IOLAN DS Family

SDS/SCS/STS
User's Guide

Version 2.8

Part #5500161-28

July 2007

FCC Note The IOLAN Device Server has been found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instructions in this Guide, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user will be required to correct the interference at his/her own expense.

EN 55022: 1998, Class A, Note

WARNING This is a Class A product. In a domestic environment this product may cause radio interference in which case the user may be required to take adequate measures.

Caution: the IOLAN Device Server is approved for commercial use only.

WARNING The IOLAN Device Server SDS T models operate in an ambient air temperature above 70 ^{o}C. However, at 70 ^{o}C and above, a burn hazard exists if the metal case is touched without proper hand protection.

Table of Contents

Preface

About This Book

This guide provides the information you need to:

- configure the Device Server
- incorporate the Device Server into your production environment

Intended Audience

This guide is for administrators who will be configuring the Device Server.

Some prerequisite knowledge is needed to understand the concepts and examples in this guide:

- If you are using an external authentication application(s), working knowledge of the authentication application(s).
- Knowledge of TFTP, the transfer protocol the Device Server uses.

Documentation

The following documentation is included on the Device Server installation CD:

- *IOLAN Device Server Family Quick Start Guide*
- *IOLAN Device Server User's Guide*
- *TruePort User's Guide*
- *TruePort Installation and Configuration Guide for Windows NT*
- Online Help in the DeviceManager (automatically installed with the DeviceManager application)
- Link to knowledge base

Typeface Conventions

Most text is presented in the typeface used in this paragraph. Other typefaces are used to help you identify certain types of information. The other typefaces are:

Typeface Example	Usage
At the C: prompt, type: `add host`	This typeface is used for code examples and system-generated output. It can represent a line you type in, or a piece of your code, or an example of output.
Set the value to **`TRUE`**.	The typeface used for `TRUE` is also used when referring to an actual value or identifier that you should use or that is used in a code example.
`subscribe` *`project subject`* **`run`** ***`yourcode`***`.exec`	The italicized portion of these examples shows the typeface used for variables that are placeholders for values you specify. This is found in regular text and in code examples as shown. Instead of entering *`project`*, you enter your own value, such as `stock_trader`, and for ***`yourcode`***, enter the name of your program.
File, **Save**	This typeface and comma indicates a path you should follow through the menus. In this example, you select **Save** from the **File** menu.
IOLAN User's Guide	This typeface indicates a book or document title.
See ***About the IOLAN Device Server on page 29*** for more information.	This indicates a cross-reference to another chapter or section that you can click on to jump to that section.

Online Help

Online help is provided in the DeviceManager. You can click on the What's This button (or) and then click on a field to get field-level help. Or, you can press the **F1** key to get window-level help. You can also get the *User's Guide* online by selecting **Help**, **Help Topics**.

Contacting Technical Support

Making a Technical Support Query

Who To Contact

Note: Perle offers free technical support to Perle Authorised Distributors and Registered Perle Resellers.

If you bought your product from a registered Perle supplier, you must contact their Technical Support department; they are qualified to deal with your problem.

Have Your Product Information Ready

When you make a technical support enquiry please have the following information ready:

Item	Write Details Here
Product Name	
Problem Description	
Your Name	
Company Name and Address	
Country	
Phone Number	
Fax Number	
Email Address	

Making a support query via the Perle web page

If you have an internet connection, please send details of your problem to Technical Support using the email links provided on the Perle web site in the **Support/Services** area.

Click here to access our website at the following URL:

http://www.perle.com

Repair Procedure

Before sending a Device Server for repair, you must contact your Perle supplier. If, however, you bought your product directly from Perle you can contact directly.

Customers who are in Europe, Africa or Middle East can submit repair details via a website form. This form is on the Perle website, **www.perle.com**, in the **Support/Services** area.

Click here to access our web site at the following URL:

http://www.perle.com/support_services/rma_form.asp

Feedback on this Manual

If you have any comments or suggestions for improving this manual please email Perle using the following address:

Email: ptac@perle.com

Please include the **title**, **part number** and **date** of the manual (you can find these on the title page at the front of this manual).

1 Introduction

About the IOLAN Device Server

The Device Server is an Ethernet communications/terminal server that allows serial devices to be connected directly to LANs. The Device Server can connect to a wide range of devices including:

- Terminals for multi-user UNIX systems
- Data acquisition equipment (manufacturing, laboratory, scanners, etc.)
- Retail point-of-sale equipment (bar coding, registers, etc.)
- PCs using terminal emulation or SLIP/PPP
- Modems for remote access and Internet access
- ISDN adapters for branch remote access and Internet access
- All types of serial printers

The performance and flexibility of the Device Server allows you to use a wide range of high speed devices in complex application environments. The Device Server will work in any server environment running TCP/UDP/IP.

IOLAN Device Server Models

The IOLAN Device Server comes in several different models to meet your production environment needs:

- **DS**—Offered as a 1-port unit, this model provides basic Device Server functionality. This model can be ordered with RJ45, DB9 male, DB25 female, or DB25 male connection options. There is also a line of DS models that support Analog Input, Temperature Input, Relay Output, and/or Digital I/O.
- **TS**—The model does everything the DS model does plus has two RJ45 serial ports (supports EIA-232 only). This model does not support the power out/power in pins or I/O.
- **SDS**—This model does everything the DS model does plus additional features such as external authentication, SSH, SSL, port buffering, email alerts, RIP, DNS/WINS, plus much more. This model has an EIA-232/422/485 switchable interface. Rack mount models have a dedicated Console port and support gigabit Ethernet. The 1-port model can be ordered with RJ45, DB9 male, DB25 female, or DB25 male connection options. Some SDS models support Power Over Ethernet or have an internal modem. There is also a line of SDS models that support Analog Input, Temperature Input, Relay Output, and/or Digital I/O; all models in this line are extended temperature models, meaning that they can operate in higher temperature environments.
- **STS**—This model does everything the DS model does plus additional features such as external authentication, SSH, SSL, port buffering, email alerts, RIP, DNS/WINS, plus much more. This model has an EIA-232 interface. Rack mount models have a dedicated Console port and support gigabit Ethernet. Some models support dual input DC power.

- **SCS**—This model does everything the DS model does plus additional features such as external authentication, SSH, SSL, port buffering, email alerts, RIP, DNS/WINS, plus much more. This model has an EIA-232 interface. Rack mount models have a dedicated Console port. This model comes equipped with PCI interface (supports the Perle PCI modem card), gigabit support, dual Ethernet, and can have dual AC power.

Device Server Features

The Device Server is a communications server used for making serial network connections. It attaches to your TCP/IP network and allows serial devices such as modems, terminals, or printers to access the LAN. It also allows LAN attached devices to access serial devices attached to the Device Server.

Hardware

The Device Server hardware features can include (depending on the model):

- Auto sensing 10/100/1000 RJ45 Ethernet interface.
- Universal, software-selectable EIA-232/422/485 interface (the SCS/STS models are only EIA-232).
- Full modem control using DTR, DSR, CTS, RTS and DCD.
- Tx and Rx activity indicators.
- External AC or DC power supply, or power over serial or Ethernet.
- LEDs for diagnostic testing.
- Self-test on power-up.
- Reset switch.
- PCI modem card.
- Dedicated console.
- Analog Input, Temperature Input, Relay Output, and/or Digital I/O.

Software

The Device Server software features include:

- Multiple ways to configure the Device Server:
  - Easy Config Wizard, an easy configuration wizard that allows you to complete basic Device Server configuration
  - DeviceManager, a fully functional Windows® configuration/management tool
  - WebManager, a web browser option for configuring/managing the Device Server
  - Menu, a window-oriented menu interface for configuration and user access
  - CLI, a Command Line Interface option for configuration/management and user access
  - SNMP, allowing remote configuration via SNMP as well as statistics gathering
  - DHCP/BOOTP, a method of automatically updating the Device Server
  - IOLAN+ interface, for IOLAN+ users, Device Server models with 16 ports or fewer can be configured using the IOLAN+ menu
- IPv6 support.
- Support for TCP/IP and UDP protocols including telnet, rlogin, and SSH.
- Remote access support including PPP, SLIP, and CSLIP.
- Printer support via LPD and RCP.
- Virtual modem emulation.

- ‘Fixed tty’ support for several operating systems (TruePort).
- DHCP/BOOTP for automated network-based setup.
- Dynamic statistics displays and line status reporting for fast problem diagnosis.
- Multi session support on a single terminal.
- Interoperability with IP routing through gateway tables.
- Domain Name Server (DNS) support.
- WINS support for Windows® environments.

Security

The Device Server security features can include (depending on your Device Server model):

- SSH connections.
- SSL connections.
- Supervisory and port (line) password.
- Port locking.
- PPP authentication via PAP or CHAP.
- Per-user access level assignment.
- Logging via Syslog.
- RADIUS accounting.
- Email notification.
- External authentication using any of the following systems:
  - RADIUS
  - Kerberos
  - TACACS+
  - NIS
  - SecurID
  - LDAP
- Trusted host filtering, allowing only those hosts that have been configured in the Device Server access to the Device Server.
- Idle port timers, which close a connection that has not been active for a specified period of time.
- Ability to individually disable daemons/services that won’t be used by the Device Server.

Supported Products/Versions

Web Browsers

The WebManager has been tested on Windows and Linux with the following web browsers:

- **Netscape**—7.x
- **Internet Explorer**—6.x
- **Mozilla Firefox**—1.x

SNTP

Versions 1, 2, 3, and 4 are supported. SNTP version must be specified for SNTP configuration.

SSH

SSH 1 is supported with the following ciphers:

- **Blowfish**
- **3DES**

SSH 2 is supported with the following ciphers:

- **3DES**
- **Blowfish**
- **AES** (128/192/256-bit)
- **CAST128**
- **Arcfour**

Typical Applications Summary

Managing the Device Server

The Device Server can be managed and configured by administrators through various methods, allowing them full configuration capabilities and easy access to management statistics and tools. Administrators can access the Device Server using the following methods:

- Connection through Ethernet using the DeviceManager, a Windows-based configuration application.
- Connection through Ethernet using WebManager, via a web browser.
- Direct connection to the serial port using a Serial Terminal or Terminal Emulation Software.
- From the network through the Ethernet interface using reverse Telnet (Port 23) or reverse SSH (Port 22).
- Through a serial port configured for PPP/SLIP allowing for remote access (Telnet session) through a modem.
- Through an SNMP agent, using the Device Server MIB.

Managing/Accessing devices attached to the Device Server

The Device Server can be configured to allow users or administrators to view or manage specific devices on the Device Server's serial port across the Ethernet interface using two different methods.

- **Direct Connect**—users can directly connect to the device on the serial port by Telnet or SSH (**Line Service** must be set to **Rev Telnet** or **Rev SSH**) using the Device Server's configured IP address and the serial device's assigned TCP port number.
- **Easy Port Access**—users can connect to the Device Server using the configured Device Server's IP address by reverse Telnet (port number 23) or reverse SSH (port number 22), and are provided with a device menu displaying the name of the device that the user has access to. This feature eliminates the need for administrators and users to recall the specific port number associated with a certain device connected to the Device Server. The user can simply connect to a specific device based upon the name of the device and then return to the device menu without disconnecting its initial reverse Telnet or reverse SSH connection.

Network Security

The Device Server provides a comprehensive suite of security features to allow an organization to implement robust security planning to prevent unauthorized access. These include several external authentication methods, trusted host filtering, and the ability to disable individual services.

For a secure LAN connection, the Device Server supports SSH version 1 and version 2 protocol. Remote server connections with SSH protocol uses an encrypted data channel with support for password and public key authentications.

2 Installation

Introduction

This chapter tells you what is packaged with your IOLAN Device Server, how to power up the Device Server to make sure it works correctly, and how to assign the Device Server an IP address through the LAN.

IOLAN Device Server Components

What's Included

When you open your IOLAN Device Server package, you should have the following components:

- The Device Server
- External power supply (unless it's a P series (power over Ethernet) or an I/O model)

Note: If the desktop Device Server model was bought in bulk, you must supply the power supply. For rack mount models, the power supply is included for AC power models only.

- *Quick Start Guide* (for I/O models, a soft copy exists on the CDROM)
- Warranty Card
- A CD-ROM containing documentation, firmware, DeviceManager, etc.
- Administration cable (consisting of an RJ45-->DB9F adapter and a 3' RJ45 cable) for models that have an RJ45 connector

Added components for rack mount models:

- Administration cable (consisting of an RJ45-->DB9F adapter and a 3' RJ45 cable)
- Rack mounting kit
- (SCS models only) IOLAN wiring starter kit (see ***Appendix F, Accessories*** on page 383 for pinout diagrams).

What You Need to Supply

Before you can begin, you need to have the following:

- A serial cable
- An Ethernet 10/100/1000BASE-T cable if you are connecting the Device Server to the network

Available Accessories

The following accessories are available for purchase for the Device Server:

- DIN Rail Mounting Kit (35mm) for the desktop models
- PCI modem card for SCS rack mount models
- 3 meter RJ45M-RJ45M 8-wire Sun/Cisco modular cable
- RJ45 to DB25 DTE Male adapter
- RJ45 to DB25 DCE Male adapter
- RJ45 to DB25 DTE Female adapter
- RJ45 to DB9 Male DTE adapter
- RJ45 to DB9 Female DTE adapter

Contact your distributor for details.

Desktop Model Power Supply Requirements

Serial Only Models

If you are providing a power supply for a desktop Device Server model, your power supply must meet the following requirements:

- Output between 9-30V DC.
- The cable attached to the power supply should be about 20AWG, length 6 feet approx. The barrel dimensions of the cable-plug are OD=5.5, ID=2.1, and length= 9.5mm, with a straight barrel, and positive polarity on the inside and negative polarity on the outside.
- Power can also be provided by pin 1 on the DS/SDS1 model; Serial Port 2, pin 1 on the SDS2 model; Serial Port 4, pin 1 on the SDS4/SCS4 models; or over Ethernet on the P series models (power over Ethernet).

I/O Models

If you are providing a power supply for a desktop Device Server I/O model, your power supply must meet the following requirements:

- Output between 9-30V DC and a minimum of 600mA current.

Note: The maximum load for the Relay channel is 1A @ 30VDC or 0.5A @ 120VAC.

Rack Mount DC Power Requirements

Read this section if your Device Server model has 48V dual DC power.

Electrical Supply Details

The Device Server is supplied with an integral Terminal Connections block to facilitate connection to a DC source(s). The DC supply(s) should have adequate over-current protection within the closed rack system and comply with local or national standards applicable to the installation territory.

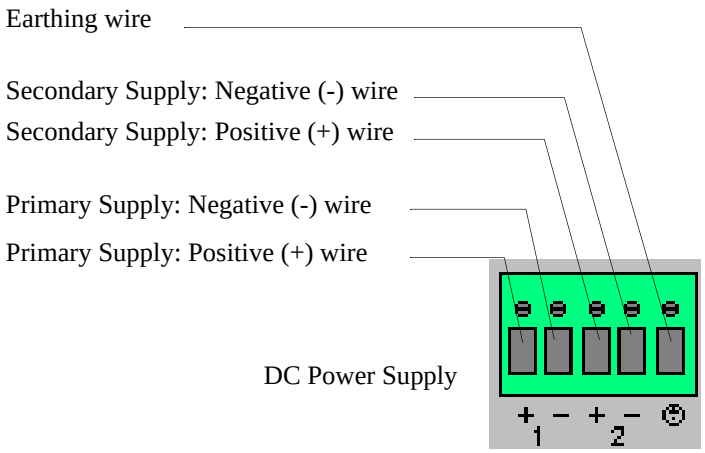
Note: The equipment must be grounded for safety and to ensure ESD protection for correct operation and protection of the internal circuitry.

Connecting DC Power Supply(s) to the Device Server

Connecting the DC supply(s) to the Device Server should be performed in the following sequence:

1. Switch Off the Power Supplies and the Device Server.
2. Connect the attached devices to the serial ports.

3. Connect the primary and secondary DC input using the following specifications:
   a. Use wire gauge 20 to 22 AWG.
   b. Strip insulation 7mm from wire ends. (If using stranded wire, twist all strands together to ensure all wire strands are used for the connection.)
   c. Connect supply with reference to the terminal block diagram and electrical specifications:



Note: When connecting only a single power supply source, ensure the connection is the primary supply and the secondary terminals are left unconnected.

Primary Supply:

Positive (+) wire to Circuit 1, terminal marked +

Negative (-) wire to Circuit 1, terminal marked -

Secondary (back-up) Supply:

Positive (+) wire to Circuit 2, terminal marked +

Negative (-) wire to Circuit 2, terminal marked -

Note: When connecting dual power supply sources, the Device Server supports a common positive (+) circuit arrangement ONLY.

Earthing Wire:

Ground wire to terminal marked with circular earthing symbol.

Screws:

Tighten terminal connector block screws to 7 lbs-inches torque.

4. Switch On the power supplies.
5. Switch On the Device Server. (The power LEDS 1 and 2 will indicate the status of the power source at the respective input. If both the primary and secondary power source are available, both LED 1 and LED 2 will be luminated indicated power detected from each input.)

Disconnecting 48V Power Supplies from the Device Server

To disconnect the power supply(s) from the Device Server, do the following:

1. Switch off the Device Server.
2. Switch off the power source(s).
3. Disconnect all DC power input cables from the Device Server terminal connector block.
4. Remove any attached devices to the serial or Ethernet port(s).

Your Device Server is ready to be moved.

Power Over Ethernet Specifications

The IOLAN Device Server SDS P models can only accept power from an IEEE 802.3AF compliant PSE device. Power Source Equipment (PSE) can provide up to 13W of power to a powered device, in this case, the Device Server, using one of the following methods:

- Using the two unused twisted pair wires (10/100Mb only).
- Using the two data pairs or "phantom power" method (100Mb).

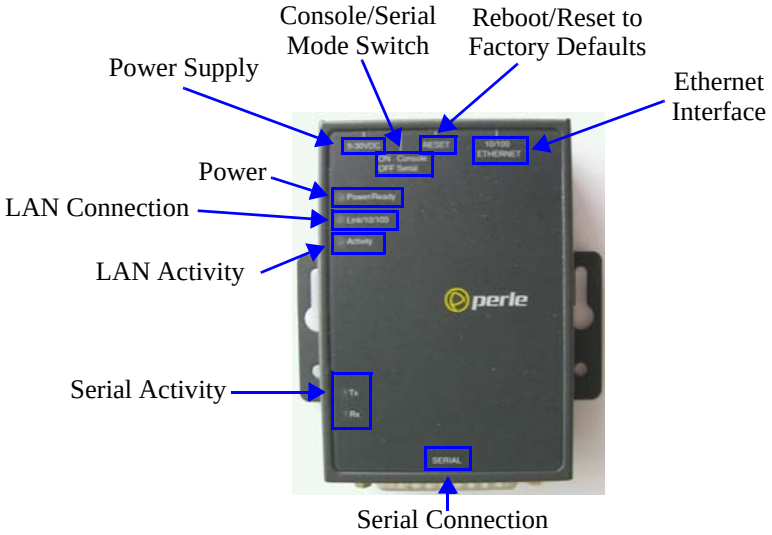
The 1-port/4-port SDS P model comes with an external power supply, while the 2-port SDS P model does not come with an external power supply option. If you are using the power over Ethernet feature in conjunction with the serial power pinout, the power output is always 5 volts, regardless of how the jumpers are set.

