * indicating any (all) phone numbers; this field has to be ended with **#** character;
- the **text of message** (can include one or several **I/O item fields** marked by **{}**, see below);
- the **;** (semicolon), which indicates the end of message.

The **I/O item fields** are enclosed in **{}** and in real SMS messages everything between **{}** is replaced by I/O item value. The **I/O item field** contains the I/O item name and for non-integer items also **<d>** or **<rn>** indicating I/O item type - correspondingly Discrete or Real (for Real I/O items **n** is the precision - digits after decimal point).

For special **measurement messages** (including several measurement values for one I/O item, see Third and Fourth Receive message examples below) the special marking between **number of message** and **phone number** is used, in general it can be described as

s(i)n

where:

- s** – sequence of measurements: characters **A(a)** or **D(d)**, indicating ascending or descending sequence;
- i** – interval between measurements in minutes;
- n** – number of measurements in the message.

Examples

The following could be some example of **config.txt** file, including all possible cases:

```
[RM]
1.+358777666555#RL001_002_press01:{RL001_002_press01};
2.*#RL001_002_temp2:{<r>RL001_001_temp2}C filt01:{<d>RL001_001_filt01};
3.A(10)6#+358555666777#RL001_002_temp02:{<r>RL001_002_temp02};
4.D(10)6###RL001_002_temp02:{<r>RL001_002_temp02};
[SM]
1.+358777666555#poll;
2.*#poll;
3.+358777666555#RL001_002_setpoint01:#{RL001_002_setpoint01}#;
4.+358444555666#RL003_004_setpoint01:#{<r2>RL003_004_setpoint01}#RL003_004_
lock02:#{<d>RL003_004_lock02}#;
5.*#lock:#{<d>Lock}#;
```

where the contents of Receive and Send messages is following:

First Receive message

```
1.+358777666555#RL001_002_press01:{RL001_002_press01};
```

This message is expected from phone number **+358777666555** and will contain the value of I/O Integer item **RL001_002_press01**. Example:

```
RL001_002_press01:200
```

If above SMS message is received then new value **200** will be transferred to I/O Integer item **RL001_002_press01**.

Second Receive message

```
2.*#RL001_002_temp2:{<r>RL001_001_temp2}C filt01:{<d>RL001_001_filt01};
```

This message is expected from any phone number and will contain the values of I/O Real item **RL001_001_temp2** and I/O Discrete item **RL001_001_filt01**. Example:

```
RL001_002_temp2:24.5C filt01:1
```

If above SMS message is received then new value **24.5** will be transferred to I/O Real item **RL001_001_temp2** and new value 1 - to I/O Discrete item **RL001_001_filt01**.

Third Receive message

```
3.A(10)6###RL001_002_temp02:{<r>RL001_002_temp02};
```

This message is expected from any phone number and will contain 6 values (measurements) of I/O Real item **RL001_001_temp2**; the values are received in **ascending** order (current value last) and interval between measurements is **10** minutes. Example:

```
RL001_002_temp02: 60.4, 59.3, 59.4, 59.0, 58.3, 60.8
```

If above SMS message is received and SMS message's time stamp is e.g. 13:40 (the SMSC time stamp), then new values are transferred to I/O Real item **RL001_002_temp02** in following order and having the following time stamps:

| | |
|------|-------|
| 60.4 | 12:50 |
| 59.3 | 13:00 |
| 59.4 | 13:10 |
| 59.0 | 13:20 |
| 58.3 | 13:30 |
| 60.8 | 13:40 |

Fourth Receive message

4.D(10)6#+358555666777#RL001_002_temp02:{{<r>RL001_002_temp02}};

This message is expected from phone number **+358555666777** and will contain 6 values (measurements) of I/O Real item **RL001_001_temp2**; the values are received in **descending** order (current value first) and interval between measurements is **10** minutes. Example:

RL001_002_temp02: 60.4, 59.3, 59.4, 59.0, 58.3, 60.8

If above SMS message is received and SMS message's time stamp is e.g. 13:40 (the SMSC time stamp), then new values are transferred to I/O Real item **RL001_002_temp02** in following order and having the following time stamps:

| | |
|------|-------|
| 60.8 | 12:50 |
| 58.3 | 13:00 |
| 59.0 | 13:10 |
| 59.4 | 13:20 |
| 59.3 | 13:30 |
| 60.4 | 13:40 |

