# Table Of Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>1</td>
</tr>
<tr>
<td>Communication Protocols</td>
<td>1</td>
</tr>
<tr>
<td>Accessing Remote Items via the Server</td>
<td>2</td>
</tr>
<tr>
<td>Installing the HIRATA Server</td>
<td>2</td>
</tr>
<tr>
<td>Installing the Server</td>
<td>2</td>
</tr>
<tr>
<td>Installing the I/O Server Infrastructure</td>
<td>4</td>
</tr>
<tr>
<td>Configuring the HIRATA Communication Server</td>
<td>4</td>
</tr>
<tr>
<td>Server Settings Command</td>
<td>5</td>
</tr>
<tr>
<td>Com Port Settings Command</td>
<td>6</td>
</tr>
<tr>
<td>Saving HIRATA Configuration File</td>
<td>7</td>
</tr>
<tr>
<td>Configuration File Location</td>
<td>7</td>
</tr>
<tr>
<td>Configure Topic Command</td>
<td>7</td>
</tr>
<tr>
<td>Item Names</td>
<td>9</td>
</tr>
<tr>
<td>Mode switch items</td>
<td>9</td>
</tr>
<tr>
<td>NOT THROUGH mode items</td>
<td>9</td>
</tr>
<tr>
<td>THROUGH mode items</td>
<td>10</td>
</tr>
<tr>
<td>Using the HIRATA Server with InTouch</td>
<td>13</td>
</tr>
<tr>
<td>Defining the Access Names</td>
<td>13</td>
</tr>
<tr>
<td>Defining the Tag names</td>
<td>14</td>
</tr>
<tr>
<td>Accessing the &quot;STATUS&quot; Item</td>
<td>17</td>
</tr>
<tr>
<td>Monitoring the Status of Communication with InTouch</td>
<td>17</td>
</tr>
<tr>
<td>Notes on Using Microsoft Excel</td>
<td>17</td>
</tr>
<tr>
<td>Reading Values into Excel Spreadsheets</td>
<td>18</td>
</tr>
<tr>
<td>Writing Values to HIRATA Points</td>
<td>18</td>
</tr>
<tr>
<td>Troubleshooting</td>
<td>19</td>
</tr>
<tr>
<td>WIN.INI entries</td>
<td>19</td>
</tr>
<tr>
<td>Troubleshooting menu</td>
<td>19</td>
</tr>
</tbody>
</table>
HIRATA Communication Server

Overview
The HIRATA Communication Server (hereafter referred to as the “HIRATA Server” or “HIRATA” or “Server”) is a Microsoft Windows 32-bit application program that as a communication protocol server and allows other Windows application programs to communicate with the HIRATA Robots (hereafter also referred as the Robots) using the RS-232C serial interface. The HIRATA Communication Server supports HRCS-RIII protocol (THROUGH-MODE items) and HRCS-VI protocol (mode change items and NOT THROUGH-MODE items). The HIRATA Communication Server runs on IBM PC or compatible computer used as a host processor connected to the Robot(s).

Any Microsoft Windows program that is capable of acting as a DDE, FastDDE or SuiteLink Client may use the HIRATA Server.

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

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

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

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

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

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

The Suite Link, FastDDE (Version 3) and DDE support for HIRATA Server is implemented by Wonderware I/O Server Toolkit ver. 7,2,1,6.

Accessing Remote Items via the Server

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

**application name**

The name of the Windows program (server) that will be accessing the data element. In the case of data coming from or going to HIRATA Robots via this Server, the application portion of the address is HIRATA.

**topic name**

Meaningful names are configured in the Server to identify specific devices. These names are then used as the topic name in all conversations to that device. The HIRATA Server considers each Robot to be a separate topic. For example, Robot1.

**item name**

A specific data element within the specified topic. When using the HIRATA Server, an item is an individual point in the Robot. The term "point" is used interchangeably with the term "item" in this User Manual. For more information on item/point names, see the Item Names section later in this manual.

Installing the HIRATA Server

**Installing the Server**

The HIRATA Communication Server installation package can be supplied:

1. As a self-extracting archive 24010xxx.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 HIRATA Server from the self-extracting archive, run the 24010xxx.EXE and proceed as directed by the HIRATA Communication Server Setup program.