Getting to Know Your Device Server

The inset RESET button will reboot the Device Server if pushed in and released quickly and will reset the Device Server to factory defaults if pushed in and held for more than three seconds.

1-Port

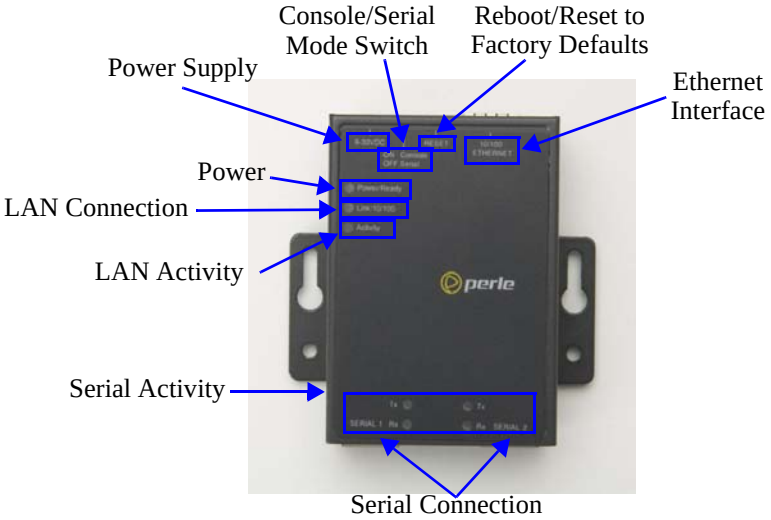
This section describes the components found on the Device Server 1-port models.



The 1-port Device Server has one serial connection that is one of the following: DB25 male, DB25 female, RJ45, or DB9 male.

2-Port

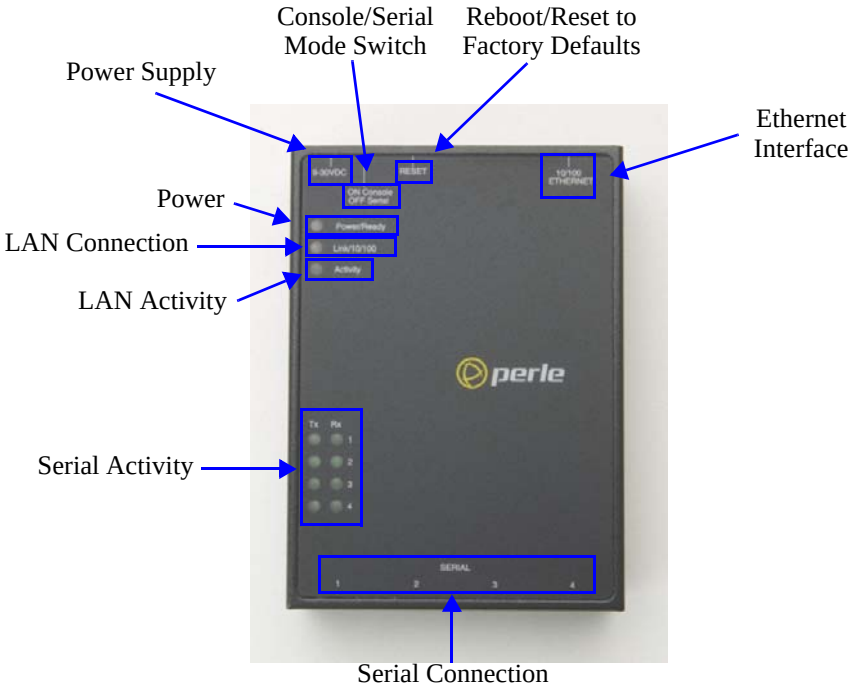
This section describes the components found on the Device Server 2-port models.



The 2-port Device Server has two RJ45 serial connections. If you are using the 2-port Device Server, you can use an 8-pin connector if you do not need the power in (pin 1) or power out (pin 10) pins. The 2-Port P model (power over Ethernet) does not come with a power supply.

4-Port

This section describes the components found on the Device Server 4-port models.

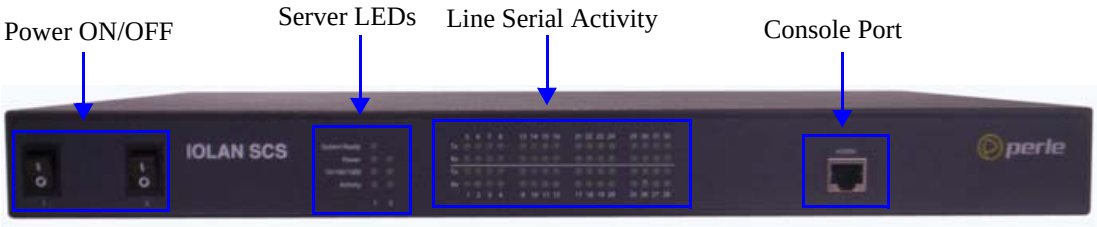


The 4-port Device Server model has four RJ45 serial connections.

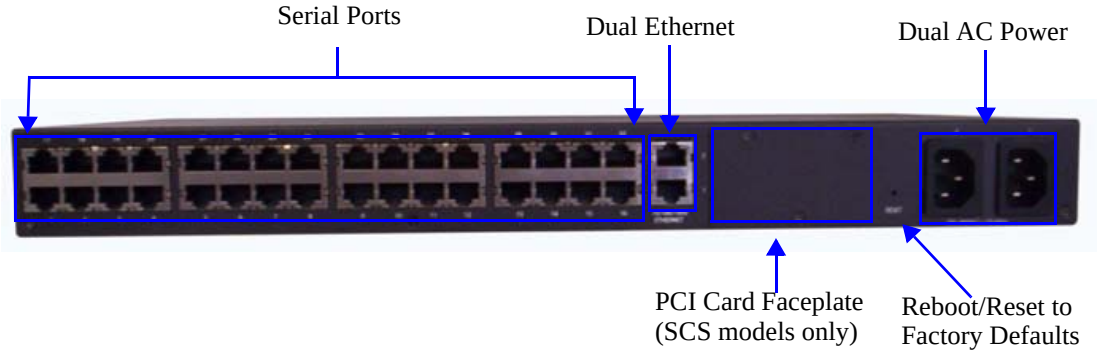
Rack Mount

This section describes the basic components of all rack mount Device Server models. This example uses an IOLAN SCS with dual ethernet and dual AC power.

Console Port/LED View



Serial/Ethernet View



Installing a Rack Mount Device Server

Using the rack mount brackets included with your Device Server, you can rack mount the Device Server from the front or the back of the chassis, depending on your environment. Make sure you don’t block the Device Server’s side air vents. Each Device Server is 1U in height, and does not require any extra space between units; therefore, you can rack mount up to five Device Servers in a 5U rack.

LED Guide

Desktop Models

The Device Server LEDs display the following information:

- **Power/Ready**—(Green/Red/Yellow) This LED starts out red at the beginning of power up. If this LED remains red, indicates that there is a critical error (see ***Hardware Problems*** on page 369). It flashes green to indicate that the Device Server is booting, then flashes green/yellow when the firmware is being updated. This LED then remains solid green to indicate that the Device Server is ready. When the Device Server is in Console mode, this LED will flash green.
- **Link/10/100**
  - Green—10 Mbits
  - Yellow—100 Mbits
  - Off—no LAN connection
- **Activity**—Flashes Green for transmit (TX) or receive (RX) LAN data
- **Tx**—Flashes with transmit serial activity
- **Rx**—Flashes with receive serial activity
- **Downloading firmware**—(Green/Yellow) The Device Server will flash green/yellow, indicating that it is downloading new firmware.

Rack Mount Models

The Device Server LEDs display the following information:

- **Power/Ready**—(Green/Red/Yellow) When the Device Server boots up, it can experience one of four possibilities:
  - **Good Boot:** When the Device Server cycles through a good boot, the Power/Ready LED cycles for several seconds and then stays a solid green.
  - **Noncritical Error Boot:** When the Device Server cycles through a boot and a noncritical error occurs, such as a bad port, the Power/Ready LED will flash red briefly before displaying a solid green. You should reboot the Device Server while monitoring the Console port to view the error information.
  - **Critical Error Boot:** When the Device Server cycles through a boot and a critical error occurs, such as corrupted firmware, the Power/Ready LED continues to flash red. View the Device Server reboot through the Console port for information on how to correct the problem.
  - **Fatal Error Boot:** When the Device Server cycles through a boot and a fatal error occurs, the Power/Ready LED stays a solid red (see ***Hardware Problems*** on page 369).
- **Link/10/100/1000**
  - Green—10/100 Mbits
  - Yellow—1000 Mbits
  - Off—no LAN connection
- **Activity**—Flashes Green for transmit (TX) or receive (RX) LAN data
- **Tx**—Flashes with transmit serial activity
- **Rx**—Flashes with receive serial activity
- **Downloading firmware**—(Green/Yellow) The Device Server will flash green/yellow, indicating that it is downloading new firmware.

Console Mode vs. Serial Mode: Desktop Models

You will notice a little switch at the back of the desktop Device Server models for switching the Device Server to either Console or Serial mode. Note that the Extended Temperature models have two switches, Switch 1 is used for Console mode and Switch 2 is unused.

When the switch is down (ON), the Device Server is in Console mode; when the switch is up, the Device Server is in Serial mode. Console mode is used when you have a direct connection between a serial device (like a terminal or a PC) and the Device Server, accessed by the Admin user to configure/manage the Device Server. You can connect directly to the Device Server in Serial mode, but the Device Server will not display all the messages/information you will get in Console mode. Console mode automatically sets the **Serial Interface** to **EIA-232**, **Speed** to **9600**, **Flow Control** to **No**, **Bits** to **8**, **Stop Bits** to **1**, and **Parity** to **None**, in addition to displaying extra system messages. Your Device Server **Line 1** will not work in a production environment in Console mode, because the Device Server ignores any **Line** settings when in Console mode. **NOTE:** When the Device Server is in Console mode, the Power/Ready LED will flash green.

Serial mode is used when the Device Server acts as a communications server, or anytime you are not connecting directly to the Device Server to configure it. On the 2-port/4-port desktop Device Server model, the Console port is Port 1.

Dedicated Console Port: Rack Mount Models

The rack mount Device Server models have a dedicated Console port, located on the side of the Device Server that displays the LEDs. You can configure the baud rate and flow control of the dedicated Console port. You can view diagnostic information when you are connected to the Console port.

Powering Up the Device Server

Serial Only Models

Before you attach the Device Server to your network or try to configure it, we suggest that you power it up to verify that it works properly. To power up the Device Server, perform the following steps:

1. Plug the external power supply into the Device Server and then into the electrical outlet or connect it to the PSE if you have a P series (power over Ethernet) model.
2. If the Device Server is working correctly, you should see the LEDs cycle for several seconds and then remain a solid green, indicating that it is ready to configure/use.

You are now ready to begin communicating with your IOLAN Device Server. The last step of the installation process is to set an IP address for the Device Server; this is necessary before it can be configured and put into production.

Before you start to configure the Device Server, you should set the desktop Device Server jumpers if you want to terminate the line or use the power in pin feature (instead of an external power supply, if your desktop Device Server model supports it).

I/O Models

Before you attach the Device Server to your network or try to configure it, we suggest that you power it up to verify that it works properly. To power up the Device Server, perform the following steps:

1. Unplug the power plugable terminal block from the Device Server.
2. Loosen the screws and then insert your positive (+) wire into the left terminal and screw it down. Insert the negative (-) wire into the right terminal and screw it down.
3. Plug the power terminal block back into the Device Server.
4. Plug the power supply into the electrical outlet.
5. If the Device Server is working correctly, you should see the LEDs cycle for several seconds and then remain a solid green, indicating that it is ready to configure/use.

You are now ready to connect your I/O peripherals to the Device Server and then begin communicating with your IOLAN Device Server. The last step of the installation process is to set an IP address for the Device Server; this is necessary before it can be configured and put into production.

Before you start to configure the Device Server, you should set the Device Server jumpers for Digital I/O (see ***Digital I/O Module*** on page 48) or Analog Input (***Analog Input Module*** on page 49) channels.

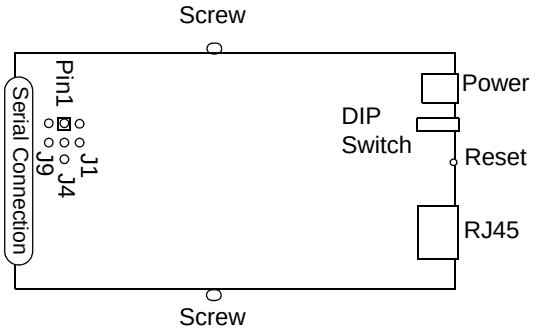
Setting Jumpers

The Device Server contains jumpers that you might need to set before you configure it and put it into production. You can set the power out pin, pin 9, to a fixed 5V DC output or to the external adapter output; this can range from 9-30V DC (if an external adapter is shipped with the Device Server, it has a 12V DC output). By default, the power out pin is set to no power. You can set the Device Server line termination to **`on`** or **`off`** (this is **`off`** by default) if you are using EIA-422/485 (not applicable for I/O models).

1-Port Device Server DB25 Male/Female

To change the settings, do the following:

1. Unplug the Device Server from the electrical outlet and disconnect everything from the box.
2. Open the case by unscrewing the two side screws, one on each side, and lifting off the top of the case. You should see the following:

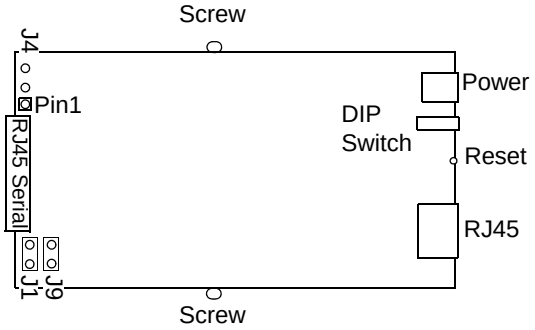


3. To change the power pin out, locate J4. For the fixed 5V DC output, jumper pins 1 and 2. For the output to equal the external adapter input, jumper pins 2 and 3.
4. To turn line termination **`on`**, locate and jumper both J1 and J9.
5. Close the Device Server case by replacing the case lid and the two screws. You can now power it on with the new settings.

1-Port Device Server RJ45

To change the settings, do the following:

1. Unplug the Device Server from the electrical outlet and disconnect everything from the box.
2. Open the case by unscrewing the two side screws, one on each side, and lifting off the top of the case. You should see the following:

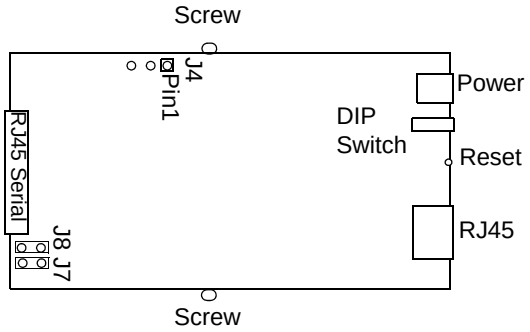


3. To change the power pin out, locate J4. For the fixed 5V DC output, jumper pins 1 and 2. For the output to equal the external adapter input, jumper pins 2 and 3.
4. To turn line termination **on**, locate and jumper both J1 and J9.
5. Close the Device Server case by replacing the case lid and the two screws. You can now power it on with the new settings.

1-Port Device Server RJ45 P (Power Over Ethernet)

To change the settings, do the following:

1. Unplug the Device Server from the electrical outlet and disconnect everything from the box.
2. Open the case by unscrewing the two side screws, one on each side, and lifting off the top of the case. You should see the following:

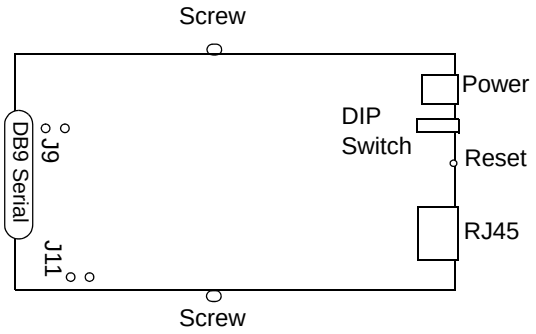


3. To change the power pin out, locate J4. For the fixed 5V DC output, jumper pins 1 and 2. For the output to equal the external adapter input, jumper pins 2 and 3.
4. To turn line termination **on**, locate and jumper both J7 and J8.
5. Close the Device Server case by replacing the case lid and the two screws. You can now power it on with the new settings.

1-Port Device Server DB9

To change the settings, do the following:

1. Unplug the Device Server from the electrical outlet and disconnect everything from the box.
2. Open the case by unscrewing the two side screws, one on each side, and lifting off the top of the case. You should see the following:

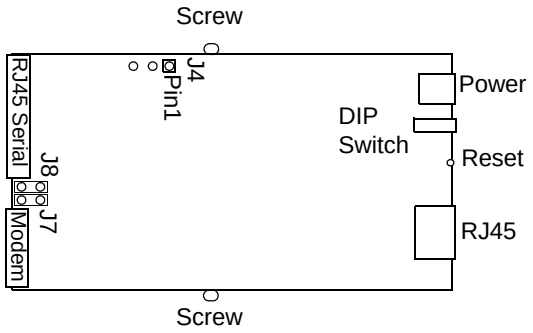


3. To turn line termination **on**, locate and jumper both J11 and J9.
4. Close the Device Server case by replacing the case lid and the two screws. You can now power it on with the new settings.

2-Port Device Server SDS1M (Modem)

To change the settings, do the following:

1. Unplug the Device Server from the electrical outlet and disconnect everything from the box.
2. Open the case by unscrewing the two side screws, one on each side, and lifting off the top of the case. You should see the following:

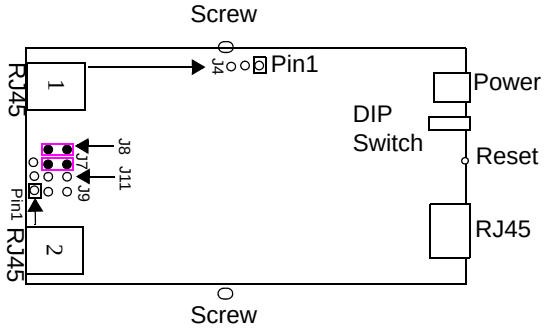


3. To change the power pin out, locate J4. For the fixed 5V DC output, jumper pins 1 and 2. For the output to equal the external adapter input, jumper pins 2 and 3.
4. To turn line termination **on**, locate and jumper both J7 and J8.
5. Close the Device Server case by replacing the case lid and the two screws. You can now power it on with the new settings.

