MULTILIN ™

# PMCS System Test Simulator

**PMCS System Test Simulator**
**User's Guide**
**GEH-6515**

*GE Power Management Control System 6.13*

***Notice***

The information contained in this document is subject to change without notice. GE makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. GE shall not be liable for errors contained herein or incidental consequential damages in connection with the furnishing, performance, or use of this material.



Document Edition - First 05/96
Second 05/97
Third 05/98
Fourth 03/00
Fifth 02/01, Sixth 01/02, Seventh 01/04, Eigth 09/04

The following are products of General Electric Company:

| | | |
|---|---|---|
| POWER LEADER™ Meter | Power Quality Meter (PQM), | GE Fanuc Series 90/30 PLC |
| POWER LEADER Modbus Monitor | 239 Motor Protection Relay | GE Fanuc Series 90/70 PLC |
| POWER LEADER Electronic Power Meter | 269 Plus Motor Management Relay | GE Fanuc MicroPLC |
| Spectra MicroVersaTrip | 369 Motor Management Relay | |
| Enhanced MicroVersaTrip-C | SR469 Motor Management Relay | |
| Enhanced MicroVersaTrip-D | SR489 Generator Management Relay | |
| MDP Overcurrent Relay | 565 Feeder Management Relay | |
| 735 Feeder Relay | EPM 3710 Electronic Power Meter | |
| Spectra Electronic Control Module | SR745 Transformer Management Relay | EPM 3720 Electronic Power Meter |
| Universal Relay devices | SR750 Feeder Management Relay | EPM 7300 Electronic Power Meter |
| | SR760 Feeder Management Relay | |



US Pat Nos 5,768,148; 5,764,155; 5,862,391

# Back to Main Menu

# Contents

# Introduction

## Welcome

GE's Power Management Control System (PMCS) is a supervisory management and control system (SCADA) for industrial or institutional electrical distribution systems. These systems can be complex and costly to build, involving large amounts of cabling and many power management devices, such as meters and trip units. With such high overhead involved, you can see the importance of being able to model and test such a system prior to physical construction and installation.

The GE PMCS System Test Simulator answers this need. The System Test Simulator is a PC-based tool for developing and configuring the software interface of the PMCS DDE Server, then testing it with client applications. The System Test Simulator makes it possible to model an entire power management system in software, without ever connecting a physical device.

The Simulator plays the role of every power management device attached to the model system, generating appropriate data, events, and responses to the requests of the PMCS DDE Server, which in turn passes the data to the requesting client applications. The client applications requesting data from the PMCS DDE Server are completely unaware that the data they are receiving are not coming from a real power management network.

The figures below Figure 1 and Figure 2 illustrate the differences between a real PMCS network and one running the PMCS System Test Simulator.

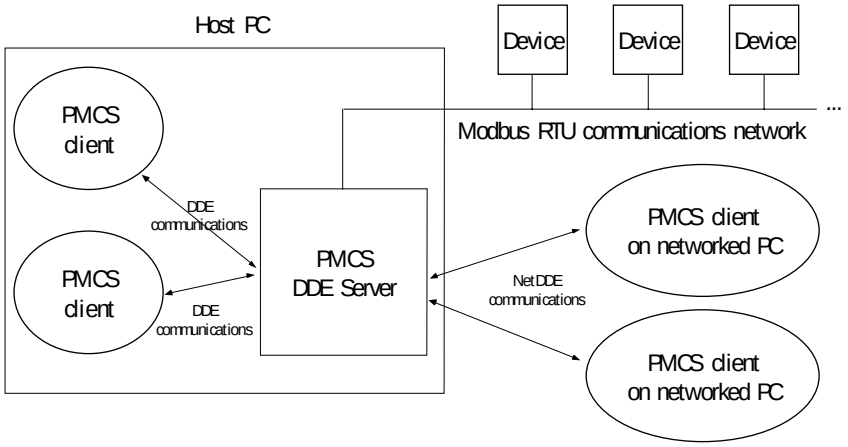



*Figure 1. Example of PMCS system with real devices.*

The first figureFigure 1 shows an example of a real PMCS network. Client applications request device data from the PMCS DDE Server via DDE or NetDDE communications. The PMCS DDE Server collects the data from the power management devices, then replies to the clients.

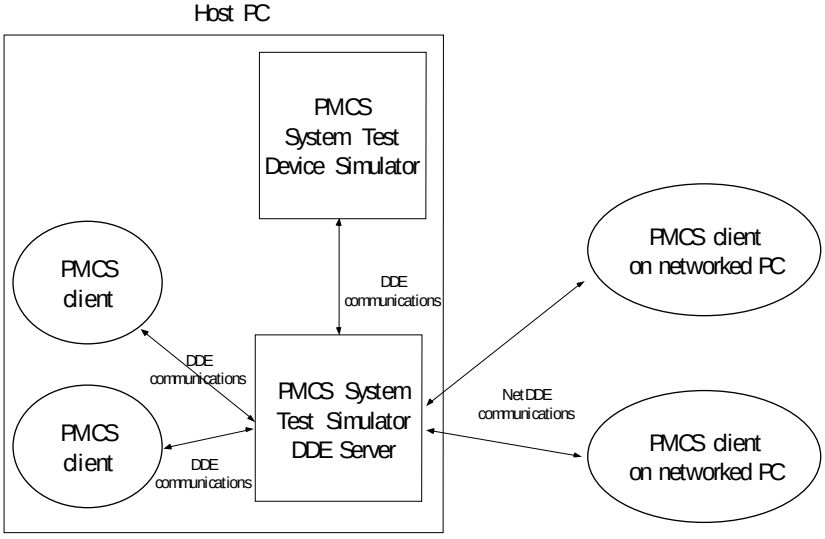



*Figure 2. Example of PMCS System Test Simulator.*

In the second figureIn Figure 2, the client applications request device data from the PMCS DDE Server, just as in a real PMCS network. However, the System Test Simulator uses a special PMCS DDE Server that has been modified to collect its data from the Device Simulator program, rather than from a network of devices. The DDE Server requests the desired data from the Device Simulator, which generates a realistic simulation of data for the appropriate device type. The DDE Server replies to the client applications, which are unaware that the data has not come from real power management devices.

# Basic Concepts

## DDE

DDE is the acronym for Dynamic Data Exchange, a communications protocol that allows independently developed MS Windows programs to share data and instructions with each other.

DDE implements a *client-server* relationship between two concurrently running programs. The *server* application provides data and accepts requests from any other applications interested in its data. The applications requesting the data are called *clients*.

Requests for data can be of two types: one-time requests or permanent data links. With one-time requests, the client program requests a “snapshot” of the desired data from the server application. An example of a one-time request is a program, such as Excel, running a report-generating macro. The macro opens a temporary link to another application, requests specific data, closes the link, then uses the data to generate the report.

Permanent data links are called "*hot links.*" When a client application sets up a hot link to another application, it requests the server application to advise the client whenever a specific parameter changes. Hot links remain active until either the client or server program terminates the link. Hot links are an efficient means of exchanging data because, once the link has been established, no communication occurs until the specified parameter changes.

The DDE protocol specification includes standardized formats for messages to be exchanged between DDE-compliant applications (such as Microsoft Excel).

## NetDDE

NetDDE for Windows NT is an extension to DDE. With NetDDE, client applications do not have to be running on the same PC as the DDE Server; a client application on one PC may request data from a DDE Server operating on another PC. Its capabilities include communication over local-area networks and through serial ports.

Two or more networked PCs running Windows NT are required to run NetDDE. The version of NetDDE that is supplied with Windows NT is the recommended version; other versions of NetDDE are not recommended for use with PMCS.

## PMCS DDE Server

The PMCS Dynamic Data Exchange (DDE) Server is the heart of a POWER LEADER™ Power Management Control System (PMCS), a tool that helps you increase productivity, reduce downtime, and improve power quality by automatically collecting the wealth of data available from devices in your power network. You select data to monitor, configure the PMCS DDE Server to communicate with the selected devices in your system, and the PMCS DDE Server collects the requested data and supplies it to your choice of software applications for analysis and trending.

---

**NOTE**: The DDE Server supplied with the System Test Simulator is a modified version of the PMCS DDE Server. Instead of collecting data from power management devices, it collects data from a special software application called the Device Simulator, which supplies the DDE Server with realistic data to pass to clients.

Apart from collecting its data from the Device Simulator rather than from real devices, the Simulator's DDE Server and the real PMCS DDE Server are functionally identical, even using the same configuration files. After setting up a PMCS network on the System Test Simulator you can immediately use the same configuration simply by launching the real PMCS DDE Server.

Because of this tightly integrated design, you should also keep in mind that any changes you make using the System Test Simulator will apply to the real PMCS DDE Server.

---

The PMCS DDE Server collects metering, status, event, and alarm data from metering, control, and protection devices on the network, or from the PMCS System Test Simulator. The Server then communicates this data to PMCS software clients, such as the third-party HMI tools or the PMCS Event Logger. The data can easily be imported into spreadsheets such as Microsoft Excel for analysis and presentation.

The PMCS DDE Server supports both DDE for sharing data with applications on the same computer and NetDDE for sharing data with other computers in a local-area network (LAN).

## Client Applications

Client applications request specific data from the PMCS DDE Server and then provide calculations, trending, and display of the data on screen and/or printer.

Various client applications are available to serve different needs. Event Logger is an example of client applications that are fully optimized for PMCS.

Any DDE-compliant application can request data from the PMCS DDE Server by initiating a “conversation” with the server and providing the correct information phrased in DDE format—i.e., what data from which device.

Other common DDE-compliant applications used to analyze data from the PMCS DDE Server are Microsoft Excel and Microsoft Access, both providing data-manipulation and analysis tools. However, any DDE-compliant application may retrieve data from thefS DDE Server. For instance, a presentation on the power consumption at an industrial facility can be created using Microsoft PowerPoint, retrieving and graphing power consumption data from the PMCS DDE Server.

# About the System Test Simulator

The PMCS System Test Simulator consists of two primary applications and several support applications. The support applications are discussed later. For now, we are only concerned with the primary applications: the Server and the Device Simulator.

The Server is a modified version of the PMCS DDE Server (Modbus version; Ethernet simulations are not supported by the System Test Simulator), which has been programmed to retrieve data from the Device Simulator rather than from a network of power management devices. The Device Simulator serves as a replacement for the network of power management devices, pretending to be whatever devices we've told it to represent. The Device Simulator responds to the Server's data queries by supplying data that accurately emulates data from real devices.

## Devices Supported by the System Test Simulator

### *Pre-installed GE proprietary devices*

The PMCS Simulator contains pre-installed profile data for the devices supported by PMCS 6.9.

**NOTE**: The Simulator supports some of the functions of the PMCS 6.9 devices. It is not intended to be a compete simulation of all device functions nor a simulation of device interaction or network behavior. For instance, increasing the voltage at one device will not cause the voltage at another device to increase, as it might in a real system. Command coils are not supported for most device types, nor is the generation of waveform data.

### *Generic devices*

The Simulator also supports a customizable generic device type, capable of register-based communications with Modbus RTU.

Usually, a generic device is a third-party power management device requiring custom configuration to work with the PMCS DDE Server. The Simulator can be configured to support generic device types; instructions are provided in *Advanced Topics.*

Several GE devices, such as the family of PLCs, are designed to be so flexible that they also can be considered generic devices, requiring special configuration.

## Who uses the System Test Simulator?

- System integrators and developers can use the Simulator for development and preliminary testing of PMCS supervisory control and data acquisition (SCADA) systems for electrical distribution components, metering, and monitoring.
- Marketing and sales engineers can use the Simulator for demonstration of PMCS systems without connecting to an actual network of devices.

## What is the System Test Simulator used for?

Some of the practical uses of the System Test Simulator are as follows:

- To build a model of a PMCS system, including planned devices and DDE Server Software.
- To test the DDE Server's interface and DDE links between the DDE Server and client applications.
- To test custom-designed Human-Machine Interfaces (HMIs), verifying the links between custom wizards and the DDE Server registers.
- In third-party HMI design tools, to validate the logic of the one-line diagram (electrical representation of the system). In the DDE Simulator, to create pseudo-device names and PLC I/O points to test against the DDE Server.

## Why use the Simulator rather than the real DDE Server?

Using the Simulator provides the following benefits:

- It allows system developers to debug the DDE links in the DDE interface software off-site, prior to full system installation.
- It reduces the time needed for development and on-site testing of the system.
- It eliminates disruptions of the customer operating system or facility.

## Features

- User-configurable power-simulation profiles and topic setup parameters.
- Pre-installed registers for all GE device parameters.
- Pre-installed values for many dynamic value registers (GE devices).
- Consistent device event and trip simulation for dissimilar devices.
- Generic support for any Modbus RTU-compliant device.
- Easy to use, Windows-based graphical user interface with a toolbar and pull-down menus for quick and easy device definition, configuration, and report generation.

## Applicable Documents

- GEH-6502, PMCS Network Architecture Guide
- GEH-6510, PMCS Network and Device Configurator Users' Guide
- GEH-6509, PMCS DDE Server Interface Reference
- GEH-6508, Modbus Concentrator Protocol Reference

# Installation Instructions

For installation instructions, refer to GEH-6514, *Read This Book First*, which contains installation procedures for all POWER LEADER™ system and application software packages.

In order to fully test your system, you may need to install additional software, including the PMCS system, Microsoft Office, and third-party HMI development tools.

## System Requirements

Refer to GEH-6514, *PMCS Read This Book First*, for system requirements to run the PMCS System Test Simulator.

## Directory Structure

Installing the PMCS Simulator software creates the following directory on the hard drive, where X is the drive letter:

```
X:\ge_pmcs\sim\
```

You should know where the files are located on your hard drive so that you do not accidentally move or erase them. In addition, you may want to make changes to these files in the testing process.

# Menu Conventions

Several standard buttons, such as **OK** and **Cancel,** are used in many dialog boxes throughout the software. You should be familiar with the general use and function of such buttons from experience with Windows software. Rather than repeating their definitions in each place where they occur, the following definitions apply globally. In general, only buttons having unique or important functions in a particular dialog box are described in the text.

| | |
|---|---|
|  | Click this button to confirm any changes made in a particular dialog box or to confirm a selection from a list. The button may be dimmed if no changes have been made or nothing is selected. |
|  | Clicking this button cancels a selection, closes the dialog box, and returns to the previous dialog box or level. |
|  | Click this button to access the Help topic for the current dialog box. |
|  | Similar to **OK**. Click on this button to finish selection and close the dialog box. |
|  | Creates a new selection, perhaps a new topic (device), and usually opens a dialog box with various parameters that need to be defined. |
|  | Usually opens a dialog box regarding the current selection, showing various parameters which may be modified. |
|  | Deletes the current selection. |

# Quick Start

## Introduction

In this section, we'll demonstrate the basic functions of the PMCS System Test Simulator software and walk through the most commonly used menus and commands of the program.

This section provides a tutorial rather than a comprehensive reference. Sections 3 and 4, *The Device Simulator* and *DDE Server Menus and Toolbars*, of the User's Guide offer in-depth descriptions of the menus and functions of the Device Simulator and DDE Server respectively.

Before working with the PMCS System Test Simulator, you should plan your network on paper, diagramming the RS-485 communications networks, the devices assigned to each network, and the Modbus address assigned to each device. For information on how to design a PMCS network, refer to GEH-6502, *PMCS Network Architecture Guide*.

## Launching the Program

To start the PMCS System Test Simulator, open the GE PMCS program group in Windows. The Simulator program icon is shown below. Launch the application from the Windows **Start** menu or by double-clicking on the icon.


Launching the System Test Simulator also starts several supporting applications: the Simulator's specially-modified DDE Server and the Graphics Server. All of these pieces work together to provide the simulated devices for client applications.