First Send message

1.+358777666555#poll;

This message will be sent to phone number **+358777666555** if built-in integer I/O item **SEND** value changes to 1. Example:

poll

Second Send message

2.*#poll;

This message will be sent to all phone numbers included in **config.txt** file (also to phone numbers, which belongs to **Receive** messages) if built-in integer I/O item **SEND** value changes to **2**. Example:

poll

Third Send message

3.+358777666555#RL001_002_setpoint01:#{RL001_002_setpoint01}#;

This message, containing the current value of I/O Integer item **RL001_002_setpoint01**, will be sent to phone number **+358777666555** at following conditions:

- 1) if “Send Message/On Flag” is checked at “CIMD2IOS Topic Definition” dialog box, built-in integer I/O item **SEND** value changes to **3** and the *Client* application (InTouch, Excel etc.) has sent (poked) at least one new value to CIMD2IOS Server;
- 1) if “Send Message/On Item Change” is checked at “CIMD2IOS Topic Definition” dialog box and if I/O Integer item **RL001_002_setpoint01** value changes in the *Client* application.

Example:

RL001_002_setpoint01:#130#

The above SMS message contains the current value **130** of I/O Integer item **RL001_002_setpoint01** (the ## in the text of message is a required syntax to receive SMS messages in Unitronics M90 PLC).

Fourth Send message

4.+358444555666#RL003_004_setpoint01:#{<r2>RL003_004_setpoint01}#RL003_004_lock02:#{<d>RL003_004_lock02}#;

This message, containing the current values of I/O Real item **RL003_004_setpoint01** and I/O Discrete item **RL003_004_lock02**, will be sent to phone number **+358444555666** at following conditions: Example:

- 1) if “Send Message/On Flag” is checked at “CIMD2IOS Topic Definition” dialog box and if built-in integer I/O item **SEND** value changes to **4**;
- 1) if “Send Message/On Item Change” is checked at “CIMD2IOS Topic Definition” dialog box and if I/O Real item **RL003_004_setpoint01** or I/O Discrete item **RL003_004_lock02** value changes in the *Client* application.

Note - the message will be sent only if both (all) current values are available; if I/O item does not exist or *Client* application has not sent at least one new value of each I/O item to CIMD2IOS Server then message will not be sent.

Example:

RL003_004_setpoint01:#177.52#RL003_004_lock02:#0#

The above SMS message contains the current value **177.52** of I/O Real item **RL003_004_setpoint01** (precision is 2) and the current value **0** of I/O Discrete item **RL003_004_lock02**.

Built-in I/O items

1) General items

SEND - I/O Integer, Read/Write (initial value has to be 0), used to send SMS messages if “Send Message/On Flag” is checked at “CIMD2IOS Topic Definition” dialog box

STATUS - I/O Discrete, Read Only, indicates the state of communication with SMSC: STATUS is **1** (on) if communication is OK, **0** (off) - if no communication (no connection, no Alive acknowledgment)

ALIVE - I/O Integer, Read/Write, can be used to change the **Alive Interval** value at CIMD2IOS Server runtime

ERROR - I/O Integer, Read/Write, contains CIMD2IOS Server current internal error value (0 – no error)

2) Monitoring items

RECV_MESSAGE - I/O Message, Read Only, contains the text of last received SMS message

SENDER_NUM - I/O Message, Read Only, contains the sender phone number of last received SMS message

RECV_MSG_DATE - I/O Message, Read Only, contains the date of last received SMS message

RECV_MSG_TIME - I/O Message, Read Only, contains the time of last received SMS message

3) Items allowing free sending of SMS messages apart from config.txt contents (even if config.txt is empty or does not exist)

MESSAGE - I/O Message, Read/Write, contains the text of SMS message to be sent

NUMBER - I/O Message, Read/Write, contains the phone number where to send the SMS message

SEND_MESSAGE - I/O Discrete, Read/Write (initial value has to be 0), used to send the SMS message containing the current **MESSAGE** text to current **NUMBER** phone number. Initially this item has to be set 0, to start the sending the *Client* should set this item to 1. After the SMS message is sent, CIMD2IOS Server returns this item value back to 0.