2-Port Device Server

To change the settings, do the following:

1. Unplug the Device Server from the electrical outlet and disconnect everything from the box.
2. Open the case by unscrewing the two side screws, one on each side, and lifting off the top of the case. You should see the following:

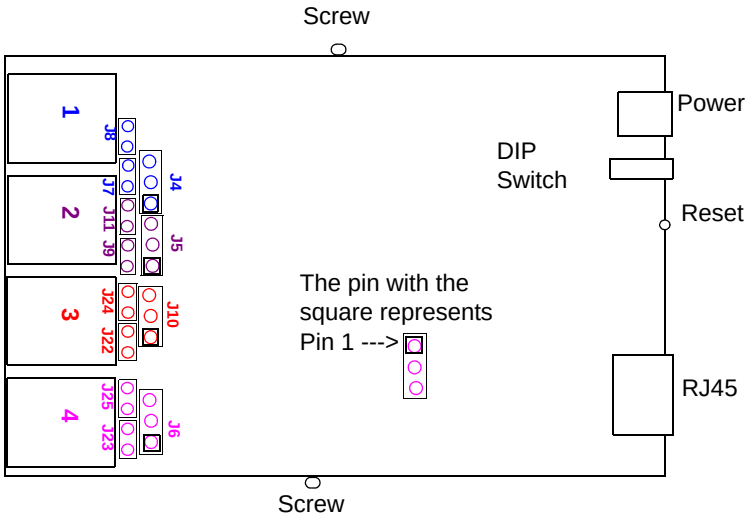


3. To change the power pin out, locate the set of three pins associated with the line you want to set (Line 1 is J4; Line 2 is the set the three pins just to the left of port 2). For the fixed 5V DC output, jumper pins 1 and 2. For the output to equal the external adapter input, jumper pins 2 and 3.
4. To turn line termination **on** for Line 1, locate and jumper both J7 and J8 (as shown in the diagram). To turn line termination **on** for Line 2, locate and jumper both J11 and J9.
5. Close the Device Server case by replacing the case lid and the two screws. You can now power it on with the new settings.

4-Port Desktop Device Server

To change the settings, do the following:

1. Unplug the Device Server from the electrical outlet and disconnect everything from the box.
2. Open the case by unscrewing the two side screws, one on each side, and lifting off the top of the case. You should see the following:



3. The following table describes how to jumper the pins for line termination, fixed 5V output, and for output equal to the external adapter input:

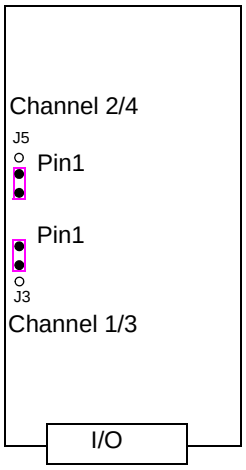
Port/Line #	Line Termination	5V Output	Input Volt Output
1	Jumper J7 and J8	J4, jumper pins 1 & 2	J4, jumper pins 2 & 3
2	Jumper J9 and J11	J5, jumper pins 1 & 2	J5, jumper pins 2 & 3
3	Jumper J22 and J24	J10, jumper pins 1 & 2	J10, jumper pins 2 & 3
4	Jumper J23 and J25	J6, jumper pins 1 & 2	J6, jumper pins 2 & 3

4. Close the Device Server case by replacing the case lid and the two screws. You can now power it on with the new settings.

Digital I/O Module

Device Servers that have Digital I/O have an input/output jumper that must be set for each channel and must match the software configuration for each channel. Depending on the model, the placement of the digital I/O board can change, so the diagram below shows how to set jumper for any digital board.To change the settings, do the following:

1. Detach the Device Server from the electrical power source and disconnect everything from the box.
2. Open the case by unscrewing the five side screws, two on each side plus the grounding screw, and lifting off the top of the case. You should see the following configuration for the digital I/O board:



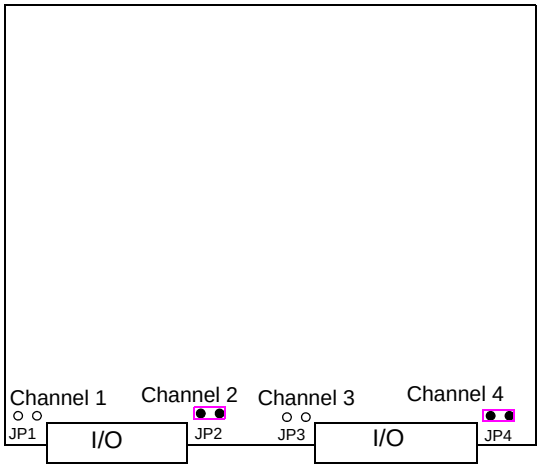
Note: Jumper pins 1 and 2 for Input. Jumper pins 2 and 3 for Output.

3. To configure either Channel 1 or Channel 3 (depending on how many Digital channels your I/O supports and following the mylar channel definitions) for Input, jumper J3 pin 1 and 2 (as shown); this is the default setting. To configure either Channel 2 or Channel 4 (depending on how many Digital channels your I/O supports and following the mylar channel definitions) for Output, jumper J5 pin 2 and 3 (as shown).
4. Close the Device Server case by replacing the case lid and the five screws. You can now power it on with the new settings.

Analog Input Module

Device Servers that have Analog Input have a voltage/current jumper that must be set for each channel and must match the software configuration for each channel. To change the settings, do the following:

1. Detach the Device Server from the electrical power source and disconnect everything from the box.
2. Open the case by unscrewing the five side screws, two on each side plus the grounding screw, and lifting off the top of the case. You should see the following configuration for the analog input board:



3. To configure Channel 1 for Voltage, no jumper should be set (as shown); this is the default setting. To configure Channel 2 for Current, jumper both J2 pins (as shown).
4. Close the Device Server case by replacing the case lid and the five screws. You can now power it on with the new settings.

Wiring I/O Diagrams

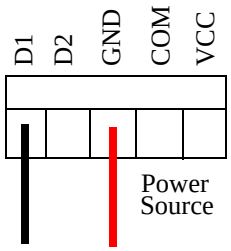
This section describes how to wire the various Device Server I/O models.

Digital I/O

Make sure the Digital I/O jumpers support the software setting; see ***Digital I/O Module*** on page 48 for jumper settings.

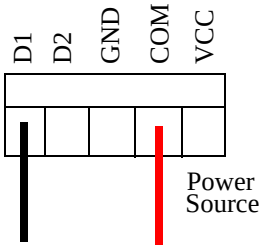
Digital Input Wet Contact

If you are using a wet contact for your Digital input, for channel D1 connect one wire to D1 and the other wire to GND. The power source is supplied by the GND (ground) connector.



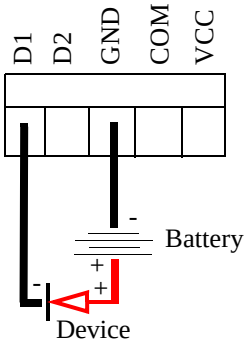
Digital Input Dry Contact

If you are using a dry contact for your Digital input, for channel D1 connect one wire to D1 and the other wire to COM. The power source is supplied by the COM (common) connector.



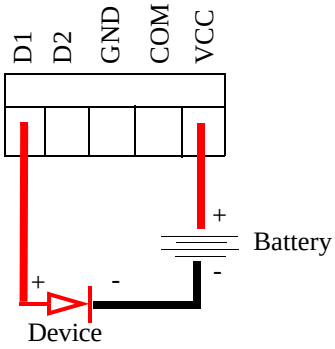
Digital Output Sink

For a Digital output sink (ground) configuration for channel D1, follow the diagram below.



Digital Output Source

For a Digital output source (voltage) configuration for channel D1, follow the diagram below.

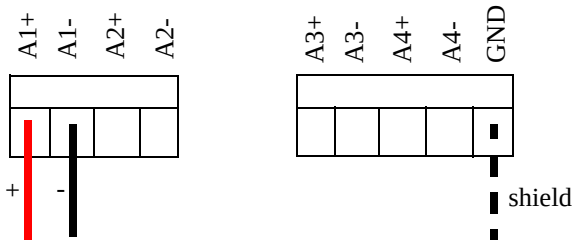


Analog Input

Make sure the Analog jumpers support the software setting; see ***Analog Input Module*** on page 49 for jumper settings.

Current

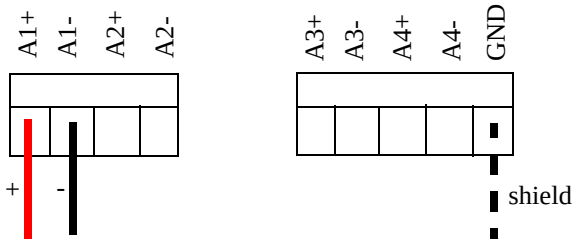
To connect channel A1 with a 2-wire shielded cable, connect the positive wire to A1+, the negative wire to A1-, and optionally the shield to GND.



If you have the positive/negative wires reversed, the output will always read 0 (zero).

Voltage

To connect to Channel A1 with a 2-wire shielded cable, connect the positive wire to A1+, the negative wire to A1-, and optionally the shield to GND.



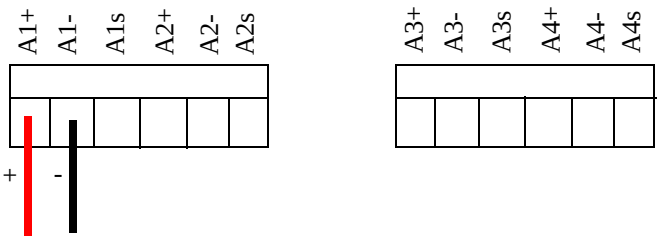
If you have the positive/negative wires reversed, the polarity of the voltage will be reversed.

Temperature Input

If you are using RTD sensors, a short detected status will be displayed if the wires are connected improperly. RTD or thermocouple sensors will display an open detection status when the circuit is broken.

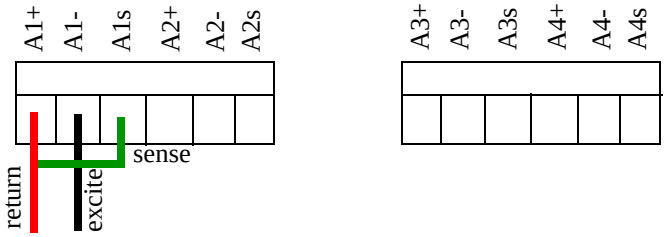
Thermocouple

To connect to Channel A1 with a 2-wire cable, connect the positive wire to A1+ and the negative wire to A1-; you will not be using the A1s connection.



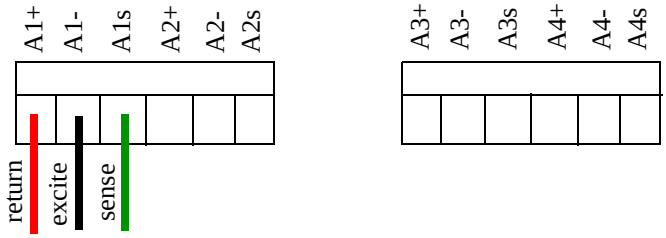
RTD 2-Wire

In a 2-wire RTD configuration, connect the excite wire to A1-, the return wire to A1+, and jumper the sense wire from A1s with a insulated wire going to A1+.



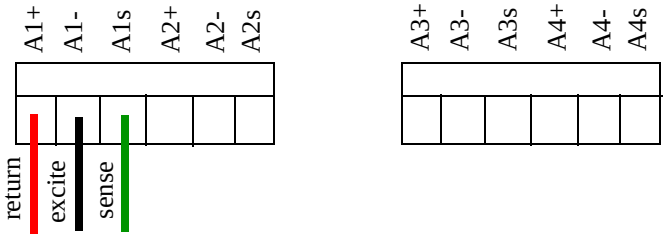
RTD 3-Wire

In a 3-wire RTD configuration, connect the return wire to A1+, the excite wire to A1-, and the sense wire to A1s.



RTD 4-Wire

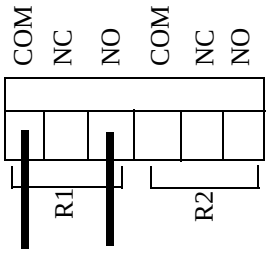
In a 4-wire RTD configuration, connect the return wire to A1+, the excite wire to A1-, the sense wire to A1s, and leave the fourth wire disconnected.



Relay Output

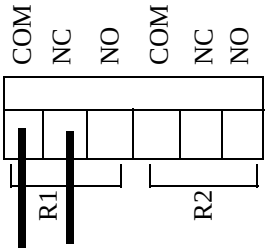
Normally Open Contact

To connect Relay channel R1 for a circuit that is normally inactive, connect one wire to the COM (common) connector and one wire to the NO (normally open) connector.



Normally Closed Contact

To connect relay channel R1 for a circuit that is normally active, connect one wire to the COM (common) connector and one wire to the NC (normally closed) connector.



Setting an Initial IP Address

This section describes the different methods you can use to set the Device Server IP address.

Following is a list of methods for setting the Device Server IP address and a short explanation of when you would want to use that method:

- **Easy Config Wizard**—The Easy Config Wizard is automatically launched from the CD ROM included with your **Device Server**. You can use the Easy Config Wizard to set the Device Server's IP address and configure the line(s).
- **DeviceManager**—Use this method when you can connect the Device Server to the network and access the Device Server from a Windows® PC. The DeviceManager is a Windows-based application that can be used for Device Server configuration and management.
- **Direct Connection**—Use this method when you can connect the Device Server directly to a dumb terminal, essentially logging directly into the Device Server. Using this method, you will need to configure and/or manage the Device Server using either the Menu or CLI.
- **DHCP/BOOTP**—Use this method when you have a BOOTP or DHCP server running and you can connect the Device Server to your network. The Device Server will automatically obtain an IP address from a local network DHCP/BOOTP server when this service is enabled (it is disabled by default).
- **ARP-Ping**—Use this method when you can connect the Device Server to the network and want to assign a temporary IP address to the Device Server by specifying an ARP entry and then pinging it.
- **IPv6 Network**—When the Device Server is connected to an IPv6 network, its local link address is determined using stateless auto configuration.

Note: Regardless of which method you use, the Device Server must reside within the same network as the host you are accessing it from.

Once an IP address has been assigned to the Device Server, in most cases, you can continue to use the same method to configure and/or manage the Device Server. See Chapter 3, ***Configuration Methods*** on page 67 for more information on the different methods you can use to manage/configure the Device Server.

Using DeviceManager

To use the DeviceManager, you must first install it on a Windows operating system (Windows NT requires Service Pack 4 or later). If the PC that is running DeviceManager resides in a different network than the Device Server, your network router must have multicast enabled in order for DeviceManager to be able to communicate to the Device Server; otherwise, both DeviceManager and the Device Server must reside in the same network. The DeviceManager installation wizard can be found on the CD-ROM included in the Device Server package.

1. Connect the Device Server to the LAN and plug it in; it will automatically boot up (rack mount models will need to be turned On).
2. From the CD-ROM that was included in the Device Server packaging, select the DeviceManager link.
3. Click on the link under **Location** and click **Open** to automatically start the DeviceManager installation.
4. Install the DeviceManager by following the installation wizard. On the last window, check the **Yes, I want to launch DeviceManager now.** box and click the **Finish** button.
5. On the **Manage Device Server** tab, click the **Search Local Network** button.
6. Any Device Server that does not have an IP address will be displayed as **Not Configured**, with the **Model** and **MAC Address** to identify the Device Server. Highlight the Device Server that you want to assign an IP address to and click the **Assign IP** button.
7. Choose the method you want to use to assign an IP address to the Device Server:
   - Type in the IP address that you want to assign to this Device Server
   - Enable the **Have the Device Server automatically get a temporary IP address** option. This will turn on DHCP/BOOTP, so the Device Server will attempt to get its IP address from your DHCP/BOOTP server. If you don't have a DHCP/BOOTP server, DeviceManager will temporarily assign an IP address in the range of **`169.254.0.1`**-**`169.254.255.255`** that will be used only for the duration of the DeviceManager/Device Server communication.

   Click the **Assign IP** button.
8. You are now ready to configure the Device Server. Double-click the Device Server you just configured IP address for to open a configuration session. Type **`superuser`** (the factory default Admin user password) in the Login window and click **OK**.
9. Expand the **Server Configuration** folder and select **Server**. You can choose a different method to assign the IP address to the Device Server. You should also enter a name in the **Server Name** field to make the Device Server easily identifiable.
10. Click the **Apply** button when you're done with the Server window. To make your edits take effect, you need to download the new configuration file and then reboot the Device Server.
11. Download the configuration file to the Device Server by selecting **Tools**, **Download Configuration to Unit**.
12. Reboot the Device Server by selecting **Tools**, **Reboot Server**.

For more information on configuring the Device Server using DeviceManager, see Chapter 5, ***Using the DeviceManager*** on page 135.

Using a Direct Connection

You can connect to the Device Server using a PC with a terminal emulation package, such as HyperTerminal or a terminal.

1. Connect the Device Server to your PC or dumb terminal. Make sure the DIP switch is in Console mode (desktop models, this sets the Device Server serial port to EIA-232) or that you are connected to the dedicated Console port (rack mount models). When connecting a terminal or PC directly (without modems), the EIA-232 signals need to be crossed over ('null modem' cable). See ***EIA-232 Cabling Diagrams*** on page 63 for cabling diagrams.
2. Using a PC emulation application, such as HyperTerminal, or from a dumb terminal, set the Port settings to 9600 Baud, 8 Data bits, No Parity, 1 Stop Bits, and No Hardware Flow control to connect to the Device Server. You can change these settings for future connections on the rack mount models (the Device Server must be rebooted for these changes to take place).
3. When prompted, type **`admin`** for the User and **`superuser`** for the Password. You should now see the a prompt that displays the model type and port number; for example, **`SCS16#`**.
4. You are now logged into the Device Server and can set the IP address by typing from the command line using the Command Line Interface (CLI).

   For single Ethernet connection models, type:

   ```
   set server internet <ipv4address>
   ```

   For dual Ethernet connection models, type:

   ```
   set server internet eth1 <ipv4address>
   ```

   Where ***`ipv4address`*** is the IP Address being assigned to the Device Server.
5. Type the following command:

   ```
   save
   ```
6. If you are going to use another configuration method, such as WebManager or DeviceManager, unplug a desktop Device Server or turn Off a rack mount Device Server. On a desktop Device Server, change the DIP switch to Off Serial (DIP switch in the up position) and connect it to your serial device. Plug the Device Server back in, automatically rebooting the Device Server in the process.
7. If you want to complete the configuration using a direct connection, see Chapter 3, ***Configuration Methods*** on page 67 and/or Chapter 7, ***Command Line Interface*** on page 243. After you complete configuring the Device Server, unplug the Device Server. Change the Device Server DIP switch to Off Serial (DIP switch in the up position) and connect it to your serial device. Plug the Device Server back in, automatically rebooting the Device Server in the process.

Using DHCP/BOOTP

If you are using BOOTP, you need to add an entry for the Device Server that associates the MAC address (found on the back of the Device Server) and the IP address that you want to assign to the Device Server. After you have made the MAC address/IP address association for BOOTP, use the following directions for BOOTP or DHCP.