**Note:** The PMCS System Test Simulator is designed to simulate only Modbus-based PMCS networks. Simulation of Ethernet-based PMCS systems is not supported by the System Test Simulator. However, you *can* use the Simulator with clients set for an Ethernet DDE Server by changing the Server name referenced by the client applications. In your client application, change any references to GE32ENET (the Ethernet DDE Server) to GE32MODB (the Modbus DDE Server/Simulator).

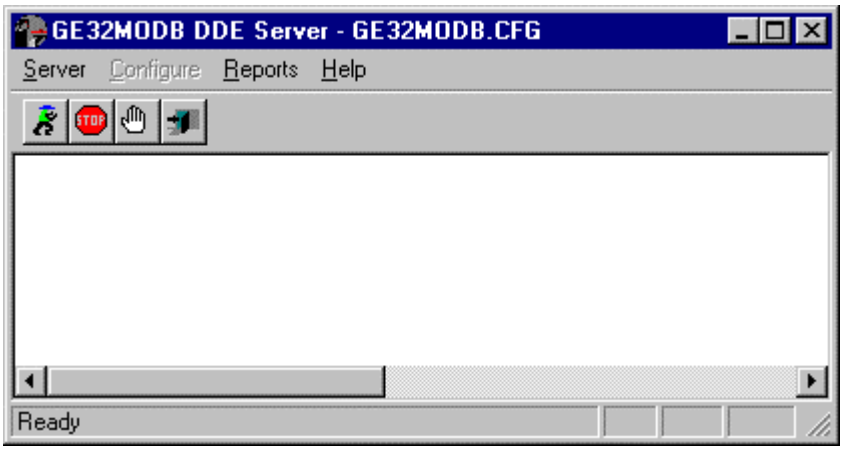
When launched, the Simulator prompts you to load profile data and register map data from a file. If you have previously saved profile data or register map data that you want to continue working with, go ahead and select the data file to load. If you are setting up a new system simulation, click **NO** and proceed.

# Setting up a Simulation

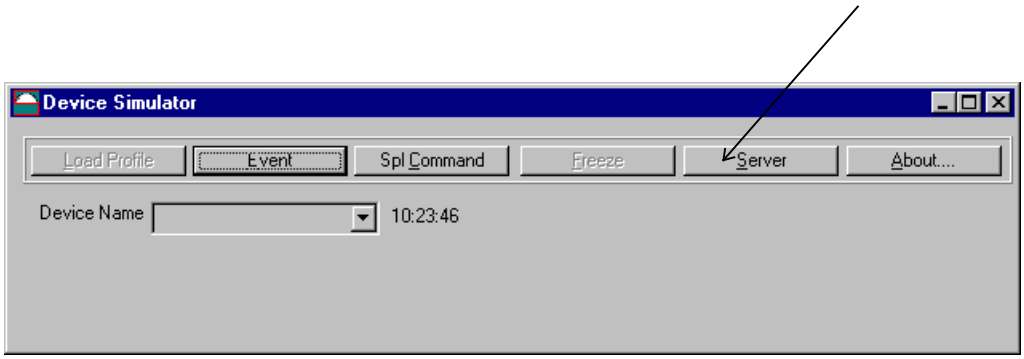
The PMCS Simulator generates data in response to requests from the DDE Server. The Simulator's responses are based on the device type associated with the DDE request; when you request data from an MDP Overcurrent Relay, that's what you get back.

Before the simulation can occur, you must tell it specifically which devices you want to include in your particular system. Therefore, the first step in setting up a simulation is to configure the devices you want to simulate. We'll walk through setting up a device to demonstrate the procedure.

To begin this exercise, you should be looking at the main window of the Simulator's DDE Server:

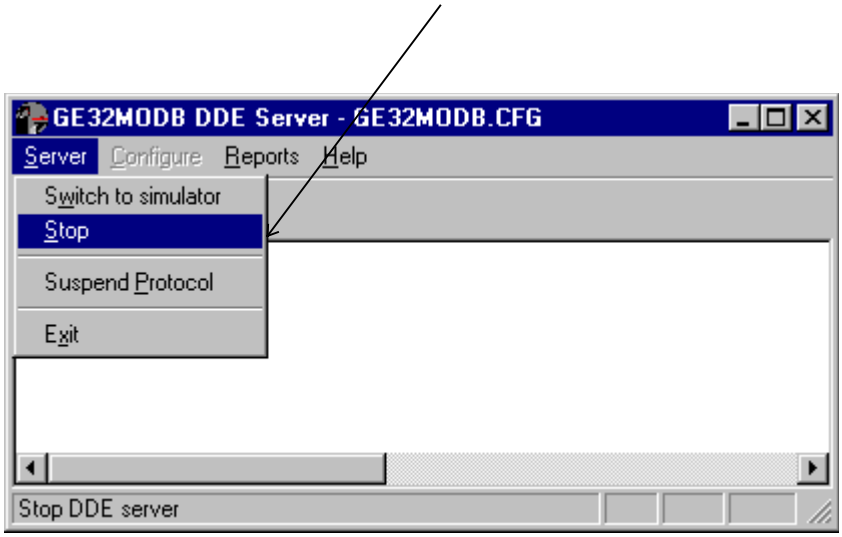


If you're looking at the **Device Simulator** window (shown below), click the **Server** button to switch to the Server window.

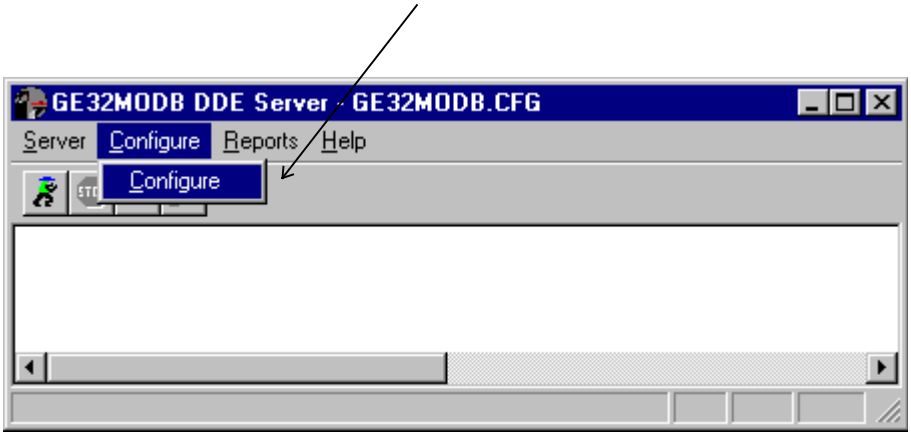



We'll configure a single device and explain how to access its data from a client application. Once you've learned the basics of configuring a device, you'll be ready to set up simulations of more involved networks. Later, we'll show you how to custom-configure generic devices and perform advanced Server configuration.

From the Server window's menu bar, pull down the **Server** menu and select **Stop**, unless it is already grayed out. The Server must be Stopped before any device configuration can be performed. (Clicking the stop sign button on the toolbar performs the same action as selecting **Stop** from the **Server** menu.)

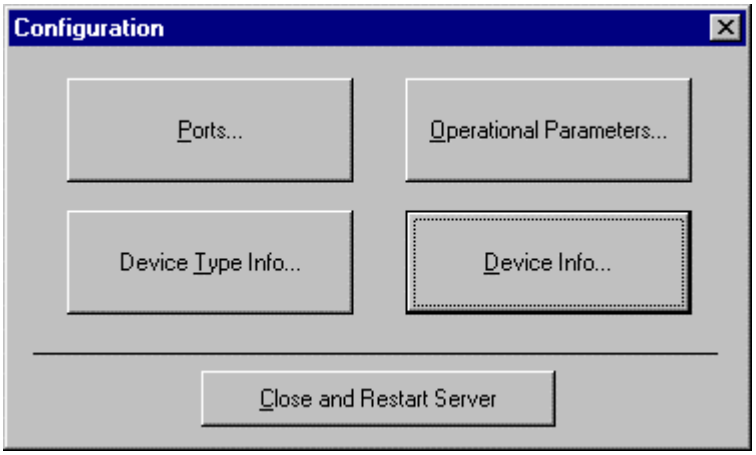


Pull down the **Configure** menu and select **Configure** (the only option on the menu).



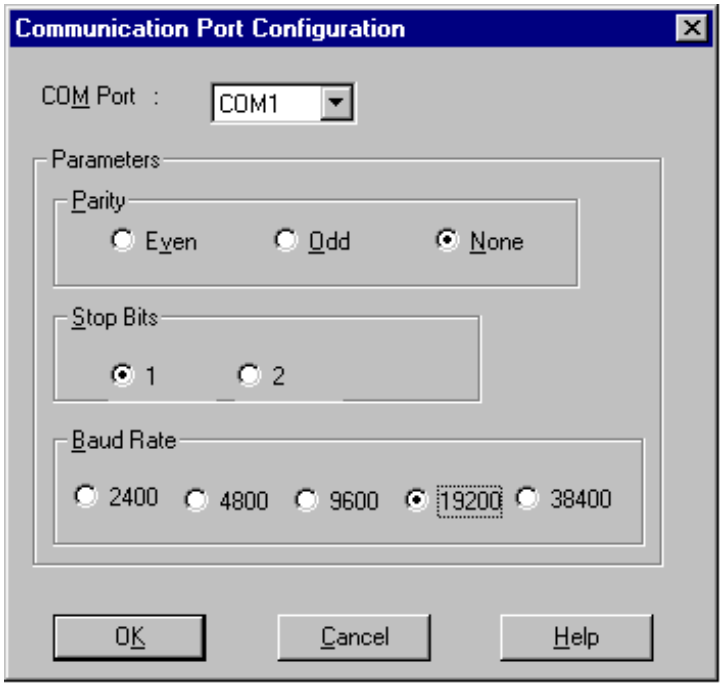

This will bring up the **Configuration** dialog box, offering four options:

- Ports…
- Operational Parameters…
- Device Type Info…
- Device Info…




Ignore the **Operational Parameters…** and **Device Type Info…** buttons for the time being–these are advanced options that will be discussed later. We'll first configure a communications port, then configure a device assigned to that port.

To configure a communications port, click the **Ports** button. and the **Communication Port Configuration** dialog box is displayed:




Select the communication port to configure from the **COM Port** pull-down list at the top of the dialog box.

Select the appropriate radio button for each of the communication settings: **Parity**, **Stop Bits**, **Flow Control**, and **Baud Rate**. The default settings are shown. Typically, only the baud rate need be changed to match the baud rate of the devices connected to the comm port. The rest of the communication settings are standard. Refer to the user manuals of the individual devices to be sure the communication settings match.

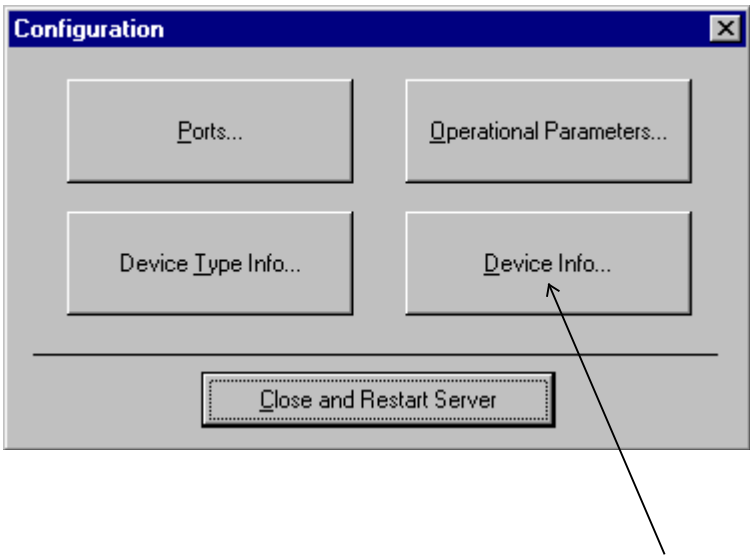
Click on **OK** to return to the **Configure** dialog box—our first com port has been successfully configured.

---

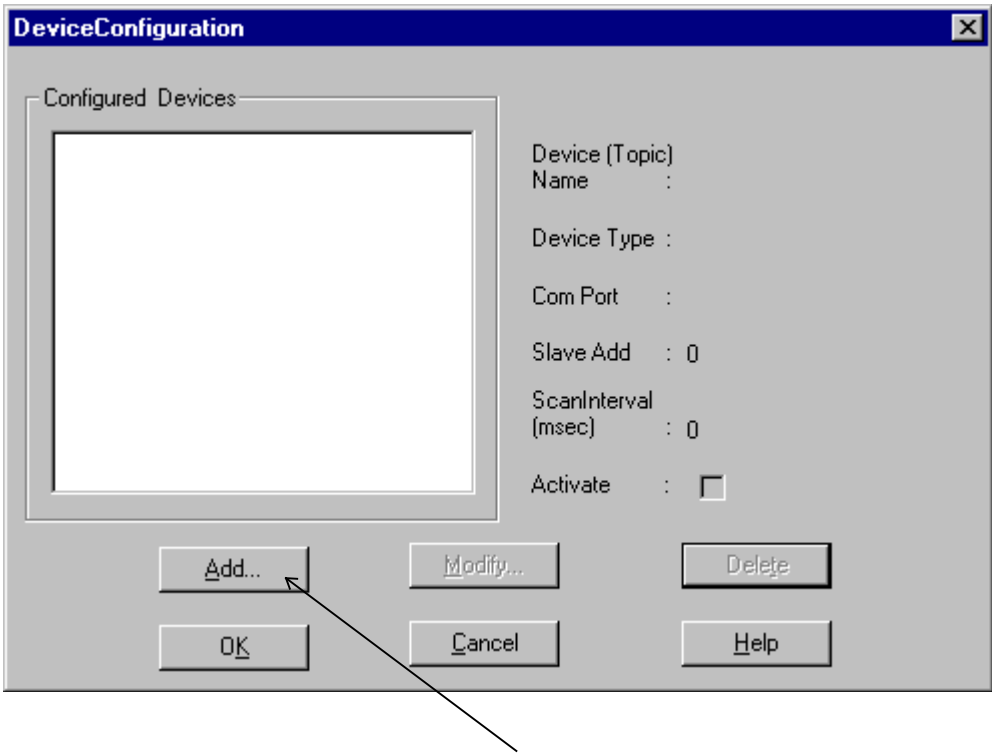
*Hint*: You don't need to leave the Communication Port Configuration dialog box to configure multiple ports. Select a port from the pull-down list, make your changes, then you can select another port from the pull-down list and configure it as well. Configure as many ports as you need to, then click OK to save your changes and return to the Configure dialog box.

---

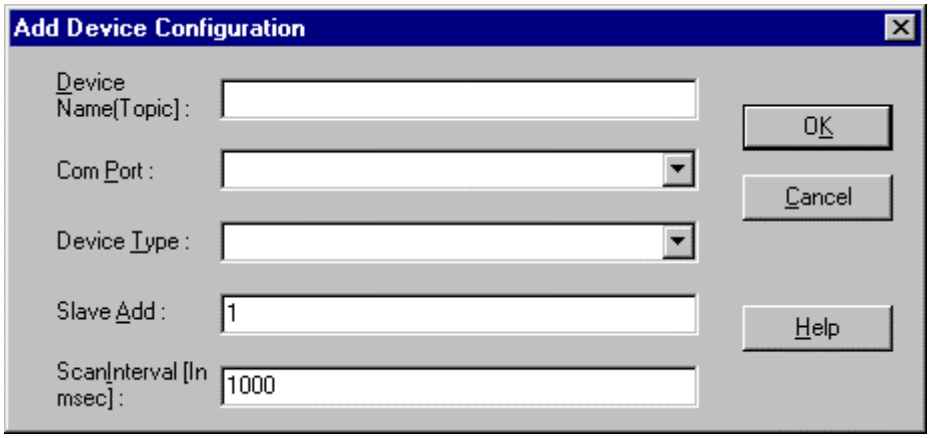
Next, we'll configure a sample device.