Monitoring Communication with SMSC

For each topic, there is a built-in discrete item that indicates the state of communication with SMSC. 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 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:

=CIMD2IOS|topic!STATUS

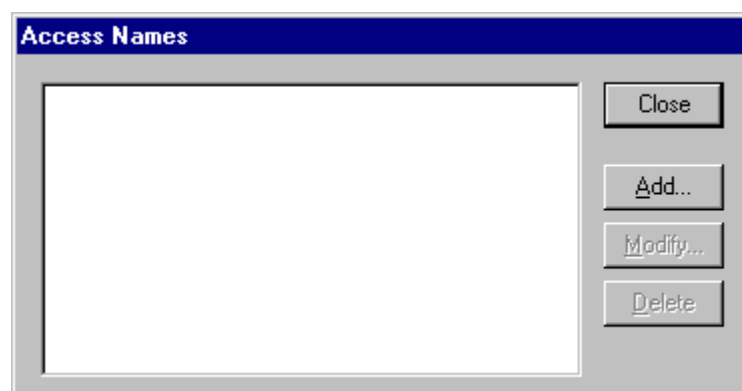
Using the CIMD2IOS Server with InTouch

To access CIMD2IOS items/points 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" or "Modify Access Name" dialog box will appear:

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

- Access:** Text box containing "CIMD2IOS".
- Node Name:** Empty text box.
- Application Name:** Text box containing "CIMD2IOS".
- Topic Name:** Text box containing "Topic1".
- Which protocol to use:** Radio buttons for "DDE" and "SuiteLink" (selected).
- When to advise server:** Radio buttons for "Advise all items" and "Advise only active items" (selected).
- Buttons:** "OK" and "Cancel" buttons on the right side.

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 CIMD2IOS Server the **CIMD2IOS** is used. Do not enter the .exe extension portion of the program name.

Topic Name

Enter the name defined for the topic in the CIMD2IOS Server to identify the topic the CIMD2IOS Server will be accessing.

The Topic Name is an application-specific sub-group of data elements. In the case of data coming from CIMD2IOS Server program, the topic name is the exact same name configured for the topic in the CIMD2IOS 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 **Configuring the CIMD2IOS Server** section.

Which protocol to use

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

When to advise server

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

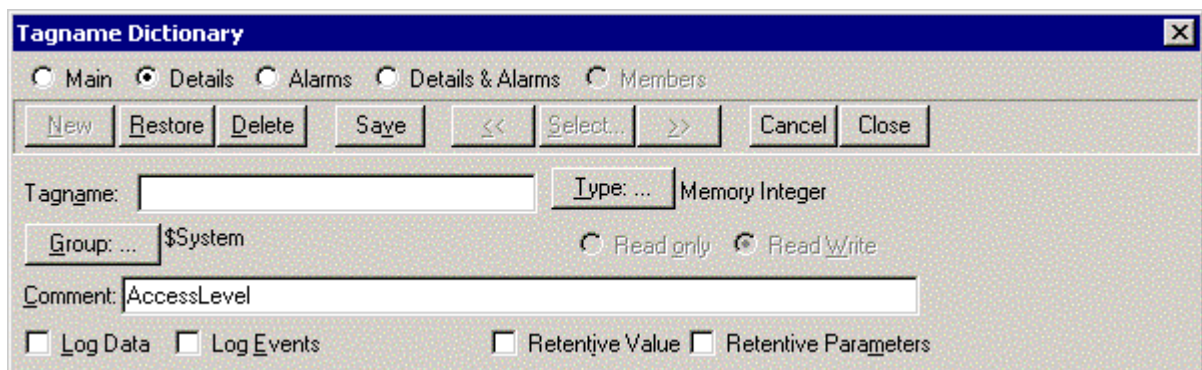
Select **Advise only active items** if you want the Server program to poll only points in visible windows and points that are alarmed, logged, trended or used in any script.