You can connect to the Device Server using a PC with a terminal emulation package, such as HyperTerminal or a terminal.

1. Connect the Device Server to your PC or dumb terminal. Make sure the DIP switch is in Console mode (desktop models, this sets the Device Server serial port to EIA-232) or that you are connected to the dedicated Console port (rack mount models). When connecting a terminal or PC directly (without modems), the EIA-232 signals need to be crossed over ('null modem' cable). See ***EIA-232 Cabling Diagrams*** on page 63 for cabling diagrams.
2. Using a PC emulation application, such as HyperTerminal, or from a dumb terminal, set the Port settings to 9600 Baud, 8 Data bits, No Parity, 1 Stop Bits, and No Hardware Flow control to connect to the Device Server. You can change these settings for future connections on the rack mount models (the Device Server must be rebooted for these changes to take place).
3. When prompted, type **`admin`** for the User and **`superuser`** for the Password. You should now see the a prompt that displays the model type and port number; for example, **`SCS16#`**.
4. You are now logged into the Device Server and can set the IP address by typing from the command line using the Command Line Interface (CLI). Type the following command:

   ```
   set server dhcp/bootp on
   ```

   For dual Ethernet connection models, type:

   ```
   set server internet eth1 dhcp/bootp on
   ```

5. Type the following command:

   ```
   save
   ```

6. The the following command:

   ```
   reboot
   ```

7. When the Device Server reboots, it will automatically poll for an IP address from the DHCP/BOOTP server. If you have a Device Server with dual Ethernet, each Ethernet connection will automatically be assigned an IP address, you can access the Device Server through either IP address.

If for some reason it cannot obtain an IP address from your DHCP/BOOTP server, you will have to either connect to the Device Server on the console port and reboot it or push the Reset to Factory button to access the Device Server.

You are now ready to configure the Device Server. See Chapter 3, ***Configuration Methods*** on page 67 for information on the different Device Server configuration methods.

Using ARP-Ping

You can use the ARP-Ping (Address Resolution Protocol) method to temporarily assign an IP address and connect to your Device Server to assign a permanent IP address. To use ARP-Ping to temporarily assign an IP address:

1. From a local UNIX/Linux host, type the following at the system command shell prompt:

   ```
   arp -s a.b.c.d aa:bb:cc:dd:ee:ff
   ```

   On a Windows® 98 or newer system, type the following at the command prompt:

   ```
   arp -s a.b.c.d aa-bb-cc-dd-ee-ff
   ```

   (where **`a.b.c.d`** is the IPv4 address you want to temporarily assign to the Device Server, and **`aa:bb:cc:dd:ee:ff`** is the Ethernet (MAC) address of Device Server, found on the back of the unit.
2. Whether you use UNIX or Windows®, you are now ready to ping to the Device Server. Here is a UNIX example of the sequence to use:

   ```
   arp -s 192.168.209.8 00:80:d4:00:33:4e
   ping 192.168.209.8
   ```

You are now ready to configure the Device Server. See Chapter 3, ***Configuration Methods*** on page 67 for information on the different Device Server configuration methods.

IPv6 Network

The Device Server has a factory default link local IPv6 address that takes the following format:

Device Server MAC Address: 00-80-D4-AB-CD-EF

Link Local Address: fe80::0280:D4ff:feAB:CDEF

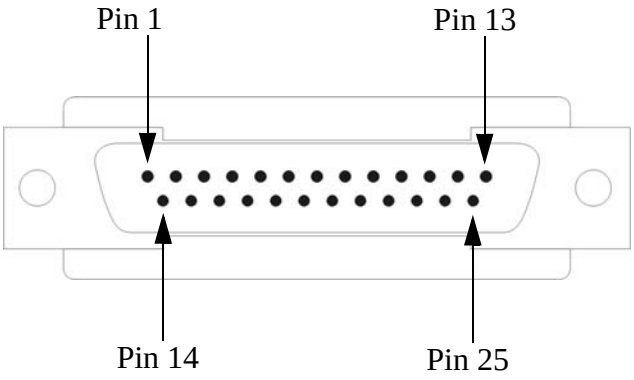
The Device Server will also listen for IPv6 router advertisements to learn a global address. You do not need to configure an IPv4 address for a Device Server residing in an IPv6 network.

You are now ready to configure the Device Server. See Chapter 3, ***Configuration Methods*** on page 67 for information on the different Device Server configuration methods.

Serial Pinouts

DB25 Male

This section defines the pinouts for the DB25 male connection used on the 1-port Device Server. The power out pin, Pin 9, is available in the SDS model only.



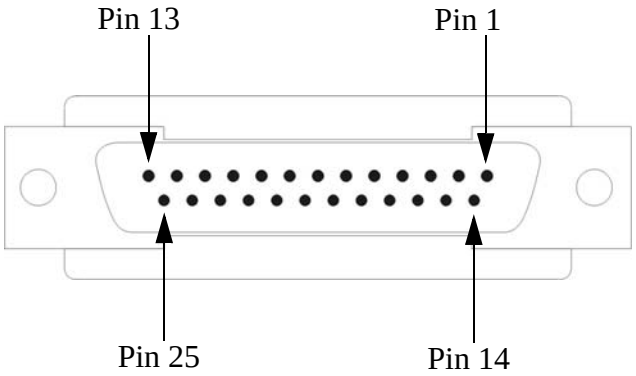
The following table provides pinout information:

Pinout	EIA-232	EIA-422	EIA-485 Full Duplex	EIA-485 Half Duplex
1	Shield	Shield	Shield	Shield
2 (out)	TxD			
3 (in)	RxD			
4 (out)	RTS			
5 (in)	CTS			
6 (in)	DSR			
7	GND	GND	GND	GND
8 (in)	DCD			
9	Power out	Power out	Power out	Power out
12	Power in	Power in	Power in	Power in
13		CTS-		
14		TxD+	TxD+	DATA+
15		TxD-	TxD-	DATA-
18		RTS+		
19		RTS-		
20 (out)	DTR			
21		RxD+	RxD+	
22		RxD-	RxD-	
25		CTS+		

The power in pin, pin 12, can be 9-30V DC.

DB25 Female

This section defines the pinouts for the DB25 female connection used on the 1-port Device Server. The power out pin, Pin 9, is available in the SDS model only.



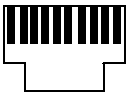
The following table provides pinout information:

Pinout	EIA-232	EIA-422	EIA-485 Full Duplex	EIA-485 Half Duplex
1	Shield	Shield	Shield	Shield
2 (in)	RxD			
3 (out)	TxD			
4 (in)	CTS			
5 (out)	RTS			
6 (out)	DTR			
7	GND	GND	GND	GND
8 (in)	DCD			
9	Power out	Power out	Power out	Power out
12	Power in	Power in	Power in	Power in
13		RTS-		
14		RxD+	RxD+	
15		RxD-	RxD-	
18		CTS+		
19		CTS-		
20 (in)	DSR			
21		TxD+	TxD+	DATA+
22		TxD-	TxD-	DATA-
25		RTS+		

The power in pin, pin 12, can be 9-30V DC.

RJ45

This section defines the pinouts for the RJ45 connection (see ***RJ45 SCS48C*** on page 60 for the SCS48C model). 1-port, 2-port, and 4-port desktop Device Server models have a 10-pin RJ45 connector and all rack mount Device Server models have an 8-pin RJ45 connector.

Pin 1  Pin 10

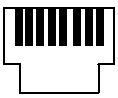
The following table provides pinout information:

Pinout 10-pin	Pinout 8-pin	EIA-232*	EIA-422	EIA-485 Full Duplex	EIA-485 Half Duplex
1		Power In	Power In	Power In	Power In
2 (in)	1	DCD			
3 (out)	2	RTS	TxD+	TxD+	DATA+
4 (in)	3	DSR			
5 (out)	4	TxD	TxD-	TxD-	DATA-
6 (in)	5	RxD	RxD+	RxD+	
7	6	GND	GND	GND	GND
8 (in)	7	CTS	RxD-	RxD-	
9 (out)	8	DTR			
10		Power out	Power out	Power out	Power out

The power in pin, Pin 1, can be 9-30V DC. The 2-port Device Server has power in on Port 2 only. The 4-port Device Server has power in on Port 4 only.

RJ45 SCS48C

This section defines the pinouts for the RJ45 connection for the SCS48C model only. The Admin port and the serial ports have different pinouts as shown in the table.

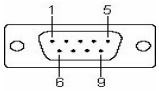
Pin 1  Pin 8

The following table provides pinout information:

Pinout 8-pin	EIA-232 Admin Port	EIA-232 Serial Ports
1	DCD (in)	RTS (out)
2	RTS (out)	DTR (out)
3	DSR (in)	TxD (out)
4	TxD (out)	GND
5	RxD (in)	GND
6	GND	RxD (in)
7	CTS (in)	DSR (in)
8	DTR (out)	CTS (in)

DB9 Male (Serial Only)

This section defines the pinouts for the DB9 male connection used on the 1-port Device Server that is serial only (not I/O).

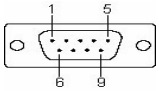


The following table provides pinout information:

Pinout 9-pin	EIA-232	EIA-422/485 Full Duplex	EIA-485 Half Duplex
1 (in)	DCD		
2 (in)	RxD	RxD+	
3 (out)	TxD	TxD+	TxD+/RxD+
4 (out)	DTR		
5	GND	GND	GND
6 (in)	DSR	RxD-	
7	RTS		
8 (in)	CTS		
9		TxD-	TxD-/RxD-

DB9 Male I/O

This section defines the pinouts for the DB9 male connection used on the 1-port Device Server I/O models.

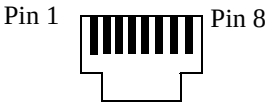


The following table provides pinout information:

Pinout 9-pin	EIA-232	EIA-422/485 Full Duplex	EIA-485 Half Duplex
1(in)	DCD		
2 (in)	RxD	RxD+	
3 (out)	TxD	TxD-	TxD-/RxD-
4 (out)	DTR		
5	GND	GND	GND
6 (in)	DSR	RxD-	
7	RTS	TxD+	TxD+/RxD+
8 (in)	CTS		
9			

Power Over Ethernet Pinouts

This section defines the pinouts for the RJ45 Ethernet connection used on the Device Server SDS P model.



The following table provides pinout information:

Pinout	Standard	802.3AF Unit-4 Wire	802.3AF Unit-8 Wire
1	Tx+	Tx+/+Voltage	Tx+
2	Tx-	Tx-/+Voltage	Tx-
3	Rx+	Rx+/-Voltage	Rx+
4	N/C		+Voltage
5	N/C		+Voltage
6	Rx-	Rx-/-Voltage	Rx-
7	N/C		-Voltage
8	N/C		-Voltage

EIA-232 Cabling Diagrams

This section shows how to create EIA-232 cables that are compatible with the Device Server.

Terminal DB25 Connector

The following diagrams show how the null modem cable should be configured when connecting to a terminal DB25.

DB25 Male

IOLAN DS1 DB25 (DTE)		Terminal DB25 (DTE)
2 (TxD)	———	3 (RxD)
3 (RxD)	———	2 (TxD)
4 (RTS)	———	5 (CTS)
5 (CTS)	———	4 (RTS)
6 (DSR)	———	20 (DTR)
7 (GND)	———	7 (GND)
20 (DTR)	———	6 (DSR)

DB25 Female

IOLAN DS1 DB25 (DCE)		Terminal DB25 (DTE)
3 (TxD)	———	3 (RxD)
2 (RxD)	———	2 (TxD)
5 (RTS)	———	5 (CTS)
4 (CTS)	———	4 (RTS)
20 (DSR)	———	20 (DTR)
7 (GND)	———	7 (GND)
6 (DTR)	———	6 (DSR)

RJ45

This cabling table does not apply to the SCS48C model.

IOLAN RJ45				Terminal DB25 (DTE)
10-pin		**8-pin**		
4	(DSR)	3	———	20 (DTR)
3	(RTS)	2	———	5 (CTS)
5	(TxD)	4	———	3 (RxD)
6	(RxD)	5	———	2 (TxD)
7	(GND)	6	———	7 (GND)
8	(CTS)	7	———	4 (RTS)
9	(DTR)	8	———	6 (DSR)

DB9 Male

IOLAN DS1 DB9 Male		Terminal DB25 (DTE)
3 (TxD)	———	3 (RxD)
2 (RxD)	———	2 (TxD)
7 (RTS)	———	5 (CTS)
8 (CTS)	———	4 (RTS)
6 (DSR)	———	20 (DTR)
5 (GND)	———	7 (GND)
4 (DTR)	———	6 (DSR)

Modem DB25 Connector

The following diagrams show how a standard straight through cable should be configured when connecting to a DB25 modem.

DB25 Male

IOLAN DS1 DB25 (DTE)		Modem DB25 (DCE)
2 (TxD)	———	2 (RxD)
3 (RxD)	———	3 (TxD)
4 (RTS)	———	4 (CTS)
5 (CTS)	———	5 (RTS)
6 (DSR)	———	6 (DSR)
7 (GND)	———	7 (GND)
8 (DCD)	———	8 (DCD)
20 (DTR)	———	20 (DTR)

RJ45

IOLAN RJ45				Modem DB25 (DCE)
10-pin		**8-pin**		
2	(DCD)	1	———	8 (DCD)
3	(RTS)	2	———	4 (CTS)
4	(DSR)	3	———	6 (DSR)
5	(TxD)	4	———	2 (RxD)
6	(RxD)	5	———	3 (TxD)
7	(GND)	6	———	7 (GND)
8	(CTS)	7	———	5 (RTS)
9	(DTR)	8	———	20 (DTR)

DB9 Male

IOLAN DS1 DB9 Male		Modem DB25 (DCE)
1 (DCD)	——	8 (DCD)
2 (RxD)	——	3 (TxD)
3 (TxD)	——	2 (RxD)
4 (DTR)	——	20 (DTR)
5 (GND)	——	7 (GND)
6 (DSR)	——	6 (DSR)
7 (RTS)	——	4 (CTS)
8 (CTS)	——	5 (RTS)

3 Configuration Methods

Introduction

This chapter provides information about the different methods you can use to configure the Device Server. Before you can configure the Device Server, you must assign an IP address to the Device Server. You can assign an IP address to the Device Server using one of the following methods:

- Using the DeviceManager as described in ***Using DeviceManager*** on page 54.
- Using ARP-Ping as described in ***Using ARP-Ping*** on page 57.
- Using a direct connection to the Admin port as described in ***Using a Direct Connection*** on page 55.

DeviceManager

The DeviceManager is a fully functional Windows Device Server configuration/management tool. You must install the DeviceManager from the CD-ROM included with the Device Server. Through the DeviceManager, you can:

- assign an IP address to new Device Servers.
- perform firmware updates.
- create configuration files, which can be immediately downloaded to the Device Server.
- save configuration files locally in the Device Server's native binary format or to a text file. The text configuration file can be edited with a text editor.
- open a session to a Device Server and import a (saved) configuration file.
- view statistics for a Device Server.
- download/upload keys/certificates to/from the Device Server.
- download custom files, such as new terminal definitions and a custom language file.
- download a configuration file to multiple Device Servers.

You can use the DeviceManager as a stand-alone application to create configuration files that can be saved locally or you can use the DeviceManager to open a session to a Device Server to actively manage and configure it.

See Chapter 5***, Using the DeviceManager*** on page 135 for information on configuring/managing the Device Server with DeviceManager.

WebManager

The WebManager is a web-browser based method of configuring/managing a Device Server for Admin users. Through EasyPort Web, all users can access clustered Device Servers, access all lines with reverse sessions and launch an SSH or Telnet connection to reverse session destination, and exercise power management capability (when using the Perle Remote Power Switch).

To access a Device Server through the WebManager, open up your web browser and type in the IP address of the Device Server that you want to manage/configure. A login screen will appear. Before you type in the Admin user password (the factory default password is **`superuser`**), select the **For a Secure Login Click Here** link if you are using the secure HTTP (HTTPS) mode (the **SSL Passphrase** must already be defined in the Device Server configuration and the SSL/TLS certificate/private key and CA list must have already been downloaded to the Device Server; see ***Keys and Certificates*** on page 98 for more information). If you are accessing the Device Server in non-secure HTTP, just type in the Admin password.

CLI

The Command Line Interface (CLI) is a command line option for Device Server configuration/management and user access. See Chapter 7***, Command Line Interface*** on page 243 for a full explanation of how to use the CLI.

If you are an existing IOLAN+ customer and would like to configure the Device Server in the native IOLAN+ CLI, you can type the command **`iolan+`** to use the native IOLAN+ CLI (you must have **User Level Normal** or higher). See your ***IOLAN User's Guide*** for information on using the IOLAN+ CLI. See ***IOLAN+ Interface*** on page 71 for more information about IOLAN+ interface.

Note: The IOLAN+ interface not supported on Device Server models with more than 16 ports or the DS1 model.

Menu

The Menu is a window-oriented Device Server configuration and user access option. To manage the Device Server, you will also need to use the CLI, WebManager, or DeviceManager, as you cannot download or upload files to the Device Server through the Menu.

If you are an existing IOLAN+ customer and would like to configure the Device Server in the native IOLAN+ menu interface, you can type the command **`iolan+`** to display and use the native IOLAN+ menu interface (you must have **User Level Normal**). See your ***IOLAN User's Guide*** for information on using the IOLAN+ interface. See ***IOLAN+ Interface*** on page 71 for more information about IOLAN+ interface.

Accessing the Menu

Menu access is available to any user whose **Line Service** is set to **DSLogin**, and whose **User Service** is set to **DSPrompt**. What the user sees depends on what the **User Level** is set to:

- **Menu**—Users with **User Level Menu** will only see the sessions that have been set up for them. They can start predefined sessions, kill (stop) a running session, resume a session, and logout of the Device Server.
- **Restricted**—Users with **User Level Restricted** can basically perform the same tasks as a Menu user, except that they have the option of performing these tasks via the Menu or the CLI.
- **Normal**—Users with **User Level Normal** can do everything a Restricted user can do, plus start a free session (connecting to any host on the network), set up their own user parameters (sessions, password, language, hotkey prefix), define their terminal, and become the Admin user (if they know the Admin password).
- **Admin**—Users with **User Level Admin** (not the Admin user), have complete access to the Device Server, the same as the Admin user. Through the Menu program, the Admin level user can configure the Device Server, although there are several tasks that can only be done in the CLI, such as downloading and uploading files and saving the configuration to FLASH.

Menu Conventions

You select an option from the Menu by using the keyboard up and down arrows to navigate the list. When the menu item you want to access is highlighted, press the **Enter** key to either get to the next list of options or to get the configuration screen, depending on what you select. When you are done configuring parameters in a screen, press the **Enter** key and then the **Enter** key again to **Accept and exit the form**. If you want to discard your changes, press the **Esc** key to exit a screen, at which point you will be prompted with **Changes will be lost, proceed? (y/n)**, type **y** to discard your changes or **n** to return to the screen so you can press **Enter** to submit your changes.

If there are a number of predefined options available for a field, you can scroll through those items by pressing the **Space Bar** or you can type **l** (lowercase L) to get a list of options, use the up/down arrows to highlight the option you want, and then press **Enter** to select it.