Click the **Device Info** button. This will bring up the **Device Configuration** dialog box, displaying a list of configured devices and options to **Add**, **Modify** or **Delete** devices.




We want to add a device, so click the **Add** button. The **Add Device Configuration** dialog box appears:




In the **Device Name (Topic)** field, enter the name by which you'll reference this device from a client application. The device name should be unique, have fewer than 20 characters, and contain no special characters (the program won't allow you to enter any, so this part is easy).

Select a device type from the **Device Type** pull-down list. For this exercise, select an MVT (MicroVersaTrip trip unit). Selecting the type of device, such as Spectra ECM or POWER LEADER EPM, tells the DDE Server what kind of register map to expect and tells the Device Simulator what kind of register map to use in simulating this device. (Details of PMCS device register maps can be found in GEH-6508, the *PMCS DDE Server Interface Reference*.)

Select the com port this device is attached to.

Enter a valid slave address; this is the Modbus address of the device. You should have this information on the network diagram you drew earlier.

The address selected must adhere to the following rules:

- For Modbus Concentrators, the slave address must be between 1 – 32.
- commnet devices must have slave addresses in the range 33 – 247
- Modbus-based devices may have any address from 1 – 247

If you're not sure whether a device is commnet- or Modbus-based, or you have questions regarding commnet and Modbus, refer to GEH-6502, *PMCS Network Architecture Guide*.

The **Scan Interval** field should be left set at 1000. We'll discuss **Scan Interval** later on.

Last, click the **OK** button to accept your entries, close the dialog box, and return to the **Device Configuration** dialog box.

Make sure the **Activate** checkbox is selected, or the Server will not recognize the device. This feature is a useful way to quickly disable individual devices.

Click **Close and Restart Server** to exit **Configuration** and begin using the device. You'll return to the Device Simulator window and view the default data being generated to simulate the device you just configured.

The Server now recognizes the device, so when a client application requests data from the device, the Server will retrieve the data from the Device Simulator and respond to the client's query.

That's the procedure for configuring any supported GE device. Configuration procedures for "generic" devices such as non-supported third-party devices, and highly customizable GE Fanuc Programmable Logic Controllers, are covered in Section 6, *Advanced Topics*.

# Setting up DDE Requests from Client Applications

To obtain data from a device on the System Test Simulator, a client opens a channel to the DDE Server by specifying these three names:

<application name> <topic name> <item name>

<application name> – the name of the Server program.

<topic name> – the name given to the field device, such as EPM1 for a device of type EPM.

<item name> – The actual address of the field parameter to be monitored—a specific valid register address or group under topic EPM1, such as R34001, or mnemonic, such as Amps_A.

For register and item names and examples of DDE protocol, refer to GEH-6509, *PMCS DDE Server Interface Reference.* An example is provided which demonstrates the use of a spreadsheet to link to a DDE item in the Server or Simulator.

Refer to documentation of your client program for details on defining a DDE item in the client.

# The Device Simulator

## Introduction

With the overview of device setup behind us, we'll now take a closer look at the Device Simulator.

As we mentioned earlier, the System Test Simulator has two major components, the DDE Server and the Device Simulator. The DDE Server is just a special version of the real PMCS DDE Server modified to retrieve data for any configured device from another piece of software, the Device Simulator. The Device Simulator pretends to be any device the DDE Server requires, supplying realistic data simulations for the parameters of each device.

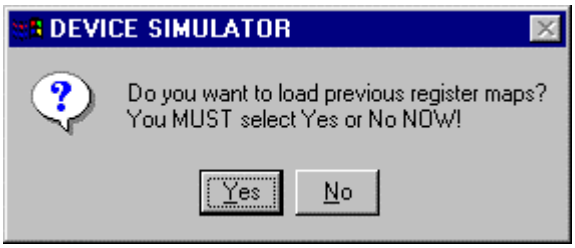
## Start-up

When the DDE Simulator is launched, the Device Simulator is launched as well. Several start-up screens, shown below, prompt you to either load previous simulation data or start from scratch (by not loading any previous data). Respond **No** to the following two prompts if you wish to start a new simulation from scratch.

First you'll be prompted to load previously saved profile data. Profile data is any special customization to your device's power profile, such as adding noise or setting arbitrary currents or voltages.


Select **Yes** if you want to load saved profile data. Select **No** if you are running the Simulator for the first time or do not have any previously saved simulator profile data files available.

Next you're prompted to load previous register maps. Like saving profile data, you may also save fixed values you may have changed in the devices' register maps.

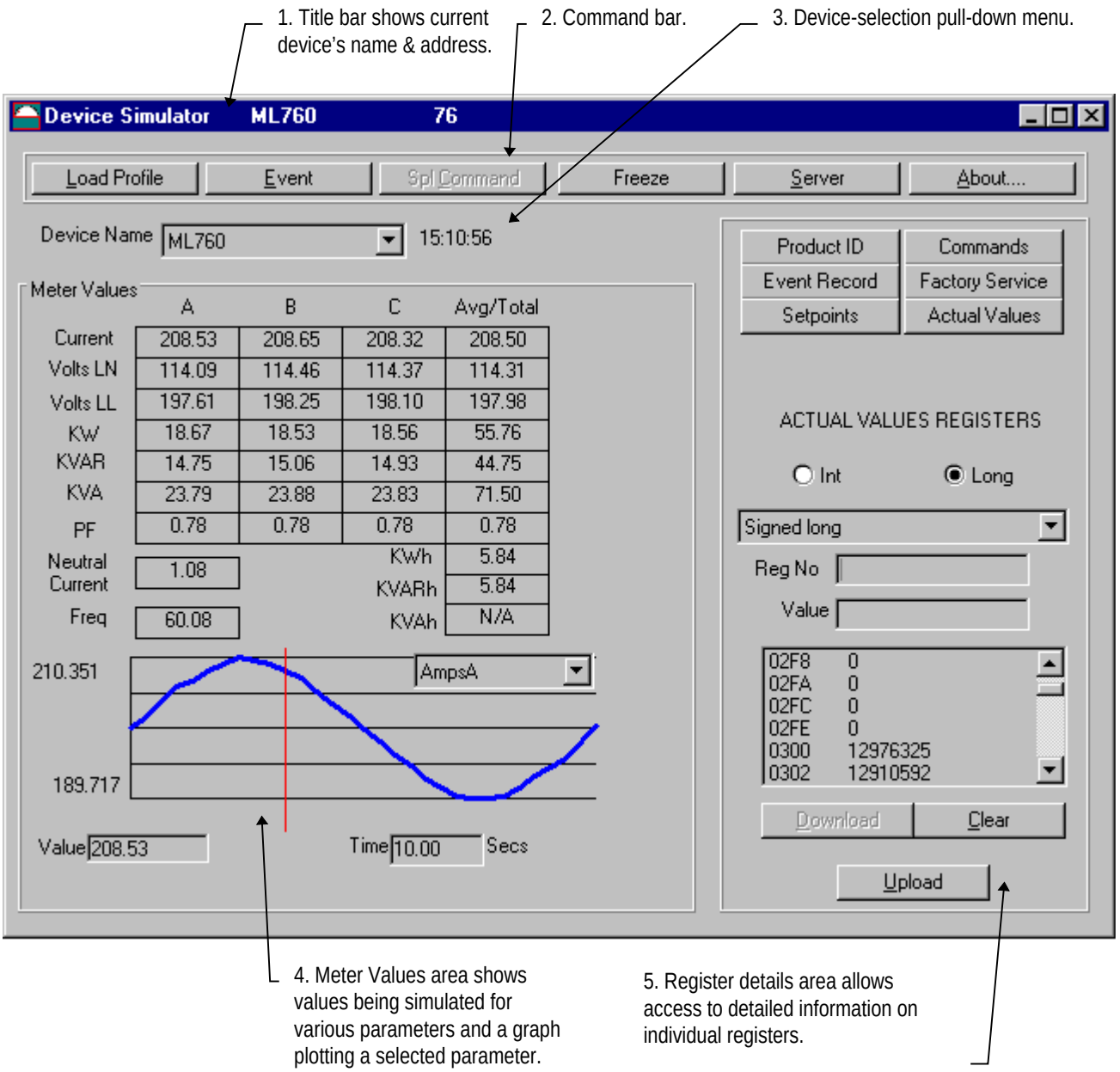



Click **Yes** if you want to initialize registers with their previously stored values, or **No** to use default register values or if you are running the Simulator for the first time.

Register maps and profiles are saved separately, allowing for more-flexible simulations. You might set up a group of devices with particular register maps and then see how the system performs under a variety of different power profiles.

When you have navigated the opening dialog boxes, the **Device Simulator** main screen appears. When you select a device for display, a screen full of simulated data is displayed, similar to the one below.

There are five important parts of the **Device Simulator** screen, indicated below and detailed in the following pages.



# Simulator Screen Controls

## Title Bar

The title bar at the top of the **Device Simulator** screen displays the name of the currently selected device and its Modbus address.

## Command Bar

The buttons at the top of the Simulator screen offer access to the following commands.

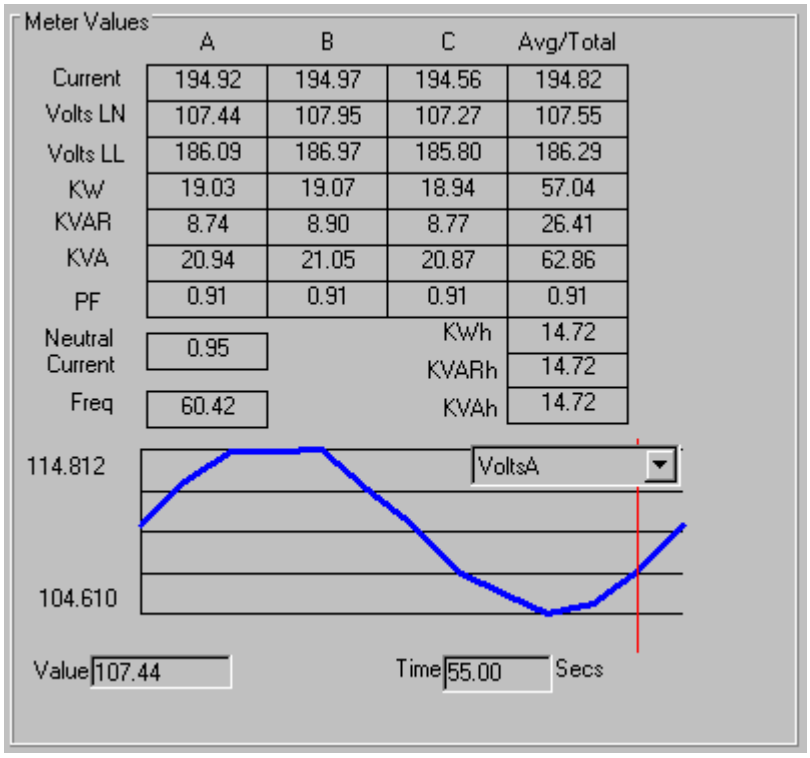
| | |
|---|---|
| Load Profile | Opens the **Configure Load Profile** dialog box, allowing you to customize details of the selected device's load profile. See *Defining Device Profiles* for details. |
| Event | Opens the **Event Generation** dialog box to simulate events for the selected device. See *Defining Events and Special Commands* for details. |
| Spl Command | Opens the **Special Command** dialog box, which allows you to simulate various device functions. See *Defining Events and Special Commands* for details. |
| Freeze | The **Freeze** button freezes data simulation for the currently selected device. Clicking this button toggles the button between **Freeze** and **UnFreeze**. In **Freeze** state, updating stops for all simulated parameters. The data being simulated at the moment the Freeze button is clicked will continue to be sent to the DDE Server until the **Unfreeze** button is clicked. |
| Server | The **Server** button allows you to switch to the DDE Server window quickly. |
| About.... | Displays version information about the Device Simulator. |

## Device Selector


The **Device Name** pull-down list allows you to select the device for which you want to see simulated data. A clock displaying the current system time is provided for your convenience to the right of the pull-down list.

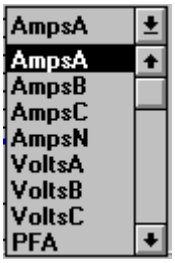
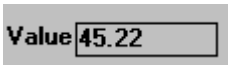
## Meter Values

The **Meter Values** area displays the data being generated by the Device Simulator for the selected device.




These data are generated using the values configured in the **Configure Load Profile** dialog. The parameters displayed vary depending on the selected device type; different device types have different capabilities.

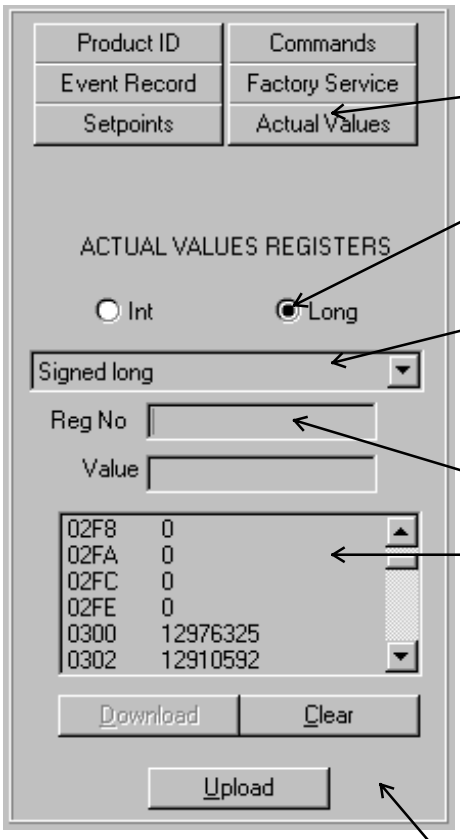
The graph area shows a plot of a selected parameter. The simulation cycle repeats continuously over the **Profile Length** selected in the **Configure Load Profile** dialog. The moving line shows the progress through the simulation while the **Time** counter shows the elapsed time in five-second increments. The current value is displayed in the **Value** box.

| | |
|---|---|
|  | Use the pull-down list box to select the parameter to be plotted on the graph. |
|  | The current value of the variable being plotted is shown in the **Value** box. |
|  | The elapsed profile time of the parameter being plotted on the graph is shown in the **Time** box. The counter ticks in one-second intervals (this increment is not user adjustable). |

## Register Details

The right side of the **Device Simulator** screen shows information on the registers of the selected device. The panel at the right side of the window offers push-buttons for each register type present in the selected device. The buttons available vary depending on the currently selected device type. You can click any of the buttons to see the related register values in the list box.

Using the controls in this area, you can directly view and change the contents of any available registers for the current device type.




1. Select a group of registers to view by clicking one of these buttons.
2. Select the display format for the register values.
3. If you selected the Int or Long radio button, you'll have access to this pull-down list to further select the display mode.
4. Enter a register number in the Reg No. field or select a register in the scrollable list box. The register number and value are shown in the fields above the list box. To change a register, enter its new value in the Value field, and press the Enter key. For most devices, the register numbers are in decimal format. The Multilin devices are an exception, and show register numbers in hexidecimal format.
5. To clear a value from a register and restore the default value, click CLEAR. To update the register map with any changes you've made, click Download. (Download does not write changes to a file; see below.) To refresh the screen with any new data changes coming from DDE clients, click Upload.

The scrollable list box shows the values present in the selected register group. Note that the scrollable list box does not immediately show new values from DDE clients; to update the display, click on the upload button.

If you make a change to a register group, then click on a button to display another group of registers, you'll be prompted to save changes to the current register group, as shown in the following dialog box. Clicking **Yes** in this dialog box writes any changes to a file.


Clicking the **Download** button updates the register map with any changes we have made. The new data is available for client access when the **Download** button is clicked. The display of registers at the Device Simulator is static and is not updated unless you have switched to a different register group and back or click on the **Upload** button.

Before exiting the DDE Simulator, you will have another opportunity to save any changes you've made. When exiting, the software prompts you to save any profile data to a disk file.