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

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

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

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

To **uninstall** the HIRATA Server, start Control Panel, select “Add/Remove Programs” and select the “HIRATA 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 HIRATA Server is developed with Wonderware I/O Server Toolkit (ver 7.2.1.6) and needs the **Wonderware FS2000 Common Components** to be installed on computer where the HIRATA 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 HIRATA 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 HIRATA Server version and these infrastructure files must be install prior to executing the HIRATA Server.

3. The HASP key is needed for full time running of HIRATA Server. The HASP Driver setup is performed during the Server setup. Without HASP Driver installed the HIRATA 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:

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 “I/O 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 HIRATA Communication Server
After the HIRATA Communication Server is initially installed, a small amount of configuration is required. Configuring the Server automatically creates a HIRATA.CFG file, which holds all of the topic definitions entered, as well as the communication port configurations. This file will automatically be placed in the same directory in which HIRATA.EXE is located unless the path where the configuration file will be placed is specified through the /Configure/Server Settings... command.

To perform the required configurations, start the HIRATA program. If the Server starts up as an icon then double-click on the icon to open the server’s window. To access the commands used for 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 a good performance and do not require changing. However, they can be changed to fine-tune the Server for a specific environment.

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

![Server Settings Dialog Box](image)

The following describes each field in this dialog box:

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

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

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

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

Start automatically as Windows NT Service
Enabling this option will cause the HIRATA 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.

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

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

![Communication Port Settings Dialog Box]

The following describes each field in this dialog box:

**Com Port**
This field is used to select the communication port to change or view the settings.

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

**Baud Rate**
The selected Baud Rate must match the settings used in all connected Robots.  
*Note: The default Baud Rate is 9600.*
When all entries have been made, click on **Done** to process the configuration for the communication port. To reset to the default values, click on **Defaults**.

### Saving HIRATA Configuration File

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

![Save Configuration dialog box](image)