DHCP/BOOTP

If you have a DHCP/BOOTP server and the Device Server's Server Service DHCP/BOOTP is enabled, the Device Server can obtain its IP address and several configuration parameters from the DHCP/BOOTP server when it boots up. However, you must use another method for creating the configuration file, like the DeviceManager, WebManager, or the CLI. See ***DHCP/BOOTP Parameters*** on page 105 for more information on the DHCP/BOOTP parameters that can be set for the Device Server.

When DHCP/BOOTP is enabled and there is a DHCP/BOOTP server within the network, the IP Address obtained from DHCP/BOOTP will always override the Device Server's configured IP Address when the Device Server is rebooted.

SNMP

Before you can configure/manage the Device Server using SNMP, you need to set the Device Server IP address and configure a read-write user for SNMP version 3 or a community for SNMP version 1 or 2. You can use DeviceManager, CLI, or the Menu to set the IP address and user/community (don't forget to reboot the Device Server before connecting with the SNMP manager to make your changes take effect).

Required Support MIBs

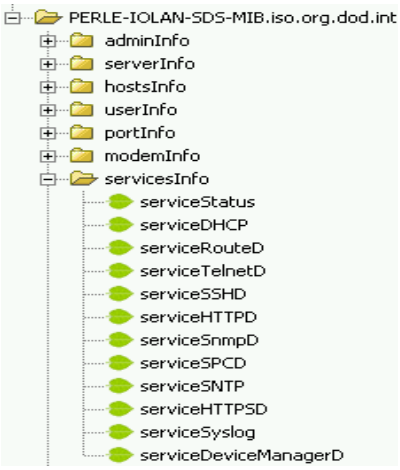
You need to have the following MIBs installed in your SNMP manager:

- SNMPv2-SMI
- SNMPv2-TC
- IPV6-TC

Configuring the Device Server Through the MIB

Once the IP address and user/community have been set, load the **`perle-sds.MIB`** file from the Device Server CD-ROM into your SNMP manager (this MIB works for all SDS, SCS, and STS models).

Connect to the Device Server through your SNMP manager using its IP address to configure/manage the Device Server. Expand the **PERLE-IOLAN-SDS-MIB** folder to see the Device Server's parameter folders. Below is an example of the configurable parameters under the **ServicesInfo** folder.



The first variable in each folder is the **Status** variable, for example, **serviceStatus**. When you perform a **`GET`** on this variable, one of the following values will be returned:

- **1**—Indicates that the container folder is active with no changes.
- **2**—Indicates that the container folder is active with change(s).

Once you have completed setting the variables in a folder, you will want to submit your changes to the Device Server. To do this, set the **Status** variable to **4**. If you want to discard the changes, set the **Status** variable to **6**.

- **4**—Indicates that the changes in the container folder are to be submitted to the Device Server.
- **6**—Indicates that the changes in the container folder are to be discarded.

If you want to save all the changes that have been submitted to the Device Server, you need to expand the **adminInfo** container folder and **`SET`** the **adminFunction** to **1** to write to FLASH. To make the configuration changes take effect, **`SET`** the **adminFunction** to **3** to reboot the Device Server.

IOLAN+ Interface

If you are an existing IOLAN+ user and would like to configure the Device Server using the IOLAN+ interface, you can type **`iolan+`** at the CLI command prompt to access the IOLAN+ configuration menu. The IOLAN+ interface is supported on all Device Server SDS, SCS, and STS models up to and including 16-ports.

Note: The Device Server and the IOLAN+ admin user share the same password. The default admin password is **`superuser`** (not **`iolan`**).

If you choose to use the IOLAN+ configuration interface, you should always configure the Device Server using the IOLAN+ interface, as fields do not map directly between the native Device Server interface and the IOLAN+ interface. Therefore, you could set a field parameter in one interface and unknowingly override a parameter (or several parameters) in the other interface. If you configure a field in the native Device Server configuration interface to a value that is invalid in the IOLAN+ interface and then attempt to use the IOLAN+ interface, the invalid field value will show up as ****** (all asterisks), although the Device Server will interpret the value as valid.

You should be aware that the following IOLAN+ configuration fields are not available in this implementation of the IOLAN+ interface:

- You no longer have the option of selecting **access**, **Authentication/Logging.** Also, **kill**, **reboot**, and **stats** are not available.
- When you select **port**, the following fields are not available on the Port Setup Menu:

```
** Administrator **          PORT SETUP MENU                          REMOTE-ADMIN
 Hardware                  Flow ctrl                      Keys
   Speed          [9600  ]   Flow ctrl    [None    ]      Hot   [^A]   Intr  [^C]
   Parity          [None ]   Input Flow   [Enabled ]      Quit  [^]]   Kill  [^\]
   Bit                [8]    Output Flow  [Enabled ]      Del   [^H]   Sess   N/A
   Stop              [1  ]                               Echo  [^E]
   Break        [Disabled]  IP Addresses
   Monitor DSR       [No ]   Src   [               ]      Mask  [               ]
   Monitor DCD       [No ]   Dst   [               ]
   Interface     [EIA-232]                              Access
 User                      Options                        Access        [Local  ]
   Name [abcd             ]   Keepalive        [No  ]      UDP Retries        N/A
   Terminal type [dumb    ]   Rlogin/Telnet       N/A      Retry Interval     N/A
   TERM   [               ]   Debug options       N/A    Authentication       N/A
   Video pages          [5]   Map CR to CR LF  [No  ]    Mode           [Raw   ]
   CLI/Menu         [CLI ]   Hex data            N/A    Connection  [None     ]
   Reset Term       [No  ]   Secure              N/A    Host  [               ]
                             MOTD             [Yes ]    Remote Port     [0    ]
                                                        Local Port      [10001]
_____________________________________________________________________________________
```

- User, Name—only when using LPD/LPR, Name no longer is used as the queue name
- Options, Rlogin/Telnet
- Options, Debug options
- Options, Hex data
- Options, Secure
- Keys, Sess
- Access, UDP Retries
- Access, Retry Interval
- Access, Authentication

- When you select **line**, **Access**, the following fields are not available on the Access Menu:

```
** Administrator **              ACCESS MENU                          REMOTE-ADMIN
 TTY Name                 Access    Authentication  Mode    UDP Retries  Interval
 1   [abcd              ] [Local  ] N/A             [Raw   ]      N/A       N/A
 2   [abcdef            ] [Local  ] N/A             [Raw   ]      N/A       N/A
```

  - Authentication
  - UDP Retries
  - Interval
- When you select **line**, **Options**, the following fields are not available on the Options Menu:

```
** Administrator **              OPTIONS MENU                         REMOTE-ADMIN
 TTY  Opt   CR     HEX    Rlogin/Telnet Keepalive
 1    N/A   [No  ] N/A    N/A           [No  ]
 2    N/A   [No  ] N/A    N/A           [No  ]
```

  - Opt
  - HEX
  - Rlogin/Telnet

- When you select **access**, **Remote access sites.**, the following fields are not available on the Remote Access Systems Screen:

```
** Administrator **     REMOTE ACCESS SYSTEMS SCREEN              REMOTE-ADMIN

 Sitename      [               ]
 User name     [               ]
 Password      [               ]

 Device type   (        )
 Service type  N/A
 Inactivity    N/A

 Phone number  [                       ]
 Login-script  N/A
```

  - Service type
  - Inactivity
  - Login-script
- When you select **access**, **Remote site devices.**, the following fields are not available on the Remote Site Device Screen:

```
** Administrator **      REMOTE SITE DEVICES SCREEN               REMOTE-ADMIN

 Type             [        ]

 IP Addresses
   Src Addr       N/A
   Dst Addr       N/A

 Modem
   Config         [                                                        ]
   Dial Comm      N/A
   Hang Up        N/A

 PPP Configuration            Dialer Configuration
   Restart timer  [3 ]          Dial Timeout  [45]
   Max Retries    [10]          Dial Retries  [2 ]

 Inactivity       [0    ]
```

  - IP Address, Src Address
  - IP Address, Dst Address
  - Modem, Dial Comm
  - Modem, Hang Up

When you select **server**, the following fields are not available on the Server Configuration menu:

```
** Administrator **         SERVER CONFIGURATION                    REMOTE-ADMIN

  Name               [wchiewsds2    ]                  Debug mode            N/A
  IP address         [172.16.22.7        ]
  Subnet mask        [255.255.0.0        ]
  Ethernet address   (00:80:d4:88:88:88)               Ethernet speed [AUTO    ]
  Language           [English  ]
  Identification     [                                    ]
  Lock               [Disabled]
  Password limit     [3 ]
  CR to initiate     N/A
  SNAP encoding      N/A
  Boot host          [                              ] Boot diagnostics      N/A
  Boot file          [                                                 ]
  Init file          [                                                 ]
  MOTD file          [                                                 ]
  Domain name        [                              ]
  Name server        [                   ]            NS Port               N/A
  WINS server        [                   ]
```

- Debug mode
- CR to initiate
- SNAP encoding
- Boot diagnostics
- NS Port

A new parameter was added, **Interface**, to the to Port Setup Menu, to specify whether you are setting up the serial line as a EIA-232 or EIA-422 line.

4 Configuring the Device Server

Introduction

This chapter provides general information about configuring the Device Server for your production environment. Although this chapter is not specific to any configuration method, there should be enough information that you can apply the information to any of the configuration methods.

When you are configuring the Device Server, remember that none of your configuration changes will be permanent until you submit/apply your changes, save to FLASH, and reboot the Device Server.

Configuring the Device Server

General Device Server Configuration

At this point, you should already have assigned the Device Server an IP address. Therefore, you have your choice of how to configure the Device Server by using the DeviceManager, WebManager, Menu, CLI, or SNMP.

Authentication

Authentication can be handled by the Device Server or through an external authentication server. Authentication is different from authorization, which can restrict a user's access to the network (although this can be done through the concept of creating sessions for a user, see ***Sessions*** on page 94 for more information on user sessions). All authentication does is ensure that the user is defined within the authentication database—with the exception of using the **Guest** authentication option under **Local Authentication**, which can accept any user ID as long as the user knows the configured password.

For external authentication, the Device Server supports RADIUS, Kerberos, LDAP, TACACS+, SecurID, and NIS. You can specify a primary authentication method and a secondary authentication method. If the primary authentication method fails (cannot connect to the server or authentication fails), the secondary authentication method is tried (unless you enable the **Only Use as backup** option, in which case the secondary authentication method will be tried only when the Device Server cannot communicate with the primary authentication host). This allows you to specify two different authentication methods. If you do specify two different authentication methods, the user will be prompted for his/her username once, but will be prompted for a password for each authentication method tried. For example, user Alfred's user ID is maintained in the secondary authentication database, therefore, he will be prompted for his password twice, because he is not in the primary authentication database.

Unlike the other external authentication methods, RADIUS and TACACS+ can also send back **Line** and **User** parameters that are used for the duration of the connection. Therefore, any parameters configured by RADIUS or TACACS+ will override the same parameters configured in the Device Server. See Appendix A, ***RADIUS*** on page 353 for RADIUS parameter information or Appendix B, ***TACACS+*** on page 361 for TACACS+ parameter information.

Device Server Services

In order to be as flexible and accessible as the Device Server is, it can run several predefined daemon and client applications. The Device Server can run the following daemon applications:

- TelnetD
- SPCD (the TruePort daemon)
- DeviceManagerD
- HTTPD
- HTTPSD
- SSHD
- SNMPD
- RouteD
- MODBUSD

If you disable any of the daemons, it can affect how the Device Server can be used or accessed. For example, if you disable HTTPSD and HTTPD, you will not be able to access the Device Server with the WebManager. If you disable DeviceManagerD, the DeviceManager will not be able to connect to the Device Server. If you do not want to allow users to Telnet to the Device Server, you can disable TelnetD; therefore, disabling daemons can also be used as an added security method for accessing the Device Server.

The following client applications can run on the Device Server:

- Syslog
- DHCP/BOOTP
- SNTP

If you do not have a DHCP/BOOTP server in your network, we recommend that you keep the DHCP/BOOTP service disabled to speed up Device Server reboots (otherwise, the Device Server waits for a DHCP/BOOTP packet until it times out, about a minute, on a reboot).

By default, all daemon and most client applications (except DHCP/BOOTP) are enabled and running on the Device Server.

TruePort

The TruePort utility acts as a COM port redirector that allows applications to talk to serial devices across a network as though the serial devices were directly attached to the server. For Device Server I/O models, you can also monitor and control I/O through the TruePort client. You can map the baud rate of the host COM port to a higher baud rate for the serial line that connects the serial device and the Device Server. You must be running the TruePort daemon on the host that is accessing the serial device for this to work. See ***TruePort*** on page 377 for more information about the TruePort utility.

Hardware Configuration

Configure the Ethernet interface that is connecting the Device Server to the LAN and the serial cable that is connecting the Device Server to the serial device.

Ethernet Connection

You need to know the Ethernet interface speed and duplex as follows, unless you are using the Auto detect option:

- 10 Mbps half or full duplex
- 100 Mbps half or full duplex
- 1000 Mbps half or full duplex (available on rack mount models only)

Serial Connection

You also need to know the serial interface specifications as follows (SCS and STS models support only EIA-232):

- EIA-232 and its speed
- EIA-422 and its speed
- EIA-485 and
  - its speed
  - half duplex with/without echo suppression or full duplex
  - TX driver control is automatic or RTS

Other

The most important thing to keep in mind when configuring the hardware parameters is to make sure that they are consistent with the serial device you have connected to the port. So, if you are connecting to a modem that sends out a DSR signal, you probably want to turn the **Monitor DSR** option on. Following is a list of just some of the other hardware configuration options:

- Data Bits—5 to 8
- Stop Bits—1, 1.5, 2 (1.5 not supported on all models)
- Monitor DSR—on, off
- Monitor DCD—on, off
- Parity—None, Odd, Even, Space, Mark
- Flow—Software, Hardware, or None (Hardware flow control is not supported by some configurations)

Port Buffering

The port buffering feature allows data activity on the Device Server's serial ports to be held in memory for viewing at a later stage without affecting the normal operation of the serial ports.

Port Buffering is required by system administrators to capture important information from devices attached to the Device Server. If a device (such as a Router) has a problem and sends a warning message out of its console port while no one is connected, the warning can be lost. With **Port Buffering** enabled, the messages will be captured in memory or in a file and can be viewed later to aid administrators in diagnosing and fixing problems.

Local Port Buffering

Port buffer information for the serial port can be viewed after successful connection to a device on a serial port. The user can toggle between communicating to the device on the serial port and viewing the port buffer data for that device by entering a configurable string (default **~view**). Note that local port buffers have a 256KB size and are flushed after a Device Server reboot.

To view the local port buffer for a particular serial port, you must connect to the device on that serial port by Telnet or SSH (the **Line Service** must be set to **Rev Telnet** or **Rev SSH**). Once you have established a connection to a device, you can enter the **View Port Buffer String** at any time to switch the display to the content of the port buffer for that particular serial port. To return to communicating to the device, press the **ESC** key and the communication session will continue from where you left off.

To navigate through the port buffer data, the following chart illustrates the keyboard keys or “hot keys” that can be used to view the port buffer data. Press the **ESC** key and to continue to communicate with the device on that particular serial port.

Keyboard	**Buttons Hot Keys**	**Direction**
Page Up	<CTRL>B	Up
Page Down	<CTRL>F	Down
Home	<CTRL>T	Top of the buffer data (oldest data)
End	<CTRL>E	Bottom of the buffer (latest data)
ESC		Exit viewing port buffer data.

Remote Port Buffers

The Device Server also supports Remote Port Buffering. The Remote Port Buffering feature allows data received from the serial lines on the Device Server to be sent to a remote server, supporting NFS (Network File System), for logging purposes. The data that is transmitted to the remote NFS server can be raw data or encrypted for security reasons. This feature only logs data from the serial line that is configured with **Line Service Rev Telnet** or **Rev SSH**. The Remote Port Buffering feature gives administrators the capability to analyse data and messages from the servers connected to the Device Server.

Remote Port Buffering data can encrypted and time stamped (configurable options) and is transmitted to an NFS server where a unique remote files is created using the Device Server’s configured **Line Name** for each line. If the **Line Name** is left at a default setting (blank), the Device Server will create unique files using the Device Server’s Ethernet MAC address and line number. It is recommended that a unique NFS directory and **Line Name** be configured if multiple Device Servers use the same NFS host for Remote Port Buffering. The filenames will be created on the NFS host with a **`.ENC`** extension to indicate data encrypted files or **`.DAT`** for unencrypted files. If the data is encrypted, the Decoder utility application, available on Windows (DOS/9x/NT/ME/2000/Server 2003/XP), SUN Solaris x86, SUN Solaris SPARC 64 and 32, Linux x86, can be run on the NFS server to convert the encrypted data to a readable file for administrators to analyze. NOTE: The Windows/DOS platform restricts the converted readable file to an 8.3 filename limitation.

The data that is sent to the remote buffer file is appended to the end of the file (even through Device Server reboots), so you will want to create a size limit on the file on your remote NFS host, to keep the buffer file size from becoming too large for your system.

Modbus Configuration

This sections provides a brief overview of the steps required to configure a Device Server for your Modbus environment. You can read the ***Modbus Gateway Settings*** on page 80 and ***Modbus Line Settings*** on page 81 sections for more specific information about the Modbus settings.

Overview

Configuring a Master Gateway

To configure a Master Gateway (Modbus Master resides on the serial side of the Device Server), do the following:

1. Verify that the default Modbus Gateway settings (the settings to the Slave Gateway do not apply here) in the Server section work in your environment; if they don't configure as required.
2. Set the **Line Service** parameter to **Modbus Master** for the Line connected to the Modbus serial Master.
3. In the Modbus Master settings, map the Modbus TCP Slave's IP addresses and their UIDs that the Modbus serial Master will attempt to communicate with.

Configuring a Slave Gateway

To configure a Slave Gateway (Modbus Master resides on the TCP/Ethernet network), do the following:

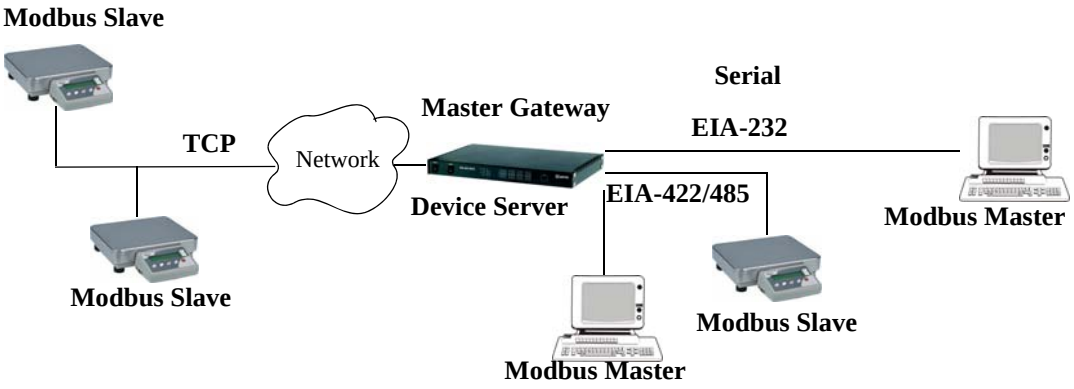
1. Verify that all the default Modbus Gateway settings in the Server section work in your environment; if they don't configure as required.
2. Set the **Line Service** parameter to **Modbus Slave** for the Line connected to the Modbus serial Slaves.
3. In the Modbus Slave settings, specify the Modbus Slave UIDs that the Modbus TCP Master will attempt to communicate with.