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

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

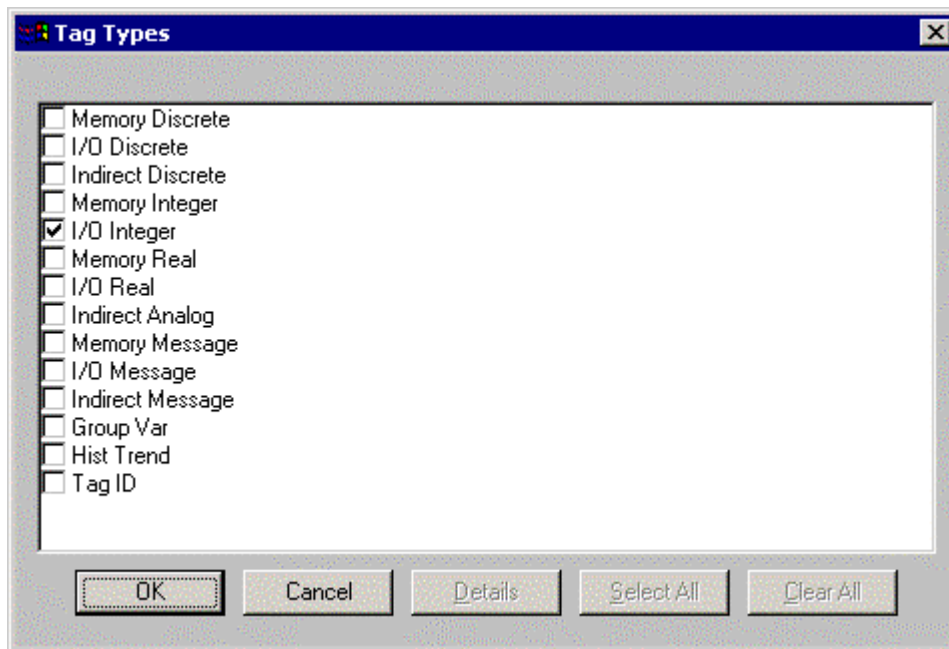
Defining the Tag names

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



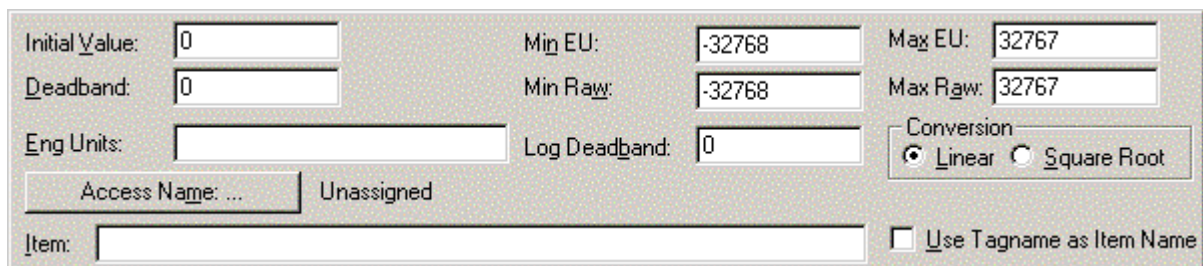
Click on **New** and enter the **Tag Name**. (The tag name defined here is the name **InTouch** will use. The CIMD2IOS Server does not see this name.)

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

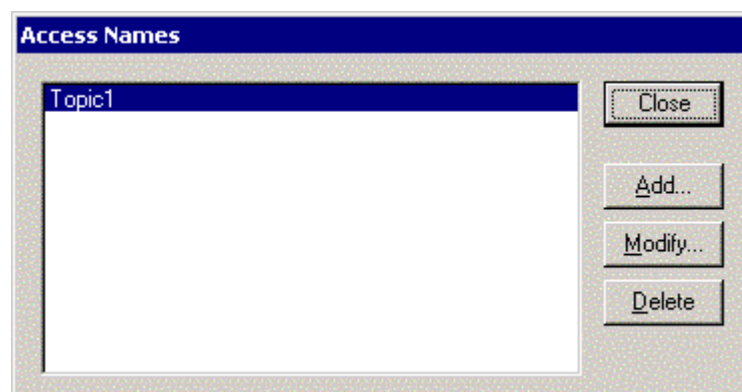


To access CIMD2IOS items, the type must be **I/O Discrete**, **I/O Integer**, **I/O Real** or **I/O Message**. Select the Tag type.