For example, suppose we want to change a Setpoint register in the EMVT-D device type. Select EMVT-D from the **Device Type** pull-down menu, then click the **Setpoint** button in the **Register Details** area. Enter the register number in the **Reg No**. field, or scroll down the register list and click on the desired register. The current value of this register will be displayed in the **Value** field. Highlight the **Value** field and type the new entry for this register. Press the Enter key, or click on the **Download** button. The new value is entered into the register, and is available for access from clients. In order to see any changes to the data registers (such as a new data value poked to the simulator from a DDE client); click on the **Upload** button on the DDE Simulator screen.

---

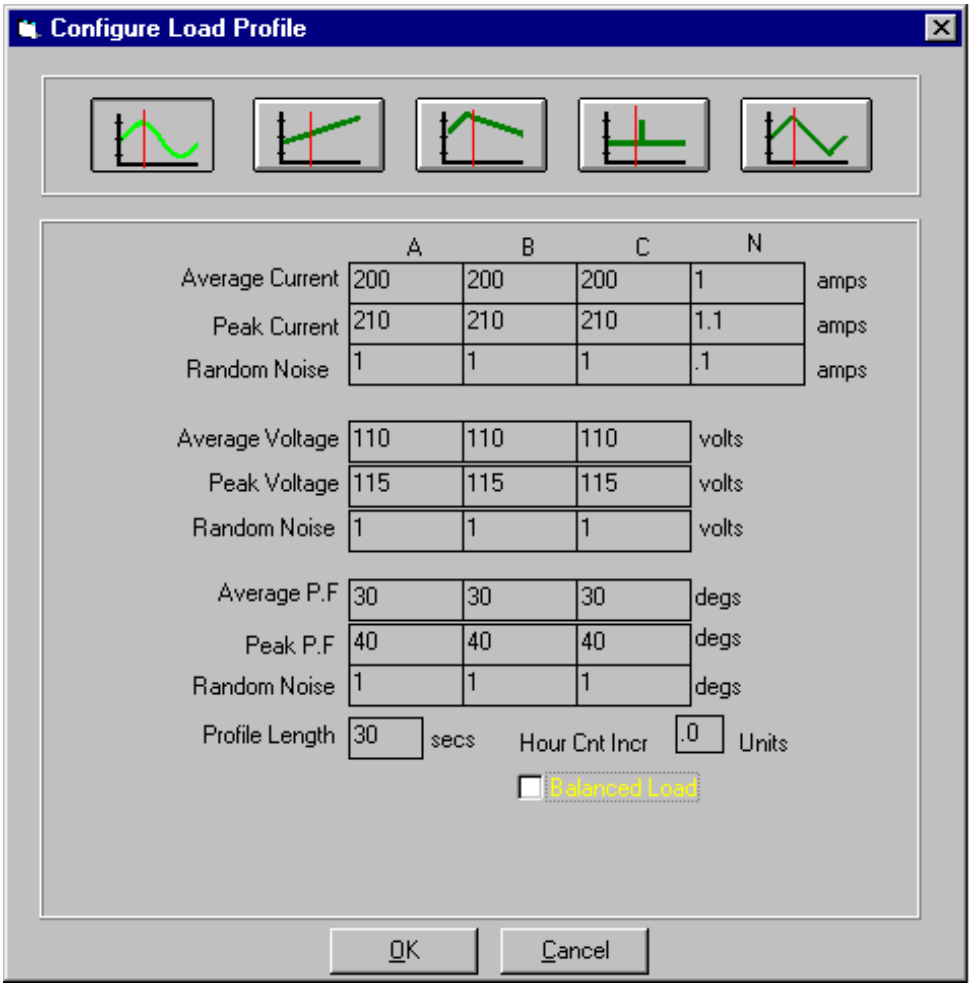
**NOTE**: Since both the DDE Server and the DDE Simulator point to the same configuration files, setting up or editing either program causes the same configuration to take effect in the other program as well.
Be careful when running both the Simulator and the real PMCS DDE Server simultaneously; be sure that the .ini file for the current application references the appropriate configuration files. To use one set of configuration files with the Simulator and a different set with the PMCS DDE Server, be sure to change the .ini file prior to launching the application.

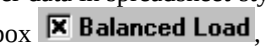
---

# Defining Device Profiles

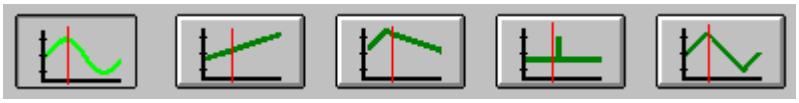
The Device Simulator allows you to customize the power load profile for an individual device.

Clicking the **Load Profile** button displays the **Configure Load Profile** dialog box for the current device:




Click in any of the cells and enter the desired power data in spreadsheet-style. To speed data entry, click the **Balanced Load** checkbox  , which automatically balances the load on all phases. All phases will have similar current, voltage, and PF values; the value you enter in one of the cells will be propagated to all the cells. You can enter values for **Average** and **Peak Current**, **Average** and **Peak Voltage**, **Average** and **Peak Power Factor** and also **Random Noise**, a value for simulating noise on any of the phases.

You can specify different profile trends by selecting the desired profile from the buttons at the top of the dialog box:



Entering the **Profile Length** sets the time duration that the Simulator will take to run the selected profile. The profile is cycled continuously.

The Simulator uses a counter to increment energy hours once every second. **Hour Cnt Incr Units** allows you to customize the increment for energy hour data – for instance, energy hours can accumulate by one unit (1, 2, 3, 4, ...) or two units (2, 4, 6, 8, ...) every second.

When you have finished configuring the load profile, click **OK** to save the changes and close the **Configure Load Profile** window. **Cancel** closes the window without saving changes.

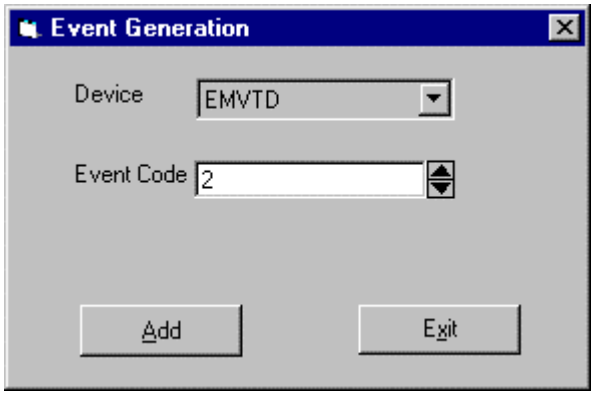
# Defining Events and Special Commands

## Events

Clicking the **Events** button opens the **Event Generation** dialog box. This dialog box allows you to simulate events for a particular device.

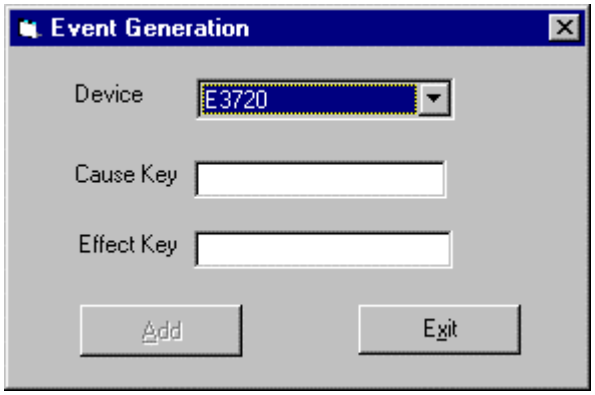
Event codes for some devices may be found in GEH-6508, *Modbus Protocol Guide*, or in the specific device's user manual or Modbus protocol guide.

Select a device from the pull-down list in the Event Generation dialog box. Select an Event Code from the pull-down list or type in the event code number. Event codes are entered as decimal (not hexadecimal) numbers. Refer to the device documentation for lists of available event codes.




Click the **Add** button to generate the selected event for the selected device. Click **Exit** to close the **Event Generation** dialog box.

**NOTE**: If you select an **EPM 3720** from the pull-down list, the dialog box is slightly different, as the 3720 requires that you specify cause and effect keys; refer to the EPM 3720 documentation for Cause and Effect Keys.



## Special Commands

The **Spl Command** button enables the simulation of various device functions, including trips, sending coil commands, and resetting energy.

Similar to the **Event Generation** dialog box, the **Special Commands** dialog box prompts you to select a Device from the upper pull-down list and a Command from the lower pull-down list:


The **Device** list contains the list of all configured devices (topics) that can execute special commands. Select a device for which you want to simulate a special command. The **Command** pull-down list shows the special commands available for the selected device. Select a command, and click **OK** to execute it. Click **Exit** to leave the **Special Commands** dialog box.

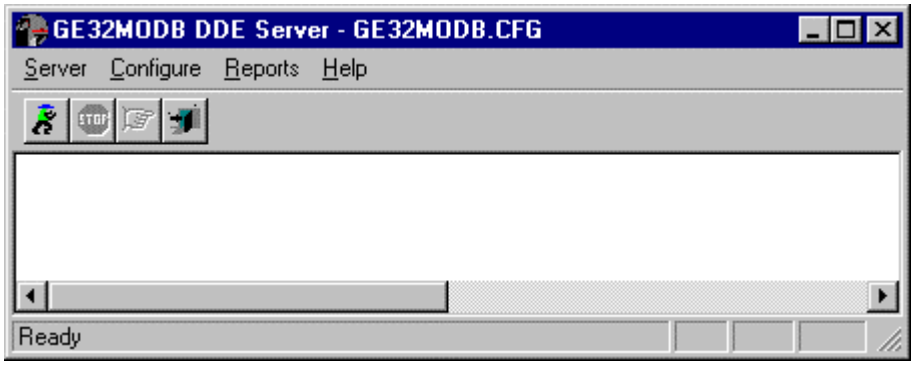
**NOTE**: Special commands are not supported for most device types.

# DDE Server Menus & Toolbars

In this section, we'll examine each of the Server's menus and toolbar buttons in detail, describing its functions and options. We'll assume that some buttons (such as **OK** and **Cancel**) are self-evident and that you can interpret their functions from general experience with the Windows interface.

## Screen Navigation

With the software up and running, you should find yourself looking at the PMCS System Test Simulator Server main window, a modified version of the PMCS DDE Server main window. If you are familiar with the PMCS DDE Server, you will find the Simulator's DDE Server is nearly identical.




The program offers two sets of navigational and operational controls, the menu bar and the tool bar:

- **Menu bar** -- Located directly beneath the window's title bar. You can access these pull-down items either with the mouse or the keyboard (Alt+underlined letter).
- **Toolbar** -- Located beneath the menu bar. These control buttons provide point-and-click access to commonly used controls.

These controls are discussed in more detail in the following sections.

Where applicable, Help icons appear below the cursor and in the status line (box at the bottom of the main window) that prompt an action or describe an object when the cursor "lingers" over an object.

## Menu Bar

The PMCS DDE Simulator main window contains four pull-down menus:

- Server
- Configure
- Reports
- Help

…as well as the standard Windows system menu. We'll take a closer look at these menus shortly.

## Toolbar

The main window toolbar contains four icons, shown below in Table 1. Click on an icon to perform the action described.

| Icon | Function | Description |
|---|---|---|
|  | Server Run | Starts the DDE Server |
|  | Stop DDE Server | Stops DDE Server |
|  | Suspend/Resume | Suspends or resumes the DDE Server's activities |
|  | Exit DDE Server | Exits DDE Server |

*Table 1. Toolbar icons.*

## Minimize Icon

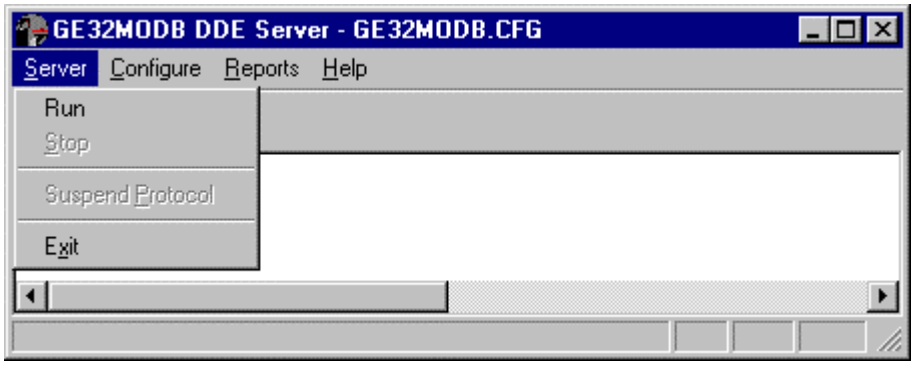
The following icon appears when the DDE Simulator Server is minimized.

# System Menu

The system pull-down menu is shown below. The menu is standard to Windows; refer to the Windows documentation if you have any questions regarding these functions.


# Server Menu

The **Server** pull-down menu is shown below. Descriptions of the Server options follow.




## Switch to Simulator / Run

The first item on the **Server** menu is either **Switch to Simulator** or **Run**, depending on the current mode.

If the item reads **Switch to Simulator**, you know that you are currently in **Run** mode—the DDE Server is able to accept DDE requests from clients and respond to them.

If the item is **Run**, the Server program is in **Stop** mode — no requests are being accepted or answered. You must be in **Stop** mode to make any configuration changes such as adding new devices or modifying existing devices. To enter **Stop** mode, close all DDE client programs to disconnect any active DDE links, then select **Stop**.

**Switch to Simulator** jumps from the DDE Server to the Device Simulator.

**Run** places the DDE Server in **Run** mode, ready to answer DDE requests from clients. The Server waits for DDE clients to become active. Once a client is actively making requests for data, the PMCS DDE Server retrieves the appropriate data from the Device Simulator and answers the requests.

Note: The **Stop** command temporarily closes the Device Simulator window, and the **Run** command will restart the Device Simulator. It may take several seconds for the Device Simulator to start up after issuing the **Run** command.

You must configure the Server with at least one device before it can run a simulation or a client requests data from it. Configuration is discussed in Section Two, *Quick Start*, and also later in this section.

All configuration information is saved to disk and is re-loaded each time the Server is run.

## Stop

**Stop** sets the PMCS DDE Server off-line, preventing it from requesting any data from the Device Simulator. Before you can stop the Server, however, DDE links with client applications must be broken from any open client applications. This may require closing the client applications.

Once the PMCS DDE Server is off-line, you can make configuration changes.

While the Server is Stopped, a DDE request from a client will automatically return the Server to Run mode, unless configuration changes are underway. Selecting **Configure** from the **Configure** menu disables this "auto-wakeup" feature.

## Suspend Protocol/Resume Protocol

This menu item is enabled once the Server starts running. **Suspend Protocol** and **Resume Protocol** are mutually exclusive options. One or the other will be displayed on the Server menu, depending on the current state of the program.

**Suspend Protocol** temporarily halts the collection of data by the DDE Server without requiring that the DDE links with clients be broken, whereas **Stop** requires that the links be broken first. When you select **Suspend**, any DDE links remain intact; they merely become idle until you select **Resume**.

When you select **Suspend Protocol**, the Server stops data acquisition and the menu item changes to **Resume Protocol.**

When you select **Resume Protocol**, the suspended DDE links become active again, and the Server resumes answering requests for data.

This is useful if you wish to pause operation of the Server to view data traffic in WWLogger . It has no effect on the Device Simulator; the Simulator continues to generate appropriate simulated data.