Modbus Gateway Settings

The scenarios in this section are used to illustrate how the Modbus Gateway settings are incorporated into a Modbus device environment. Depending on how your Modbus Master or Slave devices are distributed, the Device Server can act as both a Slave and Master Gateway(s) on a multiport Device Server or as either a Slave or Master Gateway on a single port Device Server.

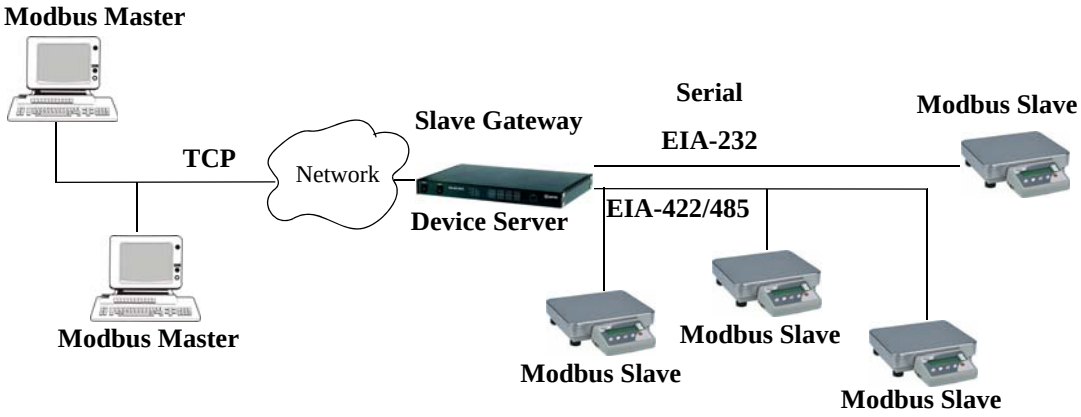
Modbus Master Gateway

The Device Server acts as a Master Gateway when the Modbus Master resides on the serial side of the Device Server. Each Modbus Master can communicate to UIDs 1-247.



Modbus Slave Gateway

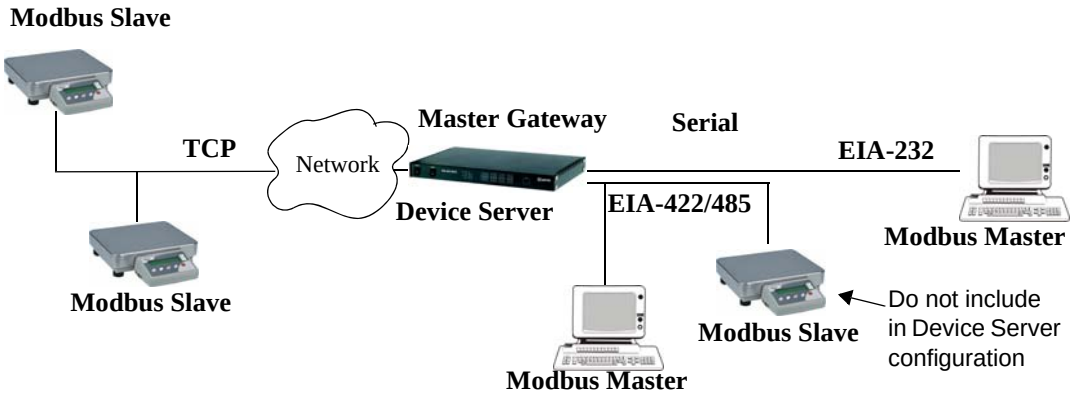
The Device Server acts as a Slave Gateway when the Modbus Master resides on the TCP/Ethernet network and the Modbus Slaves reside on the serial side of the Device Server. Note that there is only one Slave Gateway for the Device Server. You can define only one Slave Gateway for the Device Server, although multiple lines/ports can participate as part of that gateway (depending on how you configure the **Line Service** settings).



Modbus Line Settings

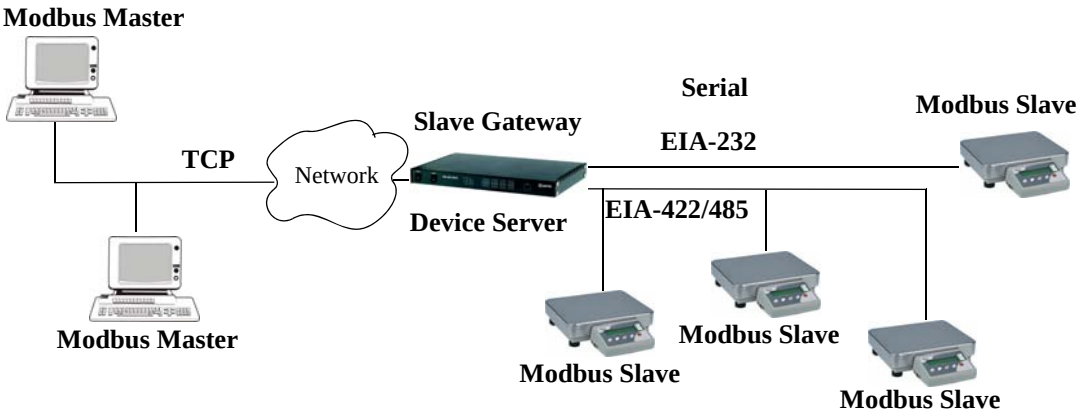
Modbus Master Settings

When you have Modbus Masters on the serial side of the Device Server, configure the Line as a Modbus Master. If you also have a Modbus serial Slave on the same serial network as the serial Modbus Master that communicates with that serial Master, do not define its UID in the Remote Slave IP Mappings settings, or the Modbus serial Slave may not function properly. You must configure the Modbus TCP Slaves (we term these as Remote Slave IP Mappings) on the TCP/Ethernet side so the Device Server can properly route messages to the appropriate UIDs configured for those remote Modbus TCP Slaves.



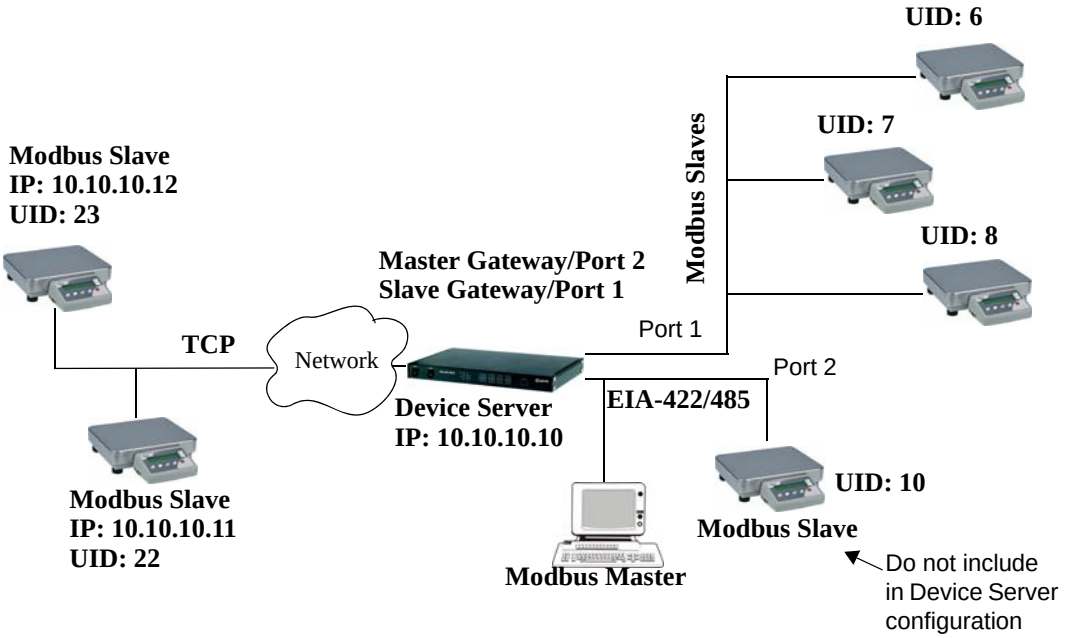
Modbus Slave Settings

When you have Modbus Slaves on the serial side of the Device Server, configure the Line as a Modbus Slave. There is only one Slave Gateway in the Device Server, so all Modbus serial Slaves must be configured uniquely for that one Slave Gateway; all Modbus serial Slaves must have unique UIDs, even if they reside on different serial ports, because they all must be configured to communicate through the one Slave Gateway.

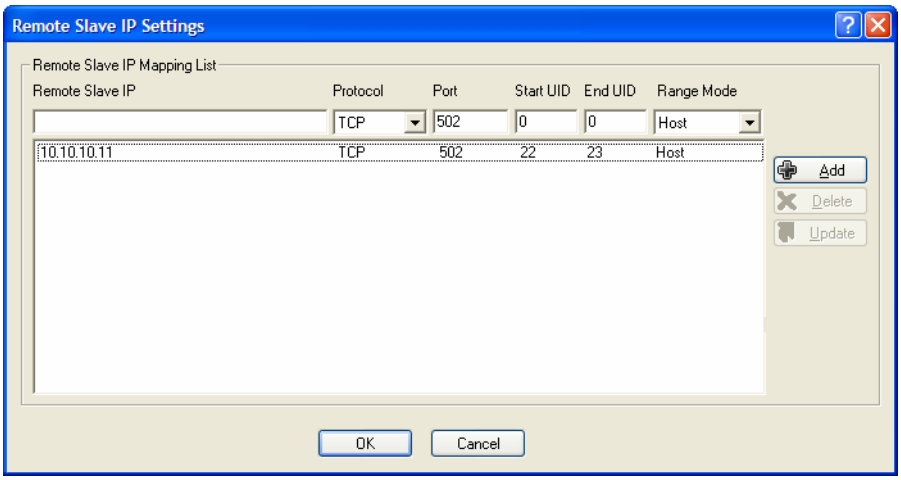


Example Scenario

The following example describes the settings that you would configure to set up a Modbus environment on a multiport Device Server, where the Modbus Master resides on a serial port/line connected to the Device Server. This scenario assumes two things, that the **Service ModbusD** (the Modbus daemon) is enabled and that the default **Modbus Gateway** settings have not been changed. The Modbus Master communicates with Modbus Slaves that reside on the TCP/Ethernet network and on another serial port defined as part of the Slave Gateway in the Device Server, and with a Modbus serial Slave (UID 10) that is on the same serial line as the Modbus Master itself. The Device Server will act as a Master Gateway for the Modbus serial Master and allow it to communicate to the remote Modbus TCP Slaves. By configuring the Device Server's own IP address as a remote Modbus Slave and having the Slave Gateway configured, the Modbus serial Master can communicate with Modbus serial Slaves on another serial port/line on the Device Server (UIDs 6-8).



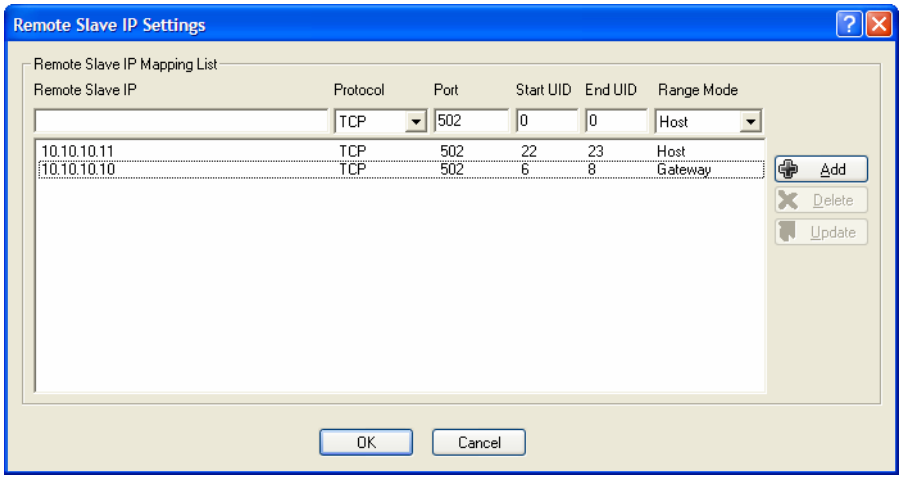
When the Modbus Master is communicating with the TCP/Ethernet Modbus Slaves, the Line/port that the Modbus Master is attached to must be configured with a **Line Service** of **Modbus Master**. By configuring the Remote Slave IP settings as:



The Device Server will send a request and expect a response from a Modbus Slave with an IP Address of 10.10.10.11 on Port 502 with UID 22 and from Modbus Slave with and IP Address of 10.10.10.12 on Port 502 with UID 23 (remember when **Range Mode** is set to **Host**, the Device Server increments the last octet of the IP address for each UID specified in the range).

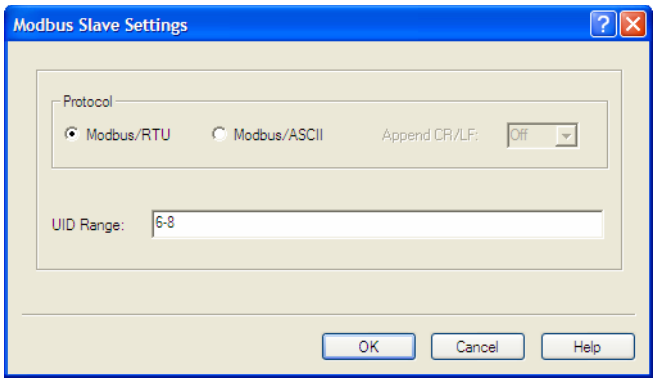
Also note that the Modbus Slave (UID 10) that is on the same line as the Modbus Master should not be configured anywhere in the Device Server's Master Gateway table for that serial port, or a Modbus Exception may be sent (if this option is enabled) because the Device Server will attempt to connect and send to a non-existent remote Modbus TCP Slave and a response timeout can occur.

To communicate with the Modbus Slaves on the serial side of the Device Server, the Device Server must also be configured to be a Slave Gateway. The Modbus Slaves on a serial port attached to the Device Server must be connected to a Line/port that is configured for the **Line Service** of **Modbus Slave**. To communicate with the Modbus Slaves on the serial port configured as part of a Slave Gateway, the Remote Slave IP settings are configured as:



The Device Server also acts as a Slave Modbus Gateway, receiving all the messages for IP address 10.10.10.10 and routing them to Modbus Slave devices with UIDs 6,7, and 8.

You must also configure the **Line Service** as **Modbus Slave** for the Modbus serial slaves as:



The Modbus serial Master will attempt to communicate through the Modbus Master Gateway to Modbus serial Slaves with UIDs 6, 7, and 8. In order to accomplish this, the communication is routed through the Device Server's Modbus Slave Gateway from the Device Server's Modbus Master Gateway, to the serial Slaves.

Email Notification

Email notification can be set at the Server and/or Line levels. You can set email notification at these levels because it is possible that the person who administers the Device Server might not be the same person who administers the serial device(s) attached to the Device Server port. Therefore, email notification can be sent to the proper person(s) responsible for the hardware.

Email notification requires an SMTP host that is accessible by the Device Server to process the email messages sent by the Device Server. When you enable email notification at the Server level, you can also use those settings for the Line, or you can configure email notification specifically for each Line. When you choose an event **Level**, you are selecting the lowest notification level; for example, if you select **Level Error**, you will get notifications for all events that trigger **Error**, **Critical**, **Alert**, and **Emergency** messages. The level order, from most inclusive to least inclusive, is as follows: Debug, Info, Notice, Warning, Error, Critical, Alert, Emergency.

The following events trigger an email notification on the **Server** for the specified **Level**:

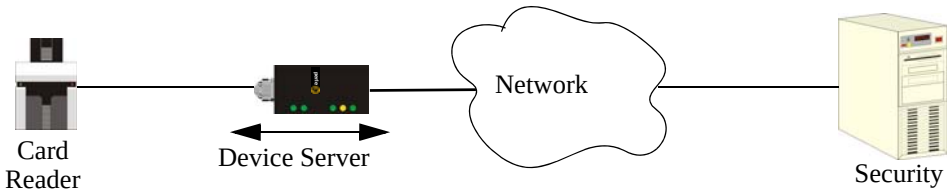
- Reboot, Alert Level
- Device Server Crash, Error Level
- Authentication Failure, Notice Level
- Successful Login, Downloads (all), Configuration Save Commands, Info Level

The following event triggers an email notification on the **Line** for the specified **Level**:

- DSR signal loss, Warning Level
- I/O alerts, Critical Level

Machine To Machine Connections

If you are using the Device Server to connect two hosts, allowing data to flow freely between them, you just need to configure the **Server** and the **Line** (no **User** required). In the following example, the serial device is a security Card Reader that needs to transmit and receive information to/from a host on the network that maintains the Card Reader's application every time an employee uses an access card to attempt to gain entry to the company.



After configuring the **Server** parameters (**Server Name**, **IP Address**, **Ethernet** and **Serial** interfaces, etc.), the **Line Service** is set to **Sil Raw**, which creates an automatic, continuous connection between the Card Reader and its associated application on the Security host (though the Device Server), by specifying the Security host name (which must already be configured in the Device Server's Host Table) and TCP/IP port number. Therefore, the Card Reader can make a request to the Security host card reader application for employee verification, also logging access time, employee name, etc., and the Security host application can send back a code that does or does not unlock the door.

Users Connecting to Serial Devices

For a user to connect to the serial device connected to the Device Server from the LAN, the **Line Service** must be set to **Rev Telnet** or **Rev SSH**. The user will either access the serial device directly or go through the Easy Port Access Menu, depending on the **User Level** setting.

Users who are **Level Admin** or **Normal** will access the serial device directly; the user must connect to the Device Server's IP address and port number (the **DS Port** parameter). The user will be asked to login with a user name and password; if this is successful, the user is automatically connected to the serial device.

Users who are **Level Restricted** or **Menu** can access the serial device through the Easy Port Access Menu, which displays the line number and name and a logout option; the user just needs to connect to the Device Server's IP address. The user will be asked to login with a user name and password; if this is successful, the Easy Port Access Menu is displayed. When a Menu-level user connects to the Device Server using SSH, the Easy Port Access Menu will display only those lines that have been configured for Reverse SSH. Similarly, if a Menu-level user connects using Telnet, the Easy Port Access Menu will display only those lines that have been configured for Reverse Telnet. If the Menu-level user connects using a protocol that is not configured on any of the Device Server's lines, nothing but **`Logout`** will be displayed on the Easy Port Access Menu; the connection protocol and the Line protocol must match.

Users Connecting to the LAN

For a user to connect to the LAN through the Device Server from a serial device, the **Line Service** can be set to any **Direct** or **Silent** setting, plus **PPP**, **SLIP**, **Bidir**, or **DSLogin**.