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

### Configuration File Location

When the HIRATA Communication 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 in **File Manager** or **Program Manager** and enter the following:

```
HIRATA /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.*

### Configure Topic Command

The user provides each connected Robot with an arbitrary name that is used as the topic Name for all references to this Robot.

To define the Topics (Robots) connected invoke the **/Configure/Topic Definition...** command. The "Topic definition" dialog box will appear:
To modify or examine an existing topic, select the topic name and click on **Modify**. To define a new topic, click on **New**. The "Topic Definition" dialog box will appear:

The following describes each dialog field in this dialog box:

**Topic Name**

Enter the **Topic Name**.

*Note: If using InTouch, the same Topic Name is to be entered in the "Add Access Name" dialog box described in the Using the HIRATA Server with InTouch section.*

**Com Port**

Select the **Com Port** to associate it with the topic. Additional topics may be associated with the same Com Port later.

**Station number**

Enter the **Station number** for this Robot.

*Note: Each Robot must have a unique address from 0 to 999.*

**Update Interval**

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

When all entries have been made, click on **OK** to process the configuration for the Topic. The "Topic Definition" dialog box will appear again.

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

The HIRATA Communication Server supports the following item/point naming system for accessing to the Hirata robot data.

Mode switch items

These items are used to switch Hirata robot to THROUGH mode and to NOT-THROUGH mode.

**MODE** - Integer, Read Only, used to query the HIRATA robot's mode. 1 = NOT-THROUGH mode, 2 = THROUGH mode, 0 = mode unknown.

**MODE_T_S** - Integer, Write Only, used to start the THROUGH mode. When switched to \( n \) the Server starts the THROUGH mode at COM port \( n \) and stops current jobs (\( > \) command sent to the robot). If \( n \) is -1 the COM port number is omitted.

**MODE_T_N** - Integer, Write Only, used to start the THROUGH mode. When switched \( n \) the Server starts the THROUGH mode at COM port \( n \) and does not stop current jobs (\( # \) command sent to the robot). If \( n \) is -1 the COM port number is omitted.

**MODE_NT** - Discrete, Write Only, used to start the NOT-THROUGH mode. When switched to 1 the Server starts the NOT-THROUGH mode (\( MC \) or \( MN \) command sent to the robot). The Server changes this item value back to 0.

NOT THROUGH mode items

Memory items

- **MBa** Memory bit - Discrete
- **MDa** Memory byte - Integer
- **MWa** Memory word - Integer
- **MTa** Memory timer - Real (floating point)

where:
- \( a \) - address: for bits, bytes and words - 0...255, for timers - 0...31;

Input/Output data items

- **IBa** Input bit - Discrete
- **IDA** Input byte - Integer
- **OBa** Output bit - Discrete
- **ODa** Output byte - Integer

where:
- \( a \) - address: for bits - 0...255, for bytes - 0...31;
- Inputs are Read Only, Outputs are Read/Write.

Examples: **MB0, MW125, MT5, IB4, OD28**.
**Robot status**

See Robot status in THROUGH mode items.

**THROUGH mode items**

**Robot position data**

Read is performed by **LD** command. Write is performed by **SD** command. The **p** is a position address, value range from 0 to 1023.

(1) The Read/Write items which values are read by **LD** command and new values written by **SD** command, where other item values are omitted by "_":

- D_Xp, D_Yp, D_Zp, D_Wp - axis data, Reals (floating point);
- D_L_Rp - arm direction, Discrete;
- D_CS p - coordinates system data, Integer;

(2) The DDE items used for setting of all position data (direct values) at once:

- **D_ALL** - all data for **LD** command, Write Only Message (string) item; this item value can be created e.g. by InTouch script or manually. The format of this string should correspond to the format used in **LD** command, the data inside the string can be separated by spaces or commas; this string should contain the following data:

  Address, X data, Y data, Z data, W data, L/R, CS, M, F, S

- **D_ALL_DO** - Integer, Read/Write item used to execute the **LD** command. When this item value is switched from 0 to 1 then **LD** command containing the item **D_ALL** data is sent to the robot. The Server changes this item value back to 0 after command execution and to -1 if command execution failed. If Server does not know the item **D_ALL** value (item is not activated) then **LD** command is not executed and error message is logged to WWLogger.

(3) The items used for setting of position data of other address to specified address or for setting of current position data to specified address:

- **D_SPC** - the specified address, Write Only Integer, value range from 0 to 999.
- **D_ORG** - the original address, Write Only Integer, value range from 0 to 999.

- **D_CPY** - Discrete, Write Only item used to execute the setting of position data of original address (item **D_ORG** value) to specified address (item **D_SPC** value). When this item value is switched from 0 to 1 then **LD** command containing the current values of items **D_ORG** and **D_SPC** is sent to the robot. The Server changes this item value back to 0. If Server does not know the values of items **D_ORG** and **D_SPC** (items are not activated) then **LD** command is not executed and error message is logged to WWLogger.
**D_SET** - Discrete, Write Only item used to execute the setting of current position data to specified address (item D_SPC value). When this item value is switched from 0 to 1 then LD command containing the item D_SPC current value is sent to the robot. The Server changes this item value back to 0. If Server does not know the item D_SPC value (item is not activated) then LD command is not executed and error message is logged to WWLogger.

Examples: D_X0, D_L_R170, D_F1000, D_CPY.

**Robot hold status**

The hold status of the robot is read by LH command and written by GH command. The following Read/Write Discrete items are used: H_X, H_Y, H_Z, H_W.

**Robot current position**

The current position of the robot is read is by LR command and stored in the following Read Only items:

R_X, R_Y, R_Z, R_W - axis data, Reals (floating point);
R_L_R - arm direction, Discrete;
R_CS - coordinates system data, Integer.

**Robot status**

The status of the robot is read is by LS command In THROUGH mode and RZ in NOT-THROUGH mode. The status also is returned by many write commands. The status is stored in the following Read Only items:

S_ERR - error, Discrete, contains 0 if the robot is in the normal status and 1 if robot is in abnormal status;
S_S1, S_S2, S_S3, S_S4 - status data, Integers;
S_E1, S_E2 - error codes, Integers, filled only if robot is in abnormal status;
S_X, S_Y, S_Z, S_W - axis data, Reals (floating point), filled only if robot is in abnormal status.
S_PORT - port, Integer, used in NOT-THROUGH mode to specify COM port;

**Perform A-CAL**

B - Discrete, Write Only item. The A-CAL (automatic origin calibration) is performed (GB command sent to the robot) when item's value is switched from 0 to 1. The Server changes this item value back to 0.