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



Select the Access Name for CIMD2IOS Server by clicking on the **Access Name:...** button. The "Access Names" dialog box will appear:



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

| | | | | | |
|------------------|--------------------------------|---------------|-------------------------------------|---|---|
| Initial Value: | <input type="text" value="0"/> | Min EU: | <input type="text" value="-32768"/> | Max EU: | <input type="text" value="32767"/> |
| Deadband: | <input type="text" value="0"/> | Min Raw: | <input type="text" value="-32768"/> | Max Raw: | <input type="text" value="32767"/> |
| Eng Units: | <input type="text"/> | Log Deadband: | <input type="text" value="0"/> | Conversion <input checked="" type="radio"/> Linear <input type="radio"/> Square Root | |
| Access Name: ... | | Topic1 | | | |
| Item: | <input type="text"/> | | | | <input type="checkbox"/> Use Tagname as Item Name |

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** equal to **Max Raw**.

Enter the CIMD2IOS item name to be associated with this tagname in the **Item:** field in the "Details" box:

| | | | | | |
|------------------|--|---------------|-------------------------------------|---|---|
| Initial Value: | <input type="text" value="0"/> | Min EU: | <input type="text" value="-32768"/> | Max EU: | <input type="text" value="32767"/> |
| Deadband: | <input type="text" value="0"/> | Min Raw: | <input type="text" value="-32768"/> | Max Raw: | <input type="text" value="32767"/> |
| Eng Units: | <input type="text"/> | Log Deadband: | <input type="text" value="0"/> | Conversion <input checked="" type="radio"/> Linear <input type="radio"/> Square Root | |
| Access Name: ... | | Topic1 | | | |
| Item: | <input type="text" value="RL001_002_press01"/> | | | | <input type="checkbox"/> Use Tagname as Item Name |

(Refer to the **Item Names** section 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 CIMD2IOS Server):

=view|DDEStatus! PLC1

or

=view|IOStatus! PLC1

Notes on Using Microsoft Excel

Data from CIMD2IOS topics may be accessed from Excel spreadsheets. To do so, enter a formula like the following into a cell on the spreadsheet.

```
=CIMD2IOS|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:

```
=CIMD2IOS|'Topic1!'RL001_002_press01'  
=CIMD2IOS|'SMSC!'RL003_004_temp02'  
=CIMD2IOS|'Station4!'RECV_MESSAGE'
```

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

Writing Values to CIMD2IOS 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("CIMD2IOS","topicname")  
=POKE(channel,"itemname", Data_Reference)  
=TERMINATE (channel)  
=RETURN()
```

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

```
channel=INITIATE("CIMD2IOS ","topicname")
```

Opens a channel to a specific topic name (defined in the Server) in an application with name CIMD2IOS (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 value included in SMS message to be sent to SMSC) 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 **Topic1** item **RL003_004_setpoint01**:

PokeMacro -Ctrl a

```
=INITIATE("CIMD2IOS","Topic1")
```

```
=POKE(A2,"RL003_004_setpoint01",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 CIMD2IOS 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 CIMD2IOS Server:

```
[CIMD2IOS]
WinIconic=0
WinFullScreen=0
WinTop=291
WinLeft=328
WinWidth=278
WinHeight=137
ProtocolTimer=10
ValidDataTimeout=60000
ConfigurationFile=C:\CIMD2IOS\
```

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):

- | | |
|---------------------|---|
| Show Send | - if checked then all outgoing SMS messages are displayed. |
| Show Receive | - if checked then all incoming SMS messages are 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 sockets, active topics and items. |
| Dump Screen | - if checked then information about currently active topics and built-in I/O items is displayed on CIMD2IOS main window. |
| Clear Screen | - if checked then information on CIMD2IOS main window is cleared. |

All debugging information (except **DumpScreen/Clear Screen** options) 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 can grow very fast.

KLINKMANN AUTOMATION
CIMD2IOS Communication Server
Revision History

Jul 2002 Rev 1.0 First Release