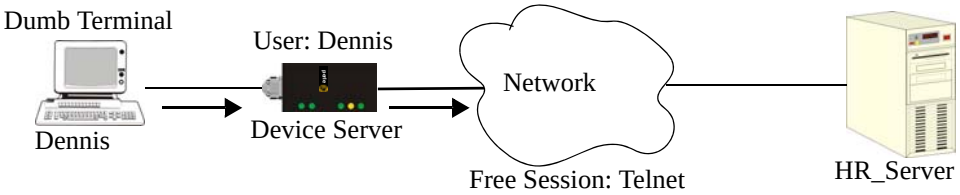
User accounts should be created when:

- authentication is being done locally by the Device Server.
- authentication is being done by an external authentication method, but there are settings that you would like to 'pick up' from the local user configuration. If you use RADIUS or TACACS+, RADIUS/TACACS+ parameters overwrite **User** parameters, which overwrite **Line** parameters.
- you want to create predefined sessions for a user to limit that user's access to the network.
- you have a user with a special use requirement, like a callback requirement.

Users can log into the Device Server without having a **User** set up, when external authentication is being done. In this case, an externally authenticated user would inherit the **Default User** configuration while logged into the Device Server.

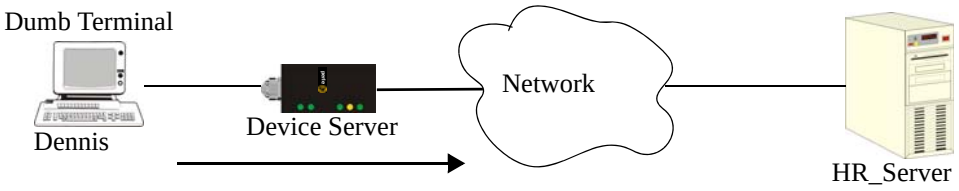
Connecting To the Device Server

When a user connects to the Device Server, that user can be authenticated either locally or externally and is usually set up with predefined sessions or given the opportunity to configure a **Free Session** to access any host using any protocol (must have a **Level** of at least **Normal** to configure a **Free Session**). In this example, the user must have a **Line** and **User Service** of **DSLogin** and **DSPrompt**, respectively. So, user Dennis is authenticated by the Device Server (either locally or externally) and then chooses to configure a **Free Session** to the HR_Server using the Telnet protocol (Dennis could have attempted to access any host on the network).



Connecting Through the Device Server

When a user connects through the Device Server, that user can be authenticated either locally or externally and is usually set up with a **User Service** that, once authentication is completed successfully, passes the user onto the specified host. Therefore, the **Line Service** is set to **DSLogin** and the **User Service** is set to whatever protocol the user will use to access the host; in this example, the **User Service** is set to **Telnet**. When **User Service Telnet** is selected, the IP address of the HR_Server is specified as the target Host IP. User Dennis will always have to log into the same server with this configuration.



Setting Up Lines

Lines and ports are often used interchangeably. They are almost the same, that is, each line has an associated port number (Line 1 starts with port 10001 by default), so port buffering settings are the same as the buffering settings for the line.

How you set up a line is really determined by the device that is connected to the line. This section goes over some of the common ways a line is used and things that you will want to keep in mind when configuring the line.

DSLogin

When you configure the **Line** for **DSLogin**, users connecting to the Device Server will have to go through some form of authentication, either local or remote authentication. Regardless of whether a user has been configured in the User table (local authentication) or is inheriting the Default User's attributes (remote or Guest authentication), when a **User Service** is selected (other than **DSprompt**), that connection (**Telnet**, **Rlogin**, **SSH**, **SLIP**, or **PPP**) will inherit the connection settings defined for **DSLogin**.

Direct/Silent/Reverse Connections

Direct connections bypass the Device Server, enabling the user to log straight into a specific host. A direct connection is recommended where a user logging in to the Device Server is not required. It is also recommended where multiple sessions are not a requirement. Direct connections require user interaction: the message **`Press return to continue`** is displayed on the user's screen and the session to the host is not initiated until **Enter** is pressed, after which the host login prompt is displayed. The message is redisplayed on logout.

Silent connections are the same as direct connections except that they are permanently established. The host login prompt is displayed on the screen. Logging out redisplays this prompt. Silent connections, unlike direct connections, however, make permanent use of pseudo tty (system) resources and therefore consume host resources even when not in use.

Reverse connections enable a host on the network to establish a connection to a serial device through the Device Server port.

Virtual Modems

Vmodem is a feature of the Device Server that provides “modem like” communication between two Device Servers on a network or between a Device Server and a host. This feature behaves like two modems connected across a telephone line. Typically, you use the **Vmodem** feature when you have multiple devices communicating with a central site. With just a single IOLAN Device Server at each end of the network, you don’t need to use multiple modems, avoiding the associated costs of calls and connections.

The data is sent in raw format from the virtual modem and can be received by another Device Server or a host. This data can be sent automatically using the **Monitor DSR** option and then configuring the host and port number of the receiver; if the receiving side is also a Device Server, set the **Line Service** to **Rev Raw** or **Vmodem** (**Rev Raw** if the Device Server is only receiving, **Vmodem** to initiate bidirectional data flow) and the Device Server port that the data is coming in on (this should match the port number on the sending Device Server). Or, you can manually start a connection by typing **ATD<***ip_address***>,<***port_number***>** and end the connection by typing **+++ATH**. The ***ip_address*** can be in IPv4 or IPv6 formats and is the IP address of the receiver. For example, **ATD123.34.23.43,10001** or you can use **ATD12303402304310001**, without any punctuation (although you do need to add zeros where there are not three digits presents, so that the IP address is 12 digits long).

VModem Initialisation Commands

Note: VModem initialization commands are only supported on Device Server firmware and configurators version 3.2 or higher.

You can initialize the modem connection using any of the following commands:

Command	Description	Options
ATQn	Quite mode. Determines if result codes will be sent to the connected terminal. Basic results codes are OK, CONNECT, RING, NO CARRIER, and ERROR. Setting quite mode also suppresses the "RING" message for incoming calls.	n=0, no result codes will be sent. n=1, result codes will be sent. (default)
ATVn	Verbose mode. Determines if result codes are displayed as text or numeric values.	n=0, display as numeric values. n=1, display as text. (default)
ATEn	Echo mode. Determines whether characters sent from the serial device will be echoed back by the Device Server when VModem is in "command" mode.	n=0, disable echo. n=1, enable echo. (default)
+++ATH	Hang up. This command instructs the Device Server to terminate the current session and go into "command" mode.	
ATA	Answer call. Instructs the VModem to accept connection requests. VModem will give the terminal up to 3 minutes to answer the call. If the ATA is not received within 3 minutes, all pending sync messages will be discarded.	
ATI0	Return the modem manufacturer name.	
ATI3	Return the modem model name.	

Command	Description	Options
ATS0	Sets the value of the S0 register. The S0 register controls the "auto answer" behaviour. In "manual" mode, the Device Server will not accept incoming sessions until an ATA is issued by the serial device. In "auto answer" mode, the Device Server will automatically accept an incoming connection request.	Register=0, sets "manual answer" mode Register=1-255, "auto answer" mode (default)
AT&Z1	Set command allows the user to store an IP address and port number or phone number to use when making a connection. The user will issue an ATDS1 to cause the Device Server to initiate the connection.	
AT&Sn	Sets the behaviour of Device Server's DTR signal. (DSR from a DCE perspective)	n=0, DTR signal always high. (default) n=2, DTR signal acts as DCD. n=3, DTR signal acts as RI.
AT&Rn	Sets the behaviour of Device Server's RTS signal. (CTS from a DCE perspective) If line is configured for hardware flow control, the RTS is used for this purpose and the setting of this command is ignored.	n=0, RTS always high. (default). n=3, RTS signal acts as DCD. n=4, RTS signal acts as RI.
AT&Cn	Sets the behaviour of the DCD signal.	n=0, DCD always on. n=1, DCD follows state of connection (off when no connection, on when TCP connection exists). (default)
AT&F	Sets the modes back to the factory defaults. This is a hard-coded default configuration which does not look at any user configuration.	
ATS2	Sets the value of the S2 register. The S2 register controls which character is used to enter "command" mode. (this is the potential replacement for the +++ (default) in front of the ATH command). This register will hold the hex value of the "escape" character. Any value > 27 will disable the ability to escape into "command" mode.	
ATS12	Sets the value of the S12 register. The S12 register controls the minimum length of idle time which must elapse between the receipt of the escape character and the A (first character of the ATH sequence). Units are 1/50th of a second. The default is 50 = 1 second.	
ATO	(ATD with no phone number) Establishes a connection using the IP and port specified in the telephone number field.	
ATDS1	Establishes a connection using the IP and port (or phone number) specified in the **Phone Number** field (stored by the AT&Z1 command).	

BIDIR

When you configure **BIDIR**, you are creating a bidirectional raw connection, meaning that the connection can be initiated from either the Ethernet or serial side. The Device Server initiates TCP connections to the configured host and port and listens for TCP connections on the **DS Port** configured for the **Line**.

TruePort

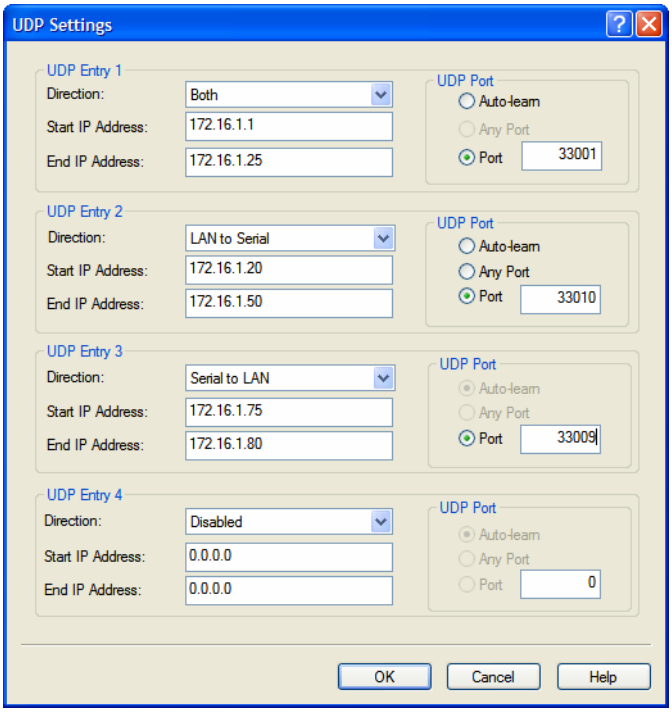
When you configure a line for **TruePort**, the Device Server provides a complete COM port interface between the attached serial device and the network. You can also set the **Client Initiated** option, which allows either the client or the Device Server to initiate communication. See ***TruePort*** on page 377 and the TruePort documentation for your operating system more information.

Signal I/O

When you configure a line for **Signal I/O**, you are using the DSR, DCD, CTS, DTR, and RTS serial pins for I/O channel digital input (DSR, DCD, and CTS) or digital output (DTR and RTS). Only after **Signal I/O** is specified as the **Line Service** can you configure the serial pins for I/O.

UDP

When you configure **UDP**, you are setting up a range of IP addresses and a port number that you will use to send UDP data to or receive UDP data from. For example:



The UDP configuration window, taken from the DeviceManager, is configured to:

- **UDP Entry 1**

  All hosts that have an IP address that falls within the range of **`172.16.1.1`** to **`172.16.1.25`** and listen to **Port `33001`** will receive UDP data from the serial device. The serial device will only receive UDP data from the hosts in that range with a source **Port** of **`33001`**. The Device Server will listen on the port value configured in the **DS Port** parameter.

- **UDP Entry 2**

  All UDP data received from hosts that have an IP address that falls within the range of **`172.16.1.20`** to **`172.16.1.50`** and **Port `33010`** will be sent to the serial device. The Device Server will not send any data received on its serial port.

- **UDP Entry 3**

  All hosts that have an IP Address that falls within the range of **`172.16.1.75`** to **`172.16.1.80`** and who listen to **Port `33009`** will receive UDP data from the serial device. The Device Server will listen for messages on the port value configured in the **DS Port** parameter. No UDP data will be sent to the serial device.

- **UDP Entry 4**

  This entry is disabled since **Direction** is set to **Disabled**.

PPP Dial On Demand

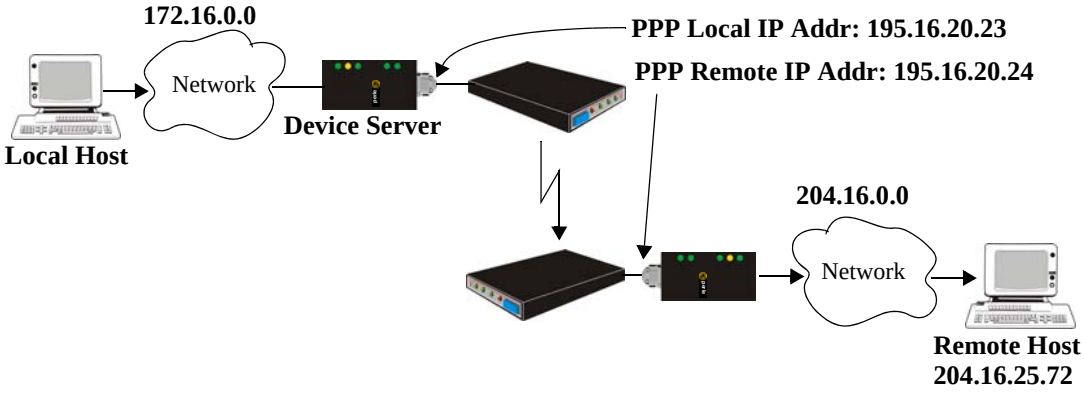
If you want to configure a line to use PPP dial on demand, do the following:

1. Create an entry for the modem and its initialisation string.
2. Set the **Line Service** to **PPP**.
3. Set the **Line Dial** parameter to **Out**, enter the **Phone Number** that the modem will be calling, and set the **Modem** parameter to the modem you just added.
4. Set the **Line Idle Timer** to a value that is *not* zero (setting this value to zero creates a permanent connection).
5. In the PPP configuration, enter either a **Local** and/or **Remote IPv4 Address** or a **Local** and/or **Remote IPv6 Interface Identifier** and create a **Host** entry for either IP address/interface identifier. Note that this IP address or interface identifier should be on its own unique network; that is, not part of the local or remote networks.

   In the example below, the local network has an IP address of 172.16.0.0/16 and the remote network has an IP address of 204.16.0.0/16, so we arbitrarily assigned the **PPP Local IP Address** as **`195.16.20.23`** and the **PPP Remote IP Address** as **`195.16.20.24`**. We also created a **Host** entry, **`PPP_GW`** with **IP Address `195.16.20.23`** (the same as the **PPP Local IP Address**).
6. Create a **Gateway** with **Service** as **Network** or **Host** with the host entry you just created. If you want the connection to be able to reach any host is the remote network, set the **Service** to **Network** and specify the network IP address and subnet/prefix bits; if you want the connection to go directly to a specific remote host, set the **Service** to **Host** and specify the host's IP address.

   In the example below, we created a **Gateway** entry using **Host `PPP_GW`**, assigned the **Service** as **Host** (meaning that the connection will automatically go to a remote host), and provided the **Destination Address** as **`204.16.25.72`**.

Any traffic that goes through the gateway will automatically cause PPP to dial out.



Printers

Remote Printing Using LPD

When setting up a serial line that access a printer using LPD, do the following:

1. Set the **Line Service** to **Printer** and configure the **Speed**, **Flow Control**, **Stop Bits**, **Parity**, and **Bits** parameters so that they match the printer's port settings.
2. Save your settings and kill the line.
3. Verify that LPD has been configured on the network host. To configure LPD on the network host, you need to know the name or IP address of the Device Server and the print queue, either **raw_p<*port_number*>** for a raw data connection or **ascii_p<*portnumber*>** for an ASCII character connection. You can optionally append **_d** or **_f** to the queue name to add a **<control d>** or **<form feed>** to the end of the print job.

Remote Printing Using RCP

When setting up a serial line that accesses a printer using RCP, do the following:

1. Set the **Line Service** to **Printer** and configure the **Speed**, **Flow Control**, **Stop Bits**, **Parity**, and **Bits** parameters so that they match the printer's port settings.
2. Save your settings and kill the line.
3. To execute a print job, use the following syntax:

   `rcp filename/ip_address DeviceServerName:p<#>`

   where **<#>** is the Device Server line port number (**DS Port**).

Remote Printing Using Host-Based Print Handling Software

Printers connected to the Device Server can be accessed by TCP/IP hosts using print handling software.

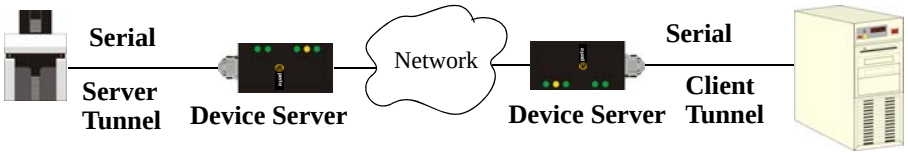
1. Set the **Line Service** to **Rev Raw** and configure the **Speed**, **Flow Control**, **Stop Bits**, **Parity**, and **Bits** parameters so that they match the printer's port settings.
2. Save your settings and kill the line.
3. The print handling software needs to know the **Name** of the Device Server and the **DS Port** number assigned to the printer port.

SSL/TLS

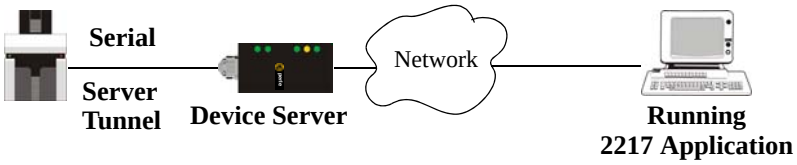
You can create an encrypted connection using SSL/TLS for any of the raw data line options: any **Raw** settings, **BIDIR**, or **VModem**. You can set up the Device Server to act as an SSL/TLS client or server. There is an extensive selection of SSL/TLS ciphers that you can configure for your SSL/TLS connection; see ***Modbus Gateway Settings*** on page 80 for a list of SSL/TLS ciphers. You can also enable peer certificate validation, for which you must supply the validation criteria that was used when creating the peer certificate (this is case sensitive, so keep that in mind when enabling this option).

Serial Tunnel Settings

The purpose of the serial **Line Service Client/Server Tunnel** is to allow two Device Servers that are connected back-to-back over Ethernet to virtually link two serial ports, based on RFC 2217. The serial device that initiates the connection is the **Client Tunnel** and the recipient is the **Server Tunnel**, although once the serial communication tunnel has been successfully established, the tunnel will stay connected and communication can go both ways.



The **Server Tunnel** will also support Telnet Com Port Control protocol as detailed in RFC 2217.



The port signals will also follow the signals on the other port. If one port receives DSR then it will raise DTR on the other serial port. If one port receives CTS then it will raise RTS on the other port. The CD signal is ignored.