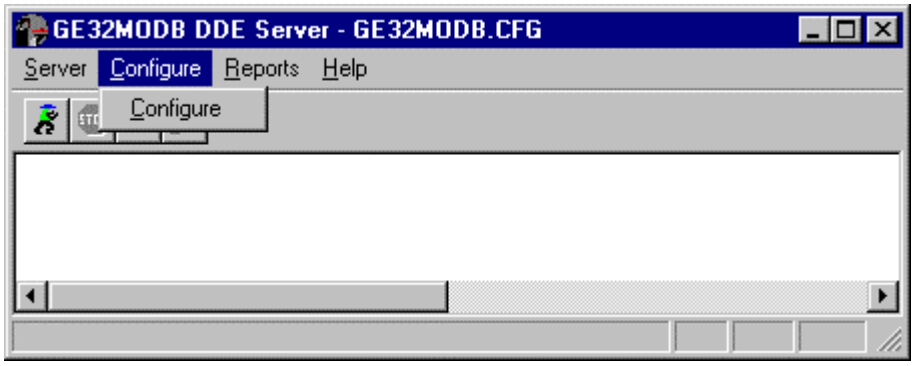
Note that Suspending the DDE Server does not permit you to make any configuration changes; the Server must be Stopped for configuration changes to be made.

## Exit

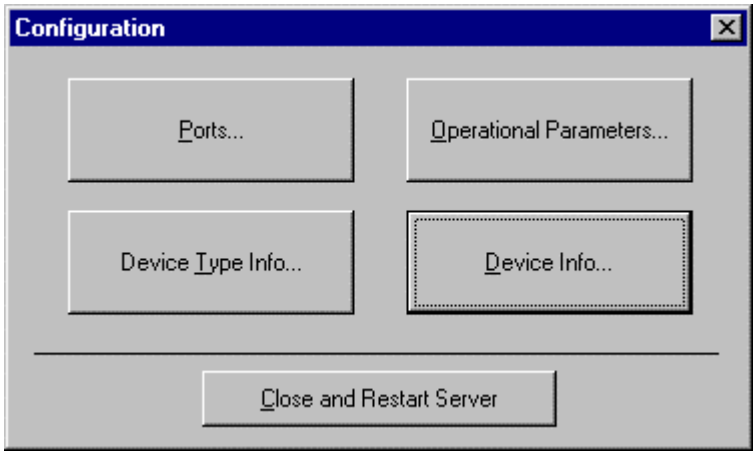
**Exit** is the standard Windows function for leaving the program. A dialog box prompts you for confirmation that you really want to exit the program. This will also exit the Simulator and you will be prompted to save the profile data and register maps.

# Configure Menu

The **Configure** pull-down menu is shown below. Descriptions of the Configure options follow.




The only option available from the **Configure** menu is **Configure**. Selecting this option brings up the **Configuration** dialog box.

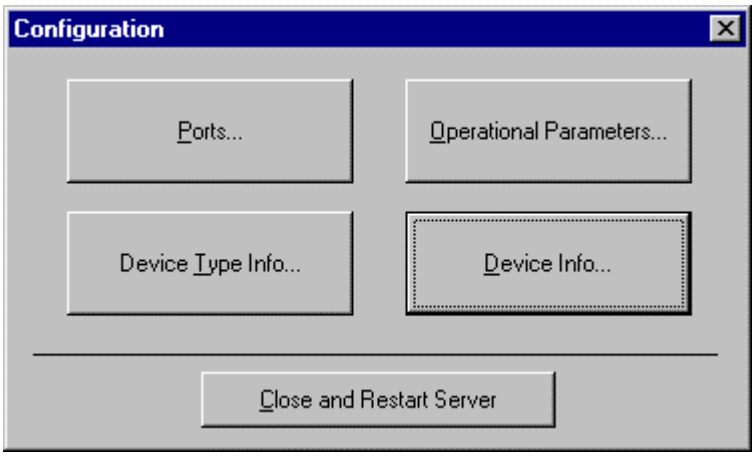



## Ports

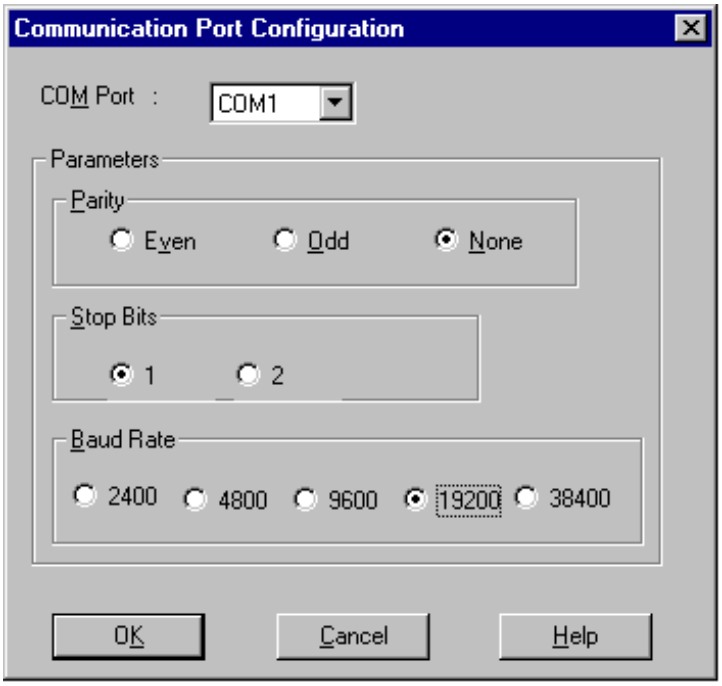
The Simulator supports up to 256 RS-485 ports. We refer to these communication ports as “comm ports” in the documentation. Communication ports must be configured before a device can be assigned to that port.

**Step 1.** Click on the **Configure** pull-down menu and select **Configure**.


At the **Configuration** dialog box click **Ports**.




This brings up the **Communication Port Configuration** dialog box:




**Step 2.** Select the communication port to configure from the **COM Port** pull-down list at the top of the dialog box.

**Step 3.** Select the appropriate radio button for each of the communication settings: **Parity**, **Stop Bits**, **Flow Control**, and **Baud Rate**. The default settings are shown.

Typically, only baud rate need be changed to match settings of the devices connected to the comm port. The other communication settings are fairly standard. Refer to the user manuals of the individual devices to be sure the communication settings match.

---

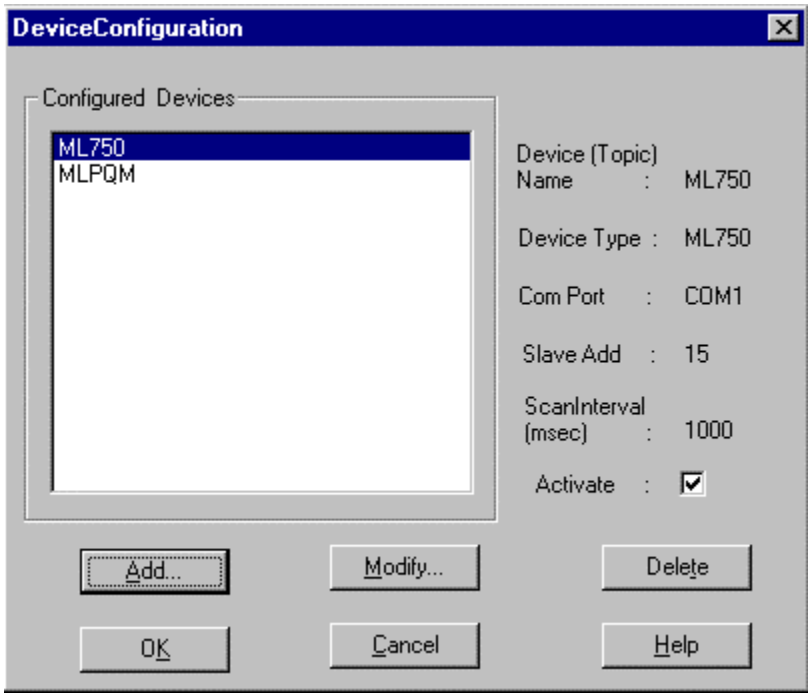
**NOTE**: Flow Control applies only to systems using an RS-232/RS-485 converter box other than the recommended converter. Some RS-232/RS-485 boxes require Hardware Flow Control to be enabled; check the product's documentation to see if your RS-485 converter requires this.

---

**Step 4.** Click on **OK** to return to the **Configure** dialog box.

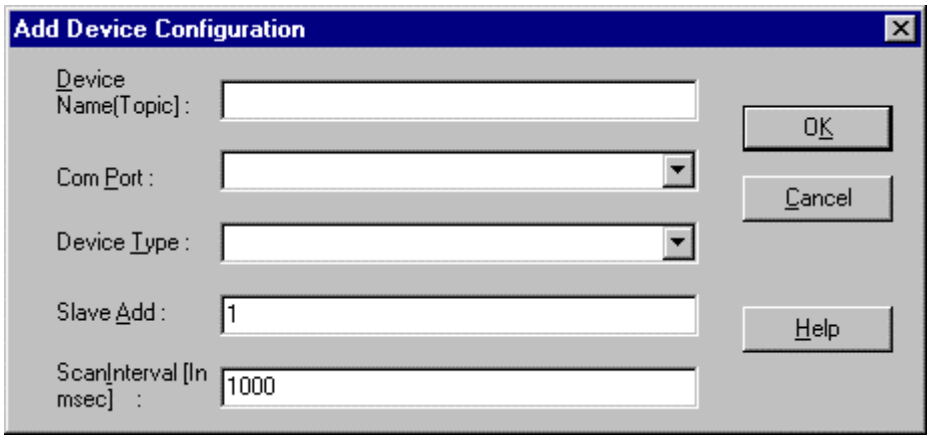
## Device Info

Click this button to display the **Device Configuration** dialog box:




### *Add*

Click this button to configure a new device. The **Add Device Configuration** dialog box appears:




The PMCS DDE Server doesn't just look at individual devices directly; instead it uses a more flexible virtual addressing scheme, which looks at *topics* at particular addresses. A topic consists of a user-specified device name, port number, address, and device type, with information on how often it is to be scanned. While there is usually just a single topic per device, it is possible to have multiple topics obtaining data from the same device. This feature allows you the ability to scan different data from the same device at different intervals.

For example, suppose you want to scan a set of registers at one device every 1000 milliseconds, while for other registers a 5,000 millisecond scan is adequate. In this

case, you could set up two separate topics, each addressed to the same device, but set to scan at different scan intervals.


Enter a name for the device. This will be the topic name referenced from client applications. This name must be unique, must begin with an alphabetic character, be no more than 20 characters long (8 characters if used with a third-party HMI client). Only alphanumeric characters and underscores are permitted; the topic name cannot include spaces or special characters, such as: + * / /, ? () “ ‘.


Select the **Device Type** from this pull-down list.


Enter the Slave address assigned to the device in this field. On a single RS485 network, all addresses must be unique, though it is acceptable to have devices with the same address if they are on different RS485 networks. For example, Network 1, Device 1 is recognized as different from Network 2, Device 1. For purposes of the DDE Simulator, if you configure multiple devices with the same Modbus address, the Simulator automatically differentiates between them as though they were on separate RS485 networks.

The device addresses must fall into the following ranges:

- Modbus devices other than the Modbus Concentrator may have addresses in the range 1–247.
- POWER LEADER Modbus Concentrators must have addresses in the range 1–32.
- Commnet devices must have addresses in the range 33–247.

For details on addressing requirements, refer to GEH-6502, *PMCS Network Architecture Guide*.


Enter the scan interval to be assigned to the device (the default is 1000). The scan interval is the target time in milliseconds at which the current device will be scanned; it is the scan time that could be achieved under optimal circumstances. This will vary greatly from network to network, depending upon the number and type of devices being managed, the amount of data being requested by the client applications, and even the architecture of the network itself. The valid range for scan interval is 1 to 9999999; scan intervals less than 100 milliseconds are usually impossible due to network timing constraints.

Clicking **OK** accepts the entries and returns to the **Device Configuration** dialog box.

### *Modify*

Select the device (topic) to be modified and click the **Modify** button. The **Modify Device Configuration** dialog box appears. The options and fields in the **Modify Device Configuration** dialog box are similar to those in the **Add Device**

**Configuration** dialog box. Make any necessary modifications and click on the **OK** button. The modified information is reflected in the **Device Configuration** dialog box.

### *Delete*

To delete a device (topic), select it and click on this button.

### *Activate checkbox*

At the bottom right of the **Device Configuration** dialog box, you'll notice a check box labeled **Activate**. This box provides a one-step way to turn a device on or off at the Server. Make sure this box is checked for each device; devices that are not activated will not communicate with the Server and will not be available to DDE clients. Conversely, for troubleshooting purposes, you can use this checkbox to quickly take a device off-line.

## Device Type Information

**CAUTION**: Advanced users only.

Do not access this option unless you have studied the *Advanced Topics* section of the User's Guide and are sure of what you are doing. Misuse of this option may cause errors or malfunction of the Server.

See Section 6 of the User's Guide, *Advanced Topics*, for information on this option.

## Operational Parameters

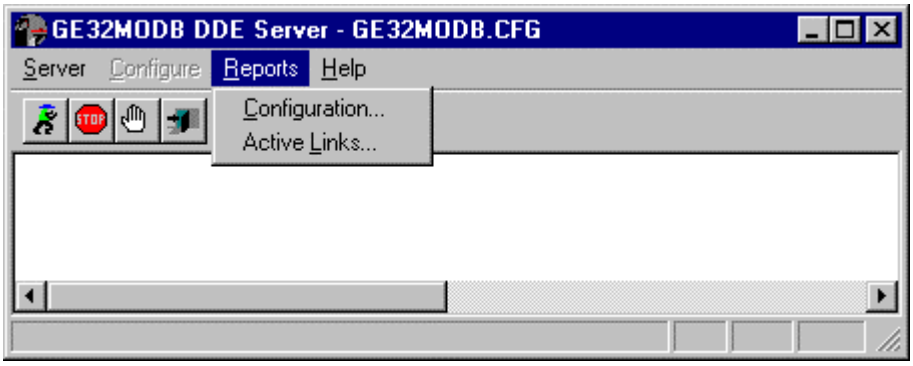
**CAUTION**: Advanced users only.

Do not access this option unless you have studied the *Advanced Topics* section of the User's Guide and are sure of what you are doing. Misuse of this option may cause errors or malfunction of the Server.

See Section 6 of the User's Guide, *Advanced Topics*, for information on this option.

# Reports Menu

The **Reports** pull-down menu is shown below. Descriptions of the reports options follow. The System Test Simulator generates two kinds of reports and saves them as .txt files on disk for viewing and printing. The **Reports** menu is always available; the Server need not be stopped to run a report.




## Configuration

Selecting this option generates a report on the current configuration of the Server. The report is saved as a text file containing the following information: configured device types (with the mnemonic list, function codes added, and register groups), configured comm ports (with port parameters), and configured devices (with DevTpName, ComPortName, SlaveAddr and ScanInterval). (Note that the Simulator assigns a com port number to each device, but that this is meaningless with regard to Simulator performance.)

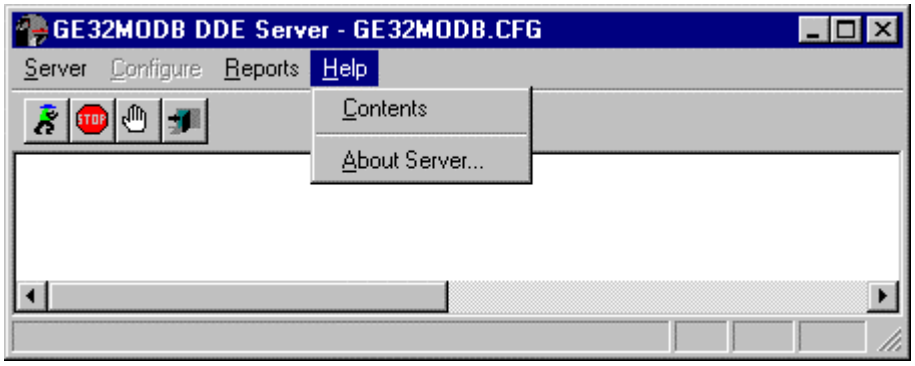
A dialog box prompts you for a file name and location to save the file. The default file name is config.txt.