Stop motion

D - Discrete, Write Only item. The robot motion is stopped (GD command sent to the robot) when item’s value is switched from 0 to 1. The Server changes this item value back to 0.

Linear motion

The following Write Only items are used to perform the linear motion (GL) command:

L_ACC - accuracy, Integer;
L_POS - position (memory address), Integer, range of values from 0 to 999;
L_X, L_Y, L_Z, L_W - axis data, Reals (floating point);
L_M, L_F, L_S - M, F and S data, Integers;
L_ADR - Discrete, Write Only item used to execute the linear motion specified by memory address; when this item value is switched from 0 to 1 then GL command containing the current values of items L_ACC and L_POS is sent to the robot. The Server changes this item value back to 0. If Server does not know the values of items L_ACC and L_POS (items are not activated) then GL command is not executed and error message is logged to WWLogger.

L_DAT - Discrete, Write Only item used to execute the linear motion specified by coordinate data and M, F, S parameters. When this item value is switched from 0 to 1 then GL command containing the current values of items L_ACC, L_X, L_Y, L_Z, L_W, L_M, L_F and L_S is sent to the robot. The Server changes this item value back to 0. If Server does not know the values of items L_ACC, L_X, L_Y, L_Z, L_W, L_M, L_F and L_S (some of items are not activated) then GL command is not executed and error message is logged to WWLogger.

Manual motion control

The following Write Only DDE items are used to perform the manual motion (GZ) command:

Z_X, Z_Y, Z_Z, Z_W - axis data, Integers;
Z_T - motion type, Integer;
Z_F - motion speed, Integer;

Z_DO - Discrete, Write Only item used to execute the manual motion. When this item value is switched from 0 to 1 then GZ command containing the current values of items Z_T, Z_F, Z_X, Z_Y, Z_Z and Z_W is sent to the robot. The Server changes this item value back to 0. If Server does not know the values of items Z_T, Z_F, Z_X, Z_Y, Z_Z and Z_W (some of items are not activated) then GZ command is not executed and error message is logged to WWLogger. To keep the robot in motion this command must be re-sent at least in every 0.5 seconds.

Note: Both uppercase and lowercase letters can be used in all item names.
Using the HIRATA Server with InTouch
To access to items on the HIRATA Robots 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.

![Access Names dialog box](image)

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

![Add Access Name dialog box](image)

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

**Topic Name**  
Enter the name defined for the topic in the HIRATA Server to identify the Robot the HIRATA Server will be accessing. The Topic Name is an application-specific sub-group of data elements. In the case of data coming from a HIRATA Server program, the topic name is the exact same name configured for the topic in the HIRATA Server.

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

**Which protocol to use**  
Select the protocol (DDE or Suite Link) that you are using.

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

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

Click **Close** to close the “Access Names” dialog box.

**Defining the Tag names**  
To define the Tag names associated with the new "Access Name", invoke the **/Special/Tagname Dictionary...** command (in **WindowMaker**). The "Tagname Dictionary " dialog box will appear:
Click on **New** and enter the **Tagname**. (The tagname defined here is the name **InTouch** will use. The HIRATA 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 HIRATA 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 HIRATA Server by clicking on the **Access Name**:... button. The "Access Names" dialog box will appear:
Select the appropriate Access Name and click on Close. (If the Access Name has not been defined as previously described, click on Add and define the Access Name now.)

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

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

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

(Refer to the Item Names section upper in this manual 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 tag name. To define additional tagnames click on the New button. To return to the WindowMaker main screen, select Close.
Accessing the "STATUS" Item

For each Topic there is a built-in discrete item (STATUS), which indicates the state of communication with the Robot. This discrete item is set to 0 when communication with the Robot fails and set to 1 when communication is successful.

From InTouch the state of communication with the Robot may be read by defining a DDE Discrete tag name and associating it with the topic configured for the Robot and using STATUS as the Item name:

![Image of InTouch configuration settings with STATUS item selected]

Monitoring the Status of Communication with InTouch

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

The status of communication between the Server and InTouch can be read into Excel by entering the following DDE reference formula in a cell on a spreadsheet (in following examples node1 is the Topic Name configured for HIRATA Server):

=VIEW|DDEStatus!node1
or
=VIEW|IOStatus!node1

Notes on Using Microsoft Excel

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

=HIRATA|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 Names section.
Reading Values into Excel Spreadsheets

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

=HIRATA|node1!MW0
=HIRATA|Robot!LH
=HIRATA|Robot!LS

The status item can be read by entering the following formula in a cell (node1 is a topic name):

=HIRATA|node1!STATUS

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

Writing Values to HIRATA Points

Values may be written to the HIRATA 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("HIRATA","topicname")
=POKE(channel,"itemname", Data_Reference)
=TERMINATE (channel)
=RETURN()

The following describes each of the above POKE macro statements:

channel=INITIATE("HIRATA","topicname")
Opens a channel to a specific topic name (defined in the Server) in an application with name HIRATA (the executable name less the .EXE) and assigns the number of that opened channel to channel.

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

=POKE(channel,"itemname", Data_Reference)
POKEs the value contained in the Data_Reference to the specified item name (actual location in the HIRATA robot) via the channel number returned by the previously executed INITIATE function. Data_Reference is the row/column ID of the cell containing the data value. For "itemname", use some of the valid item names described in the Item (Point) Naming section.

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

=RETURN()
Marks the end of the macro.
The following is an example of Excel macro used to poke value from cell B2 to topic Robot1 item MW0:

PokeMacro -Ctrl a
=INITIATE("HIRATA","Robot1")
=POKE(A2," MW0",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 HIRATA Communication 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. Windows Notepad (*do not use a program that formats text, such as Word or Write unless the file is saved as DOS text*). The following is a typical entry for the HIRATA Communication Server:

```
[HIRATA]
ProtocolTimer=50
ConfigurationFile=C:\HIRATA\WinIconic=0
WinFullScreen=0
WinTop=112
WinLeft=0
WinWidth=200
WinHeight=168
DebugMenu=1
ShowSend=0
ShowReceive=0
ShowErrors=1
```

**Troubleshooting menu**

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

```
[HIRATA]
ProtocolTimer=50
ConfigurationFile=C:\HIRATA\WinIconic=0
WinFullScreen=0
WinTop=112
WinLeft=0
WinWidth=200
WinHeight=168
DebugMenu=1
ShowSend=0
ShowReceive=0
ShowErrors=1
```
Suspend Protocol / Resume Protocol - these choices permit you to turn protocol processing on and off, what means that you can suspend access to the Robot.

Show Send - if checked then all outgoing data is logged in hexadecimal format.

Show Receive - if checked then all incoming data is logged in hexadecimal format.

Show Errors - if checked then all information about errors is logged.

Verbose - if checked then additional debugging information is logged.

Dump - logs all information about opened ports, active topics and data items.

Dump Screen - if checked then information about active messages are displayed on the HIRATA main window. This also can be used to find out how many messages are actually sent to Robot(s).

Log Screen - if checked then HIRATA Server Internal Logger is activated and all debug information is displayed also on HIRATA main window.

Log File - if checked then HIRATA Server Internal Logger is activated and all debug information is saved into HIRATA Internal Logger file. The HIRATA Internal Logger file is named in the format:

D_YYYYMMDD.LOG

where YYYY is a year, MM is a month and DD is a day.

All debug is logged 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.
Revision History

Mar 97    Rev 1.0    First Release.
Aug 97    Rev 1.1    Robot axis’ manual motion control added.
              Robot status monitoring in NOT-THROUGH mode added.
Mar 2002   Rev 1.2   Installation from CD information added.
Mar 2005   Rev 1.3   SuiteLink support added.