## Active Links

The **Active Links** option (available only when in **Run** mode) creates a text file containing the following information: currently active comm ports and active devices (along with their device type, register groups, and mnemonic items). You will be prompted for a file name (default file name is active.txt). The information in this text file can be useful for debugging communications problems.

# Help Menu

The **Help** pull-down menu is shown below. The options are all standard Windows functions.




### *Contents*

Brings up a listing of all the topics for which help is available. The interface is standard Windows and is easy to use. If you are not sure how to navigate the Help windows, refer to your Windows manuals.

### *About Server*

Presents the program version and copyright information.

# Toolbar

The Simulator DDE Server's main window toolbar contains four buttons, shown below in Table 2. Click a button to perform the action described.

| Icon | Function | Description |
|---|---|---|
|  | Server Run | Starts the DDE Server. |
|  | Stop DDE Server | Stops DDE Server. |
|  or  | Suspend/Resume | Suspends or resumes the DDE Server's activities. When in Suspend mode, this button Resumes the Server's operation. When the Server is running, this button Suspends the Server. |
|  | Exit DDE Server | Exits DDE Server. |

*Table 2. Toolbar icons appearing on the Server main window.*

(This page left blank intentionally.)

# Troubleshooting

## Trouble-shooting the PMCS System Test Simulator

This section describes some simple steps that can be taken to isolate and correct communication problems. The problems described here represent the most probable causes of communication failure.

**NOTE**: This is a general trouble-shooting guide–it is not intended to cover every possible problem. If you still cannot establish communications after reading this section, call the GE Resolution Center, at 1-888-GE-RESOLV.

### WWLogger

WWLogger is a software application that is included with the PMCS installation. It is found in the GE_PMCS/TOOLS directory, and can be used for diagnostics purposes. The WWLogger application tracks all error messages generated by DDE communications or Server to device communications. Obviously this can be of tremendous value for trouble-shooting purposes. The messages tracked by WWLogger describe the application and topic or device that caused a given error. This information can then be used to diagnose and correct the cause of the error.

If you are experiencing trouble, launch WWLogger, and attempt to reproduce the error condition. You can then use the error messages displayed by WWLogger to troubleshoot your problem.

Error messages that may be encountered in the WWLogger are listed at the end of this chapter, in the section titled *Error Messages*.

## Communications – Client to Server

This section explains the most common error situations that can occur when attempting to establish a DDE conversation between client applications and the DDE Server.

When a client requires the status of a DDE item, it opens a link with the DDE Server and requests the data. The DDE Server retrieves simulated device data from the Simulator and sends it to the client application.

The DDE Server also begins monitoring the data and advises the client whenever a parameter changes. The client application simply tells the DDE Server what register, coil number, or I/O point to read or write. The DDE Server then automatically updates the client upon any change of that parameter.

Always start the DDE Server before starting any client software. If a client attempts to establish a link with a Server that is not running, an error message will result.

When an error message is displayed, note the information shown. You can often determine the source of the error message from the message itself – usually an application that is not running or a topic that is not properly configured.

Below are several situations that will cause a DDE conversation error message to appear:

1. **The DDE Server application is not running.** You can verify this by opening the Windows Task List (press Shift+Ctrl+Esc keys) and checking the list of currently running applications for the Server.
2. **The Server program name is misspelled in the DDE Access Name definition.** The Server is running, but its name is misspelled in one or more DDE Access Name definitions. The name entered in the DDE Access Name definition must be the DDE Server's actual program name (less the .exe extension) as seen in the Windows File Manager. If the name is misspelled, the Server can not be found.
3. **The topic is not defined in the Server or it is misspelled.** The Server may be functioning properly, but if a client requests data from a topic that doesn't exist, an error is generated.
4. **The mnemonic or register address is not defined in the Server or it is misspelled.** The Server may be functioning properly, but if a client requests data from a mnemonic or register address that doesn't exist, an error is generated.

Let's assume that the Server's name is spelled correctly in the client's DDE request and that the Server is running. The client is now looking for a topic defined in the Server; for example, Device1. To check the topic name, close the client (the Server cannot be configured if the client is running) and open the Server's program window. Invoke the **Configure:Device Info...** command. Is there a configured device listed as Device1 in the dialog box? Is it spelled exactly the same (including spaces, etc.) as it is in the DDE Access Name definition?

---

**NOTE**: Make sure there are no blank spaces after the topic name in both the Server or in the DDE Access Name definition in the client**.**

---

If nothing appears in the Server's window, try using WWLogger to track your error messages. The most probable error message indicates that the item used with one or more tagnames is unrecognized by the Server. Tagnames use specific naming conventions when accessing data from a Server, and deviation from these conventions can cause errors.

# NetDDE Troubleshooting

If you are using NetDDE to share a Server's data with clients across a network, make sure a share called GE32MODB|* exists in the system and has been trusted with proper parameters. A share need to be created only once, but each NT user should trust the share explicitly.

Refer to the NetDDE documentation for further information on how to set up shares and trusting a share. Information may also be found in the Microsoft Knowledgebase by searching for keyword "NetDDE".

**NOTE**: Make sure you're using the version of NetDDE supplied with Microsoft Windows NT 4.0 Workstation. Other versions of NetDDE are not supported by the PMCS DDE Server.

# Trouble-Shooting Chart

The following table lists a variety of common problems, possible causes, and recommended solutions.

| Error Condition | Possible Cause | Solution |
|---|---|---|
| Client application times out while setting up DDE link or receives negative acknowledgment. | Valid Data Timeout is set too low. | Increase the Valid Data Timeout value; it should be around 5000 milliseconds (5 sec) for most networks. |
| #NAME on Excel spreadsheet (client application) | Item name is incorrect. | Check for precise spelling and correct mnemonic name. Make sure the device is not deactivated. |
| #N/A on Excel spreadsheet (client application) | Data is currently unavailable. | Server may be down, suspended, or stopped, or port may be suspended.<br>Data request may not yet have been processed. |
| The Simulator or Server are showing an incorrect device configuration. | The Simulator may be using a different set of configuration files. | Check the GE32MODB.ini file to ensure that the Simulator is referring to the correct configuration files. |
| Starting the Event Server forces the GE32MODB Modbus DDE Server to start up, and closing GE32MODB forces the Event Server to shut down. Cannot run GE32MODB and the Simulator at the same time. | Your system components are configured as NT Services. Preconfigured Service dependencies are causing this behavior. | Shut down the PMCS NT Services. Start the DDE Simulator. Restart the PMCS NT Services. GE32MODB will not start if the DDE Simulator is already running. Be sure to shut down and restart the PMCS NT Services after you finish with the Simulator to restore normal operatoin. |

# Error Messages

This section provides descriptions of the error messages which may be generated by the PMCS DDE Server. These messages are logged in the WWLogger

**Activation of <item name> item failed, item not found in register's item list.**

Activation of an item failed when the client requested a link to this item. Please contact Customer Service. This error message does not appear in normal circumstances — it appears only if the item list is corrupt or if the toolkit gives an out-of-sequence ProtActivateItem() call.

**Could not open RMS data file <file name>**

The data file to store MicroVersaTrip device Peak Demand computation parameters could not be opened by the Server as defined in GE32MODB.INI. The .ini file entry may be wrong or the file may have been deleted.

**CServerApp::OnDeleteDev - Invalid toolkit index**

Internal error in the Server. Please contact Customer Service.

**DDE Execute <DDE execute command> failed**

Possible causes are:

a) The execute command is misspelled.

b) Parameters passed with the execute command are not proper. Refer to the Server interface specifications for correct parameter syntax and range.

c) The execute command string has some extra space characters. The Server does not accept extra space characters in execute commands. Please follow the Server interface specifications strictly.

d) The DDE execute command is not supported for the topic on which it was issued.

**<device name> device got deleted – discarding packet**

This is an informational message. A client application deleted a topic and the Server is deleting a scheduled communication packet.

**Error allocating new topic (<topic>, <toolkit id>) => 0**

The Server is unable to initiate a DDE transaction with the topic. The possible causes are :

a) The topic is not configured on the Server. Configure the topic in the Server.

b) The communication port associated with the topic could not be opened or initialized properly. In this case, the Server displays communication port-related error messages immediately preceding this error message in WWLogger.

c) The device type of the topic is not configured. Configure the topic's Generic Device Type in the Server.

**Failed to activate item(<logical device handle>, <Protocol Handle>)=>0**

Server internal error. Please contact Customer Support.

**Failed to create item (<logical device handle>) (<toolkit handle>) (<item name>) => 0**

Possible causes are :

a) The item name is not spelled correctly on the client.

b) The item register(s) is (are) not configured in any of the register groups of the device type. Check the device type's register-group configuration.

c) If the item name uses a mnemonic convention, then the mnemonic is not defined on the Server. Check the device type's mnemonic configuration.

**Failed to deactivate item(<logical device handle>, <Protocol Handle>)**

Server internal error. Please contact Customer Support.

**Failed to delete item(<logical device handle>, <Protocol Handle>)**

Server internal error. Please contact Customer Support.

**No Write to Individual Discrete Bits**

The client tried writing to individual bits of a read/write register, which is not allowed.

**RegFormat Name is INVALID**

The register name (either as mnemonic equivalent or as item name) is not valid.

**The device type <device type name> is NOT a valid type for topic**

The Server detected an unconfigured device type for a topic being created. Possible reasons are :

a) The <device type name> has been deleted from the device-type configuration file (GE32MODB.CFG). Reconfigure the device type that generated the error or, if you intend to remove all the devices of that particular type, go to device(topic) configuration and remove all the topics with this device type. NOTE: This applies to generic devices only.

b) GE32MODB.INI entry DocFile points to an incorrect device-type configuration file. Enter the correct path to the configuration file.

**Unable to free topic (hLogDev = <logical device handle>)**

Internal error for the Server. Please contact Customer Support.

**Unable to open RMS data file for rewriting RMS data**

The RMS data file has been deleted since the Server started. Restart the Server. The Server will create a new RMS data file during startup.

(This page left blank intentionally.)

# Advanced Topics

## Introduction

**CAUTION**: For advanced users *only*!

These advanced options are for experts only! Do not attempt these actions if you are not sure of what you are doing; it is possible to render your PMCS DDE Server inoperable, forcing you to reinstall it and potentially lose your device configurations. Please read all instructions before attempting any advanced configuration.

The information contained in this section applies to two types of devices:

- the broadly-defined generic devices that must be completely defined by the user
- the highly flexible GE Fanuc PLC family of products.

The GE Fanuc PLC 90/70, PLC 90/30, and Micro90 PLC consist of a backplane to which may be attached a wide variety of modules with a broad range of functions, from metering and data collection to process control. Because these devices are so flexible and there is no way to predict the options that will be chosen by the end user, they must be configured in the same manner as a generic device.

## Device Type Information – Adding Generic Devices

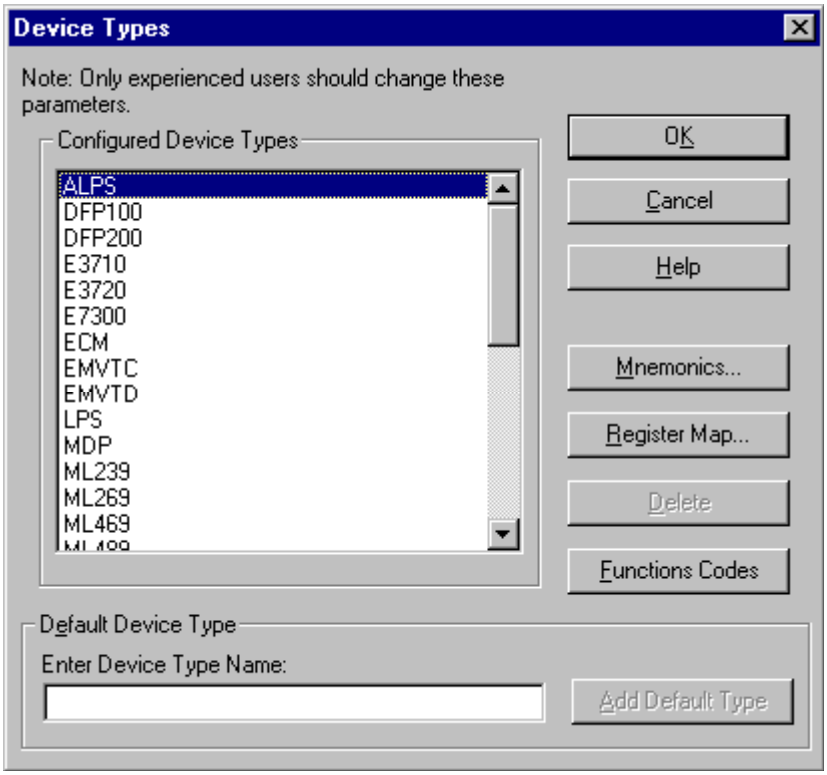
The PMCS DDE Server is preconfigured to support the POWER LEADER family of power-management devices, as well as a variety of third-party devices. However, you may add additional kinds of devices by creating your own device type. These "generic" devices have the register maps and functions that you assign, as explained in this section. The generic device type does not support the use of the DDE Server's special device handling, such as automatic time synchronization, waveform capability, or event handling.

The procedure for adding a generic device is as follows. The details of each option mentioned are provided after the procedure. Remember that the Server must be stopped before you can do any configuration.

**Step 1.** Select the **Device Type Info** option from the **Configuration** menu to display the **Device Types** dialog box, shown below.

**Step 2.** Enter the name for your new generic device in the field at the bottom of the dialog box. The field is labeled **Enter Device Type Name**. Click the **Add Default Type** button to add the new device type to the list of **Configured Device Types**.

**Step 3.** Next, select the generic device from the **Configured Device Types** list, and click the **Function Codes** button to add the functions codes supported by this device type. Setting up the register groups for a device type is mandatory; you may also enter an optional mnemonic list. To do this, you'll need to have on hand a complete protocol reference for the device before proceeding.




## Add Default Type


Entering the name of a new device type in the **Enter Device Type Name** box enables the **Add Default Type** button. Names may be up to 20 alphanumeric characters long and cannot include embedded spaces or special characters, such as + * / /, ? () " '.

Click on this button to add your new device type. You will need to do further configuration after creating a new device type. See the Register Map section for details on defining the device type's register map.

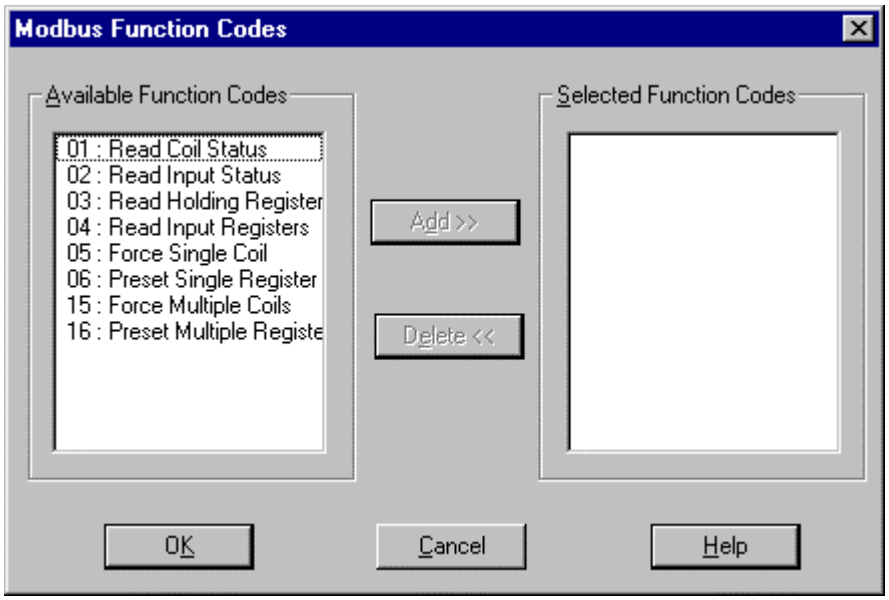
## Delete

The standard device types may not be deleted, so this button is grayed when a standard device type is selected. Only generic (user-defined) device types may be deleted. Select a user-defined device type and click **Delete** to delete that device type.

## Function Codes


Click on this button to attach or detach function codes to a device type. You will need the device's Modbus RTU protocol specification to know the correct codes to add for the new device type.

Clicking this button displays the **Modbus Function Codes** dialog box:




The two list boxes, **Available Function Codes** and **Selected Function Codes**, are multiple-selection list boxes. The **Available Function Codes** box lists the function codes available to any generic device type; the **Selected Function Codes** box lists those function codes that are to be assigned to a specific device type. Place the desired function codes into the **Selected Function Codes** box using the **Add** and **Delete** buttons, then click **OK**.


Select the function codes to be added from the **Available Function Codes** list box and click this button. The selected function codes are added for the device type and displayed in the **Selected Function Codes** list box.


This button is displayed only when there are no user-configured register groups for the device type. To remove a function code support from the device type, all the register groups that are added are removed. (To delete a register group from the

device type, refer to *Delete* under the *Register Map* subsection.) Select the function codes to be deleted from the **Selected Function Codes** list box and click this button. The selected function codes are deleted for the device type and displayed in the **Available Function Codes** list box.

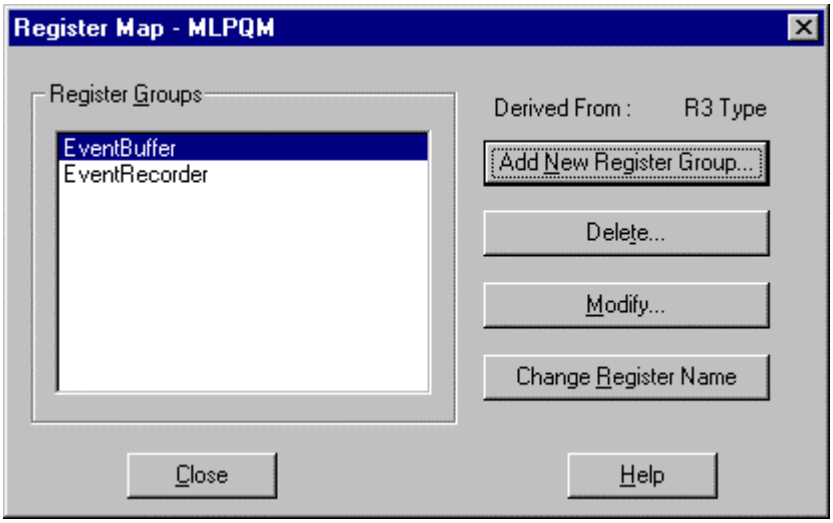
**NOTE**: You cannot delete function codes if any user-configured register group exists for this device type. To delete function codes, you must first delete all the register groups calling those functions.

## Register Map


Click this button to add, delete, or modify register groups for the selected device type.

**NOTE**: Register Maps for preconfigured devices may not be modified or deleted. When a preconfigured device type is selected, you will find that buttons for changing register maps, function codes, and mnemonics are disabled. Only generic devices and PLC type-devices are intended to have any modifications to their Register Groups.

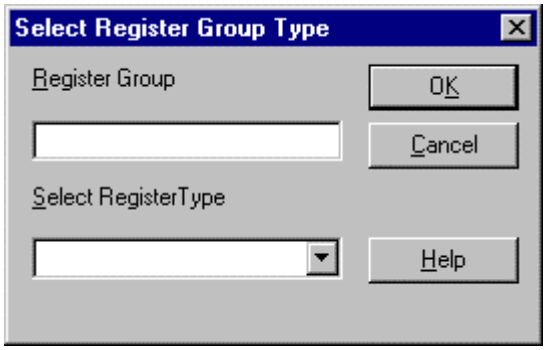
When this button is clicked, the **Register Map** dialog box appears:




The **Add New Register Group...**, **Delete...**, **Modify...**, and **Change Register Name** buttons are described below.

### *Adding a Register Group*


Click this button to add a new register group. The **Select Register Group Type** dialog box appears:



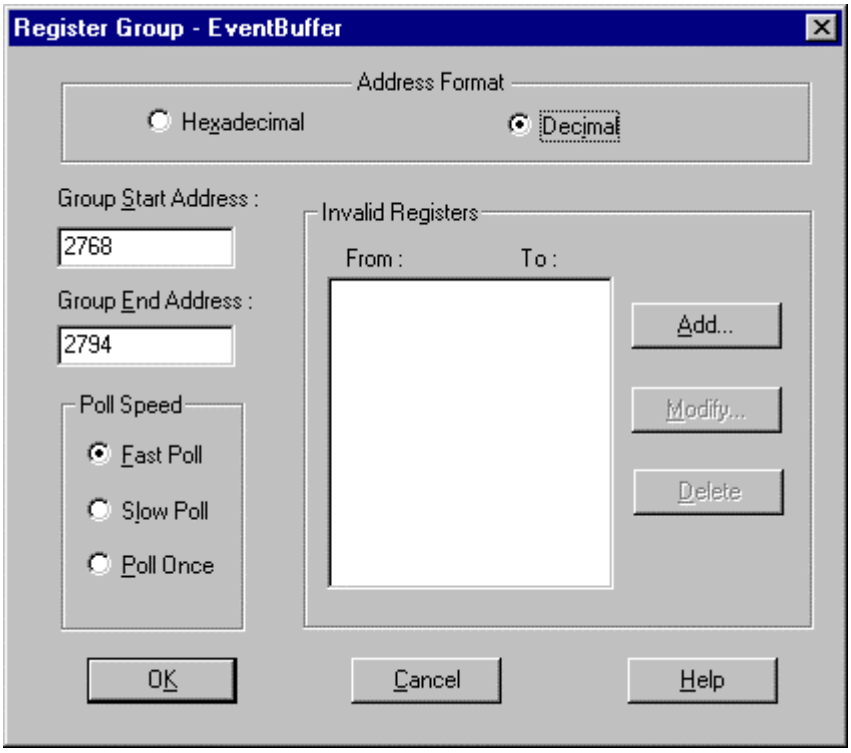

Enter the **Register Group** name. The name may be up to 20 alphanumeric characters and cannot include embedded spaces or special characters, such as: + * / /, ? () “ ‘.

Select the register type from the **Select Register Type** pull-down list. The **Modbus Register Types** shown are determined by the function codes selected previously. The four types of register groups that support the Modbus function codes are R0, R1, R3, R4. The table below describes the types, registers, supported codes, and uses.

| Type Code | Type of Register | Supported Function Codes | Use |
|---|---|---|---|
| R0 | Coils | 01 | Reading coil status |
| R0 | Coils | 05 | Setting/forcing/executing coils |
| R0 | Coils | 15 | Setting/forcing multiple coils |
| R1 | Contacts or discrete inputs | 02 | Reading contact/discrete inputs |
| R3 | Actual value or input register | 04 | Reading actual value or input registers |
| R4 | Setpoint or holding register | 03 | Reading setpoint or holding registers |
| R4 | Setpoint or holding register | 06 | Presetting single setpoint register |
| R4 | Setpoint or holding register | 16 | Presetting multiple registers |

You will need the device’s Modbus RTU protocol specification for the correct codes to add or delete.

Click **OK** to accept your new register group; the **Register Group** dialog box is displayed for you to finish defining the characteristics of the new register group.




The PMCS DDE Server supports both decimal and hexadecimal address formats. Select the type of addressing you wish to use (refer to the device user manual if you are unsure) by clicking either the **Decimal** or **Hexadecimal** radio button. Next, enter the **Start Address** and **End Address** for the register group in the appropriate boxes. The PMCS DDE Server supports Modbus RTU extended addressing; in decimal mode, the range is 0 to 65535; in hexadecimal mode, the address range is 0 to FFFF. Refer to *Appendix A – Register Addressing Conventions* for more information on formatting register addresses.

Next, set the desired polling speed by clicking one of the radio buttons - **Fast Poll**, **Slow Poll**, or **Poll Once**.

The polling speed is based on the scan interval for the topic. A polling speed of **Fast** attempts to scan at the assigned **Scan Interval**. This is the default. Setting the polling speed to **Slow** causes this register group to be read once every ten scan intervals. This is typically used for setpoints or low-priority parameters that change infrequently. Slow Poll is a multiple of **Fast Poll**, with a default value of 10. This multiple is controlled by a setting in the PMCS DDE Server's .ini file, and may be changed if desired. See the section *PMCS DDE Server .ini File*.

For example, if the scan interval is set to 1000 msec (1 second – this is the default value), setting the **Poll Speed** to **Fast Poll** causes the Server to scan as close to every 1000 msec as possible (based on the network load, communication bandwidth availability, etc.); setting the **Poll Speed** to **Slow Poll** reduces the scan attempts to once every 10,000 msec (10 seconds), or as close to this as possible; i.e.; not less than 10 seconds (≥ 10,000 msec.)

**Poll Once** means that this register group is scanned only the first time data is requested by a client application. All future requests for the same data receive the data from this poll. If the device is later declared to be dead, the device is scanned periodically until it comes back on-line, at which point the **Poll Once** data are reread.

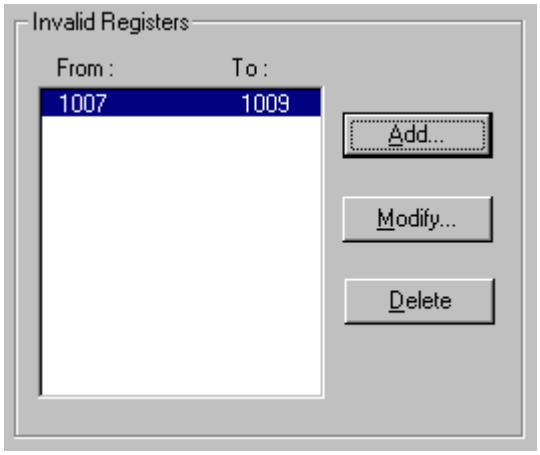
This is useful for registers with settings of a more permanent nature, which need to be read but are unlikely to change.

### *Invalid Register Blocks*

A register block consists of a contiguous piece of a device's memory that contains multiple pieces of data. Within this block, however, some addresses may be unused by the device. We can make the PMCS DDE Server aware of these unused or extraneous addresses by specifying them as invalid register blocks. (The Server's register maps for supported devices have already been optimized; only generic devices' invalid register blocks must be entered manually.)

An invalid register block is a list of registers within a register group that should not be polled for data by the Server, perhaps because they are unused by the device itself or simply because they are not desired.

Invalid register blocks allow the Server to focus its attention on only those registers of interest; by not asking the Server to poll addresses that contain no data of interest, we can improve the Server's potential performance. Also, some devices, if asked for a valid register block that happens to contain invalid registers within it, will return a message that the entire block is invalid. Thus, it can be important to identify any invalid blocks to the Server.




## Add

Click this button to access the **Add Invalid Register Range** dialog box.

Enter the start and end addresses for the invalid block, following the addressing conventions.

## Modifying an Invalid Register Block

To modify an invalid register block, select the block from the **Invalid Registers** list box and click the **Modify** button. The **Modify Invalid Register Range** dialog box appears. Follow the method used for adding invalid register blocks to make changes to the register block.

## Deleting an Invalid Register Block

Clicking the **Delete** button deletes the selected invalid register block.

### *Deleting a Register Group*

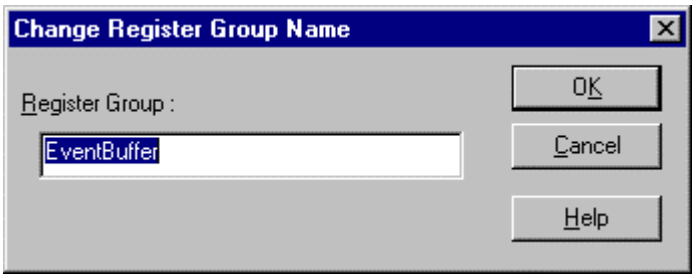
Clicking the **Delete** button deletes the selected register group.

### *Modifying a Register Group*

Click the **Modify** button to modify the register group. The **Register Group** dialog box appears. Make any necessary changes to the register group **Start Address**, **End Address**, **Poll Speed**, and **Invalid Register** blocks, then click on the **OK** button.

### *Changing a Register Group Name*


Click on this button to change the name of the selected register group. The **Change Register Group Name** dialog box appears:

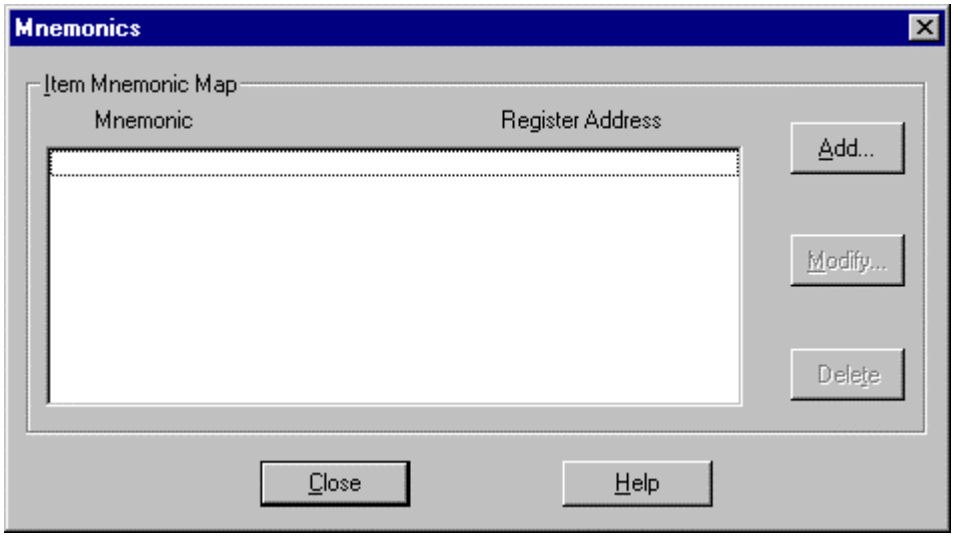



Enter the new name in the **Register Group** field, then click **OK** to accept or **Cancel** to back out of this dialog box.

## Mnemonics

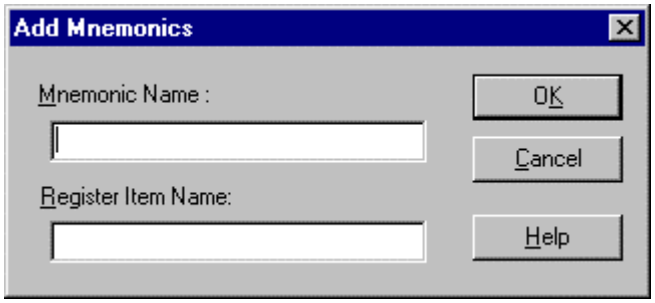

Mnemonics are an optional way of naming registers or groups of registers, by assigning an easily remembered name to a register address. From the **Device Types** dialog box, click this button to add, delete, or modify mnemonics. Mnemonics are useful to speed selection of registers – it's much easier to remember *Trip_Set_Points* than *R41234A5*.

When this button is clicked, the **Mnemonics** dialog box appears:




### *Adding a Mnemonic*

Click the **Add** button to add a mnemonic. The **Add/Modify Mnemonics** dialog box appears:




Enter the **Mnemonic Name**. You may use up to 20 alphanumeric characters, but no embedded spaces or special characters, such as + * / /, ? () “ ‘.

Enter the register address in this field. Follow the register-naming conventions presented in Appendix A.

Click **OK** to accept your changes.

### *Modifying a Mnemonic*

Click on the **Modify** button to modify the selected mnemonic. The **Add/Modify Mnemonics** dialog box appears. The procedure is similar to adding a mnemonic. Make any necessary modifications to the mnemonic name or register address, then click **OK**.

### *Deleting a Mnemonic*

Select a mnemonic from the **Item Mnemonic Map** list box and click the **Delete** button to remove it from the list.

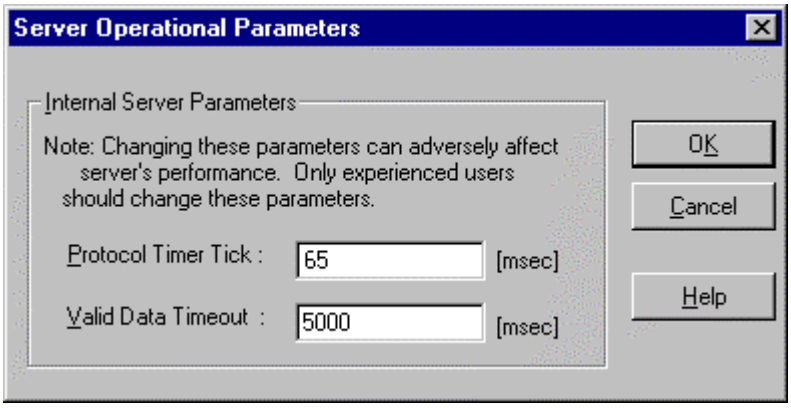
# Optimizing Server Performance

Every network has different devices, different numbers of RS485 ports to support, a different number of devices, and various client software requesting different amounts of data. System administrators with Modbus RTU network experience may be able to analyze their particular network's needs and adjust some of the PMCS DDE Server's advanced settings to improve its performance. These settings are discussed below.

With regard to the System Test Simulator and the server Operational Parameters, it is recommended that you not change any of these settings. There is little reason to alter these parameters, since they concern communications between the DDE Server and the devices in a real network. In the Simulator, the peculiarities and nuances of network communications have been replaced by the software of the Device Simulator. They are explained here to help you better understand PMCS.

## Operational Parameters

Clicking on this button in the **Configuration** dialog box causes the **Server Operational Parameters** dialog box to appear:




### *Protocol Timer Tick*

The **Protocol Timer Tick** is the time interval (in milliseconds) at which protocol is executed; the frequency at which the Server checks for work to do. A good analogy might be that the protocol timer tick serves as a metronome or heartbeat to the Server. Every tick, the Server executes a function, whether it is to listen at a particular port or to send a message to a device. The default is 65 milliseconds. At values below this level, the Server may query devices too quickly for the devices to consistently answer correctly.

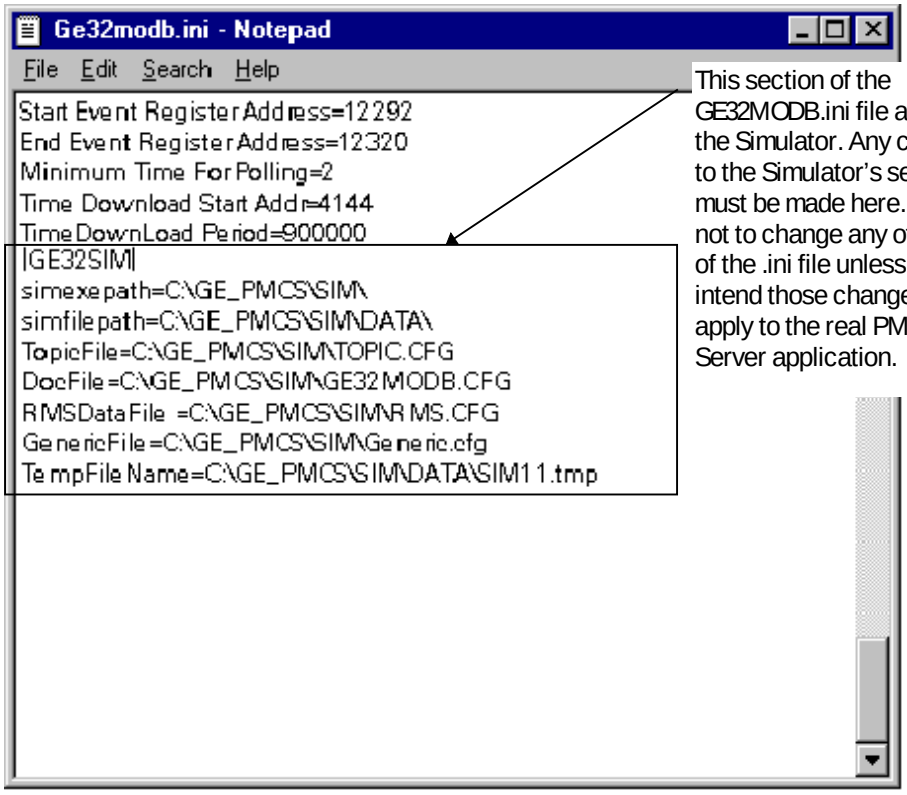
When the **Protocol Timer Tick** field has been changed, the **OK** button is enabled. It is disabled until any changes are made. Click on this button to accept changes. The changes take effect the next time the Server software is launched (not to be confused with **Run** from the Server menu).

**CAUTION**: Changing the Server operational parameters can adversely affect the Server's performance. If you are unsure of how to adjust any of these settings, consult Customer Support before making any changes.

## PMCS DDE Server .ini File

The application program's .ini file contains several lines that may be modified by knowledgeable and qualified personnel to fine-tune performance of an individual network. Do not modify these items unless you are experienced with the PMCS DDE Server.

**NOTE**: Both the real PMCS Modbus DDE Server program and the Simulator DDE Server program use the same .ini file, which points to configuration files for both applications. Be careful when making changes to the .ini file; be sure that you are making changes to the Simulator section of the file only.




As explained above, these settings should not usually be modified for purposes of the System Test Simulator. They are useful primarily for real PMCS networks.

**CAUTION**: Back up the application's current .ini file before attempting any modifications.

Open the application's .ini file with Windows Notepad. This file is named GE32MODB.ini and is located in the Windows NT directory. The parameters are grouped by subject:

[Server Operational Parameters],
[GE32MODB], etc.

Look for the GE32SIM group. These are the settings that apply specifically to the PMCS System Test Simulator.

### *Slow Poll Count*

The slow poll count is not used by the Simulator. Do not change this parameter.

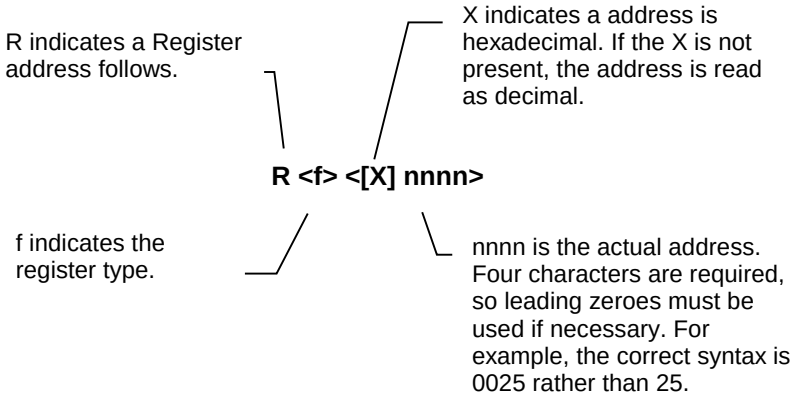
**NOTE**: Do not change any other parameters of the .ini file, as doing so may adversely affect Server performance.

# Appendix A - Register Addressing Conventions

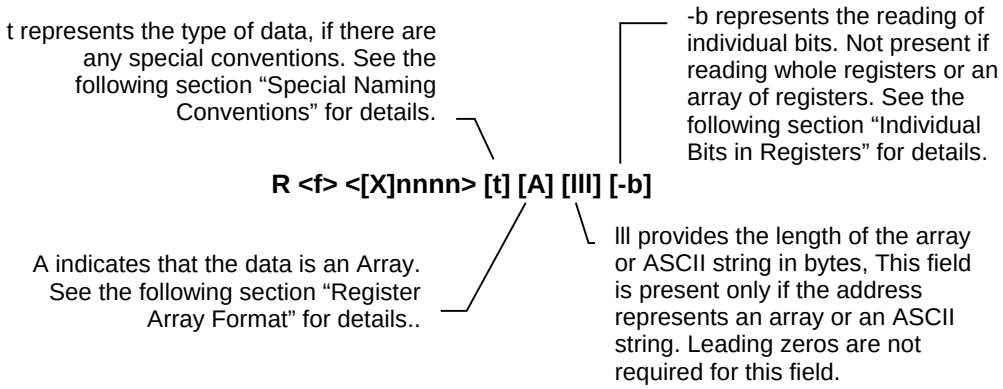
## Data-Addressing Conventions

The PMCS DDE Server is capable of interpreting both decimal and hexadecimal addresses. This allows access to the Modbus RTU protocol's extended register mode. The two addressing schemes are identical with one exception; in hexadecimal mode, an "X" is inserted prior to the address number to indicate that the address following is in hexadecimal format. The R character is ALWAYS present. Items in < > represent a variable numeric value. Capital letters in brackets [ ] indicate a hard character that does not change; it is either present or not. Lower-case letters in brackets [ ] indicate switches that may or may not be present; refer to the following sections for details.

The basic addressing scheme is as follows:




Switches may be used to modify addresses. The possible switches are shown below, and are detailed in the following sections:




## Standard Data Organization

Data is organized according to data type, numeric range, tag type, and access type.

### *Data Types*

There are four data types typically used by the GE devices. These four data types are the possible values for ‘f’ in the address. (Each data type is organized in a separate table for each device in this manual):

1. Dynamic Value
2. Setpoint
3. Command Coil
4. Fixed Value

Each data type is assigned a range of register numbers, tag type, and access as shown below:

| Data Organization | | | | | |
|---|---|---|---|---|---|
| **Data Type** | **Use** | **Register Range (hex)** | **Register Range (decimal)** | **DDE Tag Type** | **Type of Access** |
| Command Coil | 1. Commands a device to take action.<br>2. Reads the status of an action or discrete input. | R0X0000 – R0XFFFF | R00000 – R09999 | Discrete | Read and Write |
| Dynamic Value | Read frequently, such as metering values which change constantly | R3X0000 – R3XFFFF | R30000 – R39999 | Analog | Read Only |
| Fixed Value | Read only once at power-up. Info such as Product ID and configuration options | R4X0000 – R4XFFFF | R40000 – R49999 | Analog | Read Only |
| Setpoint | Read infrequently | R4X0000 – R4XFFFF | R40000 – R49999 | Analog | Read Only |

### *Examples*

Here are some examples of different types of register numbers:

| Register number | Represents |
|---|---|
| R00005 | Coil command, number 5, with Read/Write access to the user |
| R31005 | Dynamic value, number 1005, Read Only access |
| R43010 | Fixed value or Setpoint, number 3010, with Read/Write access to the user |

## Special Naming Conventions

Special handling of data from devices can be done by using the following conventions:

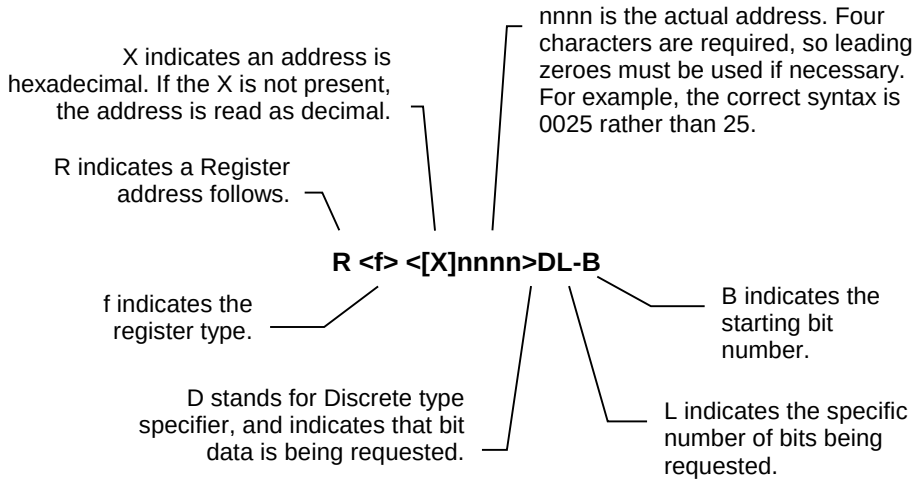
### *Long Words and Special Numbers*

By default, a register item is treated as an unsigned integer. To treat the contents of any register differently, refer to the table below:

| Special Data Item | Naming Convention | Example |
|---|---|---|
| Unsigned 16-bit Integer | Default | R40001 |
| 16-bit Signed Integers with values between -32,768 and 32,767 | Append letter I to item number. | R40001I |
| 32-bit Signed Integers (Long Integers) | Append L to item number. | R40001L |
| 32-bit floating point numbers | Append F to item number. | R40001F |
| Modulus 10000<br>Used in 32-bit register mode for EPM 3710 and 3720. | Append E to the item name<br>NOTE: See 3710 and 3720 ACM Modbus Protocol Manuals for details. | R40010E |
| ASCII data string | Append S to item number.<br>[lll] field immediately after S character represents the number of characters to read. If no length is specified ([lll] field is not provided), only one register of characters (2 or 4) will be read. The High byte represents the first character, and the Low byte represents the second character.<br>NOTE: No array type is allowed with S data items, nor are ASCII strings supported for coil registers.<br>NOTE 2: For most devices (16-bit mode), there are 2 characters per register. For 32-bit mode devices, there are 4 characters per register.<br>NOTE 3: The maximum value for the S string is 250. | R40010S020 |

### *Individual Bits In Registers*

Individual bits in registers are read-only. They can be read as discrete/integer tags by using the following notation (explained beneath):




### *Examples*

| Register Number | Represents |
|---|---|
| R40001D1-0 | Specifies least significant bit of first holding register |
| R30008D1-15 | Specifies most significant bit of an input register |
| R40001D2-5 | Specifies $6^{th}$ and $7^{th}$ from the least significant bit of first holding register. |

## Register Array Format

If multiple data items are being requested from a single topic, it is more efficient to request a block of contiguous registers than to place multiple requests for single registers. This is referred to as *register array format*. The register array format is used for the following applications:

- to read a block of register values into a column of cells in a worksheet (Excel or Access).
- to pass waveform data to a client application (refer to GEH-6509, *PMCS DDE Server Interface Reference*, for details)

The rules for register arrays are as follows:

1. A register array, or series of consecutive registers, can be treated as a block of numeric values. Up to 100 sixteen-bit registers or 50 thirty-two-bit registers can be read as a block. Enter the starting register address, and append it with type specifier “A”, followed by the length field. For example, the register address R30501A12 accesses registers 501 through 512 as a block.

2. When the DDE Server returns a new value for a register array to the client, it is in the form of a character string containing a value for each register, separated by a carriage return and line feed.

For example, for R30021A6, the values returned might look like this:

50<cr><lf>

17<cr><lf>

0<cr><lf>

5<cr><lf>

1007<cr><lf>

20<cr><lf>

NOTE: All arrays must be terminated by a null character (ASCII 0).

3. When the client application writes a value to a register array, it must be in the form of a character string containing a value for each register in the array. The register values can be separated by commas, tabs, spaces, carriage returns or line feeds. For example, for R40001A6, the value string could be written:

1,2,3,4,5,6

or

1<tab>2<tab>3<tab>4<tab>5<tab>6

or

1 2 3 4 5 6

(This page left blank intentionally.)

# Index

***GE Consumer & Industrial***

*General Electric Company*
*215 Anderson Avenue, Markham ON, L6E 1B3, CANADA*
*USA & Canada: 1-800-547-8629 Global: (905) 294-6222 Fax: (905) 201-2098*
*Email: multilin.tech@indsys.ge.com <mailto:multilin.tech@indsys.ge.com>*