Setting Up Users

You can create up to four users, in addition to the Admin user (who cannot be deleted) on all desktop Device Server models. On rack mount models, you can create up to 48 users, in addition to the Admin user (who cannot be deleted).

A user can even represent a device, like a barcode or a card swipe device, that you want to be authenticated.

User Accounts

When a serial device (like a dumb terminal or a barcode reader) is trying to access a host through the Device Server, you can configure user accounts when users:

- are authenticated by the Device Server (either locally or by an external authentication server) and then connect to a network host.
- want a single or multiple session(s) to a network host; here they initially login to the Device Server before starting that session. The Device Server is used to configure and start the session.
- need a profile different from the Default user profile.

When a host is accessing a serial device (like a modem or a server), you can configure user accounts where users:

- are being provided a remote access service, like a SLIP or PPP connection, and they are being locally authenticated by the Device Server.
- are using a reverse telnet connection to manage a UNIX server or a router and want to be authenticated.
- are using reverse SSH to connect to the Device Server not as a Guest user.
- need a profile different from the Default user profile.

Note: You do not need user accounts for users who are externally authenticated.

User Levels

There are four **User Levels**: **Admin**, **Normal**, **Restricted**, and **Menu**. Setting up users is only necessary when the users are actually connecting to the Device Server. Oftentimes, the Device Server is used as a gateway to a network and the user never actually logs into the Device Server itself. Users who do log into the Device Server (**Line Service** set to **DSLogin** and **User Service** set to **DSPrompt**) will have to navigate by either the Menu or CLI (except for users with **Menu** privileges, who can only use the Menu).

- **Admin**—Users with **Admin** privileges have full administrative access to the IOLAN Device Server. This is not the same as the Admin user, but has equal authority (the Admin user is a permanent, factory-set user on the IOLAN Device Server).
- **Normal**—Users with **Normal** privileges have access to the Sessions menu and associated CLI only. They can start sessions, define and predefine sessions, and can change their own user environment.
- **Restricted**—Users with **Restricted** privileges have access to a restricted Sessions menu and associated CLI; they can only open sessions predefined for them by the Admin user, but not alter their own environment or sessions. Predefined sessions can also be configured to start automatically at login.
- **Menu**—Users with **Menu** privileges have access to predefined session. All other functionality is unavailable.

When the Admin user logs into the Device Server, the prompt ends with a **#**, whereas all other users' prompts ends with a **$** or **£**, depending on the character set.

Sessions

Sessions are defined for users who are coming in through a serial device and are connecting to a host on the LAN.

Users who have successfully logged into the Device Server (**User Service** set to **DSprompt**) can start up to four login sessions on LAN hosts. These users start sessions through the Menu option **Sessions**.

Multiple sessions can be run simultaneously on the same host or on different hosts. Users can switch between different sessions and also between sessions and the Device Server using hotkey commands.

Users with **Admin** or **Normal** privileges can define new sessions and connect through them, even configure them to start automatically on login to the Device Server. **Restricted** and **Menu** users can only start sessions predefined for them by the Admin user.

Users can be configured to have access to a specific port and access modes for this port, such as **Read/Write** (RW), **Read Input** (RI), **Read Output** and **Read Both** (RI & RO).

Users Connecting from LAN to Device Server to Serial Device

Easy Port Access Menu

The Easy Port Access Menu is displayed when a **Restricted** or **Menu** level user logs into the box from the Ethernet side (**Line Service** set to **Rev Telnet** or **Rev SSH**) to access a serial device. The Easy Port Access Menu displays the line number, line name, line protocol (either **`rev-tel`** or **`rev-ssh`**), and a logout option. You can only access a line if it has the same connection protocol as the one you used to log into the Device Server. So, if you used SSH to log into the Device Server and the **Line Service** is set for **Rev Telnet**, you will not be able to access the serial device connected to that line.

Reverse Sessions and Multisessions

(2 Port+ only) The interaction between reverse sessions and multisessions is somewhat complex, so the definition of each parameter is provided to give you a context for their interaction.

Reverse Session—The definition of a Reverse Session is a TCP connection to a TCP port (defined in the Line configuration as the **DS Port**) on a line that is configured for **Reverse Telnet**, **Reverse SSH**, or **Reverse Raw**. Other LAN to Device Server connections, such as the connection to the well-known Telnet or SSH ports (the management TCP ports of 23 for Telnet and 22 for SSH), do not count as reverse sessions.

Multisessions—The Line **Multisessions** limit is the number of additional reverse sessions (beyond the first reverse session) allowed on the line. Therefore, if this number is set to 5, the total number of reverse sessions allowed on the line is 5+1 or 6.

The default and minimum value is **0**, which means that reverse **Multisessions** is disabled and you only get the 1 default reverse session.

Reverse Session Security—If this Line parameter is enabled, the user must login when making a **Reverse Telnet** connection to a configured **DS Port** (the TCP port associated with the line that is configured for a reverse connection). If the line is configured for **Reverse SSH**, this option is redundant, because a login is always required when SSH is specified.

So, in order to make multiple reverse connections to a line, the line must be configured to log users in so the user's Line Access Rights can be obtained from the User profile (regardless of whether that profile comes from a defined User, the Default User's settings, or is passed to the Device Server from TACACS+ or RADIUS). This means that one of the following conditions have been met:

- The line is configured for **Reverse Telnet** and **Reverse Session Security** is enabled.
- The line is configured for **Reverse SSH**.

If users are connecting to the Device Server by the management TCP port (or well-known port of 23 for Telnet and 22 for SSH), they always have to login to the Device Server and depending on their access level can get:

admin—standard CLI/Menu with admin privileges

normal—standard CLI/Menu with normal privileges

restricted—Easy Port Configuration Menu

menu—Easy Port Configuration Menu

If users are connecting to the Device Server by the IP address and configured **DS Port** (for example, the line has a configured **DS Port** of **88**, so the user would connect with the `telnet 101.170.12.15 88` command), a line set up for **Reverse Telnet** would require a login if **Reverse Session Security** is enabled, otherwise, the User settings would dictate what the user sees. For **Reverse SSH**, the user always has to login to the Device Server.

Configuring Network Options

Hosts

This is probably one of the first Device Server options you want to configure, since so many other configuration options require a preconfigured host. You can use any host name you want, since the host name is used only by the Device Server. You can configure up to 20 hosts using IPv4 or IPv6 internet addresses or a Fully Qualified Domain Name (FQDN) on desktop Device Server models; you can configured up to 49 hosts on rack mount Device Server models.

Gateways

Gateways are hosts that connect Local Area Networks (LANs) together. If you want to access a host that isn't on your local network, you will be connected via a gateway. Gateways route data via other gateways until the destination local network is reached. There are three types of gateways:

- **Default**—A gateway that provides general access beyond your local network.
- **Host**—A gateway reserved for accessing a specific host external to your local network.
- **Network**—A gateway reserved for accessing a specific network external to your local network.

You can specify up to 20 gateways on desktop Device Server models; you can specify up to 58 gateways on rack mount Device Server models.

RIP

The Routing Information Protocol (RIP) is a routing protocol used with almost every TCP/IP implementation. Its function is to pass routing information from a router or gateway to a neighbouring router(s) or gateway(s). RIP messages contain information about destinations which can be reached and the number of hops which are required. The hop-count is the basic metric of RIP and so RIP is referred to as a 'distance vector protocol'. RIP messages are carried in UDP datagrams.

RIP for Clients Configuration and Operation

The administrator can selectively advertise networks remotely connected via a SLIP/PPP link on the Ethernet connection, and pass RIP routing information to remotely connected clients. As this can be undesirable in some environments, this behavior is configurable and is defaulted to the non-routing behavior.

Additional PPP and SLIP Functionality - RIP Packet Exchange

Transmission and reception of Routing Information Protocol (RIP) packets over PPP and SLIP connections can be configured on a per user basis or on a per line basis. The **Routing** parameter associated with a line and each local user determines the exchange of RIP packets between the Device Server and remotely connected users connected from the serial side. For a user authenticated by RADIUS, the **Framed-Routing** parameter determines the exchange of RIP packets.

The administrator has four options for setting the routing and Framed-Routing parameters:

- **None**—Routing information is not exchanged across the link. This is the default setting for a line and a locally defined user.
- **Send**—Routing information is only transmitted to the remote user.
- **Listen**—Routing information is only received from the remote user.
- **Send and Listen**—Routing information is transmitted to and received from the remote user.

The setting for the **Line Routing** parameter is the default for a connection, but the setting for the local **User Routing** parameter or RADIUS **Framed-Routing** parameter is used if this differs from the **Line Routing** parameter.

DNS/WINS

You can configure up to four DNS and four WINS servers. You can configure WINS servers for PPP-client name resolution and DNS servers for PPP-client name resolution and Device Server host name resolution (for example, when specifying **Bootup** file).

Syslog

The Device Server can be configured to send system log messages to a syslog daemon running on a remote host if the **Syslog** service is activated. You can configure a primary and secondary host for the syslog information and specify the level for which you want syslog information sent.

SNMP

If you are using SNMP to manage/configure the Device Server, or to view statistics or traps, you must set up a User in SNMP version 3 or a Community in SNMP version 1,2 to allow your SNMP manager to connect to the Device Server; this can be done in the DeviceManager, WebManager, CLI, or Menu. You must then load the perle-sds.MIB (found on the CD-ROM packaged with the Device Server) file into your SNMP manager before you connect to the Device Server.

Configuring Time

The Device Server has a real-time internal clock, allowing the date and time to be set and viewed. It will maintain the time over a short power outage and after reboots of the Device Server. If you do not set the time, it will start the clock at the factory set time.

Setting the Device Server's Time

When you set the Device Server's time, the connection method and time zone settings can affect the actual internal clock time that is being set. For example, if you are connecting to the Device Server through the DeviceManager and your PC's time zone is set to Pacific Standard Time (GMT -8:00) and the Device Server's time zone is set to Eastern Standard Time (GMT -5:00), the Device Server's time is actually three hours ahead of your PC's time. Therefore, if you set the Device Server's time to 2:30 pm in the DeviceManager (**Tools**, **Set Unit Date/Time**), the Device Server's actual internal clock time is 5:30 pm. This is the only configuration method that interprets the time and converts it between time zones, as necessary.

All other configuration methods set the Device Server's internal clock time to the time specified, with no interpretation.

Time Settings

You can set standard and summer time (daylight savings time) in the Device Server. You can specify the summer time settings as absolute, on a fixed date and time, or relative, on something like the third day of the third week at this time in June.

SNTP

You can configure your SNTP client in the Device Server to automatically synchronize the Device Server's time.

Keys and Certificates

When you are using SSH, SSL/TLS, LDAP, or HTTPS, you will need to install keys and/or certificates or get server keys in order to make those options work properly. All certificates need to be created and all keys need to be generated outside of the Device Server, with the exception of the Device Server SSH Public keys, which already exist in the Device Server. SSH keys must be generated using the OpenSSH format.

Certificate Authorities (CAs) such as Verisign, COST, GTE CyberTrust, etc. can issue certificates. Or, you can create a self-signed certificate using a utility such as OpenSSL.

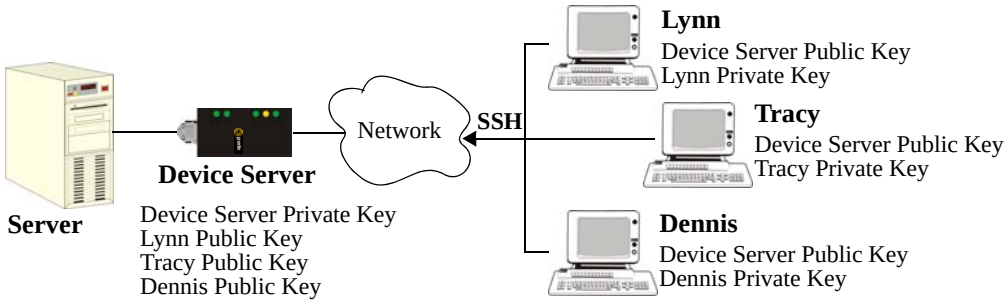
SSH

When you are using the SSH connection protocol, keys need to be distributed to all users and the Device Server. Below are a couple of example scenarios for key/certificate distribution.

Users Logging into the Device Server Using SSH (Reverse)

In the following example, users are connecting to the Device Server via SSH from the LAN. Therefore, the following keys need to be exchanged:

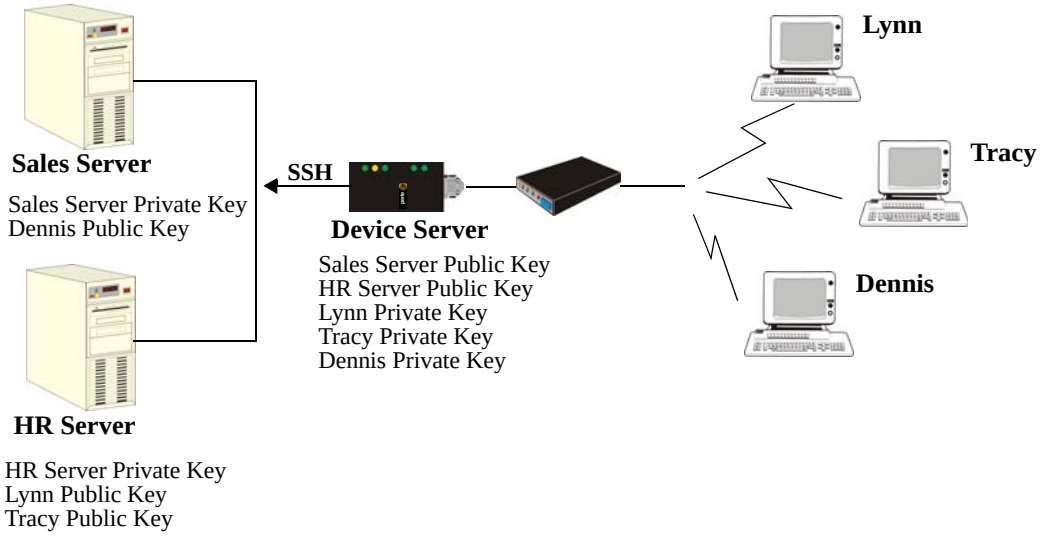
- Upload the Device Server **SSH Public Key** to each user's host machine who is connecting and logging into the Device Server using SSH.
- Download the SSH Public Key from each user's host machine who is connecting and logging into the Device Server using SSH.



Users Passing Through the Device Server Using SSH (Dir/Sil)

In the following example, users are connecting to servers on the LAN through a serial device (a modem). The **Line Service** is set to **DSLogin** and the **User Service** is set to **SSH**, therefore, users first log into the Device Server and then are connected to a specified host (configured for the user when **User Service SSH** is selected) through an SSH connection. Lynn and Tracy automatically connect to the HR Server and Dennis automatically connects to the Development Server via SSH through the Device Server. All the SSH negotiation is being done between the Device Server and the target servers, therefore, the following keys need to be exchanged:

- Download the **SSH Host Public Key** to the Device Server for each of the hosts that the Device Server is connecting to.
- Download the **SSH User Private Key** for each user whose **User Service** is set to **SSH**.
- Copy the SSH User Public Key to the host that the user is connecting to (this is done outside the scope of the Device Server).



LDAP

If you are using LDAP external authentication, you must download a CA list to the Device Server that includes the certificate authority (CA) that signed the LDAP certificate on the LDAP host. See ***Keys and Certificates*** on page 98 for more information.

HTTPS

If you are using the WebManager in secure mode (HTTPS), you need to download the SSL/TLS private key and certificate to the Device Server. You also need to set the **SSL Passphrase** parameter with the same password that was used to generate the key. See ***Keys and Certificates*** on page 98 for more information.

SSL/TLS

You can configure the SSL/TLS server to encrypt data that is sent between the Device Server and an SSL/TLS client. You can configure the cipher combinations that you want the connection to use and configure peer validation, if you want to use it.

Language support

Two language files, in addition to English, are supplied on the supplemental CD, French and German. You can use any of these language files to create a translation into a language of your choice. You can download the language file (whether the language is supplied or translated) into the Device Server and select the **Language** option of **Customlang** (custom language), making the Menu, CLI, and WebManager field labels display in your language.

You can view Menu, CLI, or WebManager in one other language only (as well as English). If you download another language file, this new language will replace the first language you downloaded.

You can revert to English at any time; the English language is stored permanently in the Device Server and is not overwritten by your new language. Each user logged into the Device Server can operate in either English or the downloaded language.

Loading a Supplied Language

This section describes how to download a language file using the CLI, since it is the least intuitive method. French and German language files are provided on the supplemental CD.

To load one of the supplied languages into the Device Server, so the Menu, CLI and WebManager fields appear in another language, do the following:

1. Open the supplemental CD and identify the language file, either **`Iolan_ds_French.txt`** or **`Iolan_ds_German.txt`**, or supply one of your own translated files.
2. Copy the language file to a host machine on the network; place it in the main file system or on the main hard drive.
3. Either use the TFTP defaults in the Device Server or, configure as necessary, TFTP in the Device Server.
4. In the CLI of the Device Server, enter the host IP address and file name; for example,

   ```
   netload customlang 172.16.4.1 /temp/Iolan_ds_French.txt
   ```

   The Device Server will download the language file via TFTP.
5. To set an individual user to the new language, go to the **Users** menu and, in the **Language** field select **Customlang**. In the CLI (only) you can set individual users or all users to the new language; see the **`set user *`** command.
6. The user will see the change of language when he/she logs out (**Main Menu**, **Sessions Menu**, **Logout**) and logs back into the Device Server. If, as Admin user, you change your language setting to **Customlang**, you will see the text menus display in the new language when you save and exit the **Change User** form. Users with **Level Normal** can also change their display language.

Note: If you download a new software version, you can continue to use your language unchanged; however, we recommend translating the new strings, which will be added to the end of the language file. A **Reset to Factory Defaults** will reload the **Customlang** as English.

On successful download, the **Customlang** in the Device Server will be overwritten by the new language.

Translation Guidance

To help you with your translation, of supplied ASCII text language files we offer the following guidance:

- The Device Server will support languages other than English (and the supplied German and French languages). The English language file, **`english.txt`**, displays the character length of each line at the beginning of the line. If a translated line goes over that character length, it will be displayed truncated in the Menu, CLI, or WebManager.
- Translate line for line, do not omit lines if you do not know the translation; leave the original untranslated text in place. Also, you must maintain the same sequential order of lines. It is a good practice to translate the file using a text editor that displays line numbers, so you can periodically verify that the line sequence has not changed from the original file (by comparing it to the original file).
- Keep all translations in quotes, otherwise the line will not display properly.
- Each line must end with a carriage return.
- If a line contains only numbers, for example 38400, leave that line in place, unchanged (unless you are using a different alphabet).

Software Upgrades and Language Files

If you receive a software upgrade for the Device Server, the language files supplied on the supplemental diskette/CD might also have been updated. We will endeavour to provide a list of those changes in another text file on the same supplemental CD.

Note: The upgrade of your software (firmware) will not change the display of the language in the Menu, CLI, or WebManager.

If you are already using one of the supplied languages, French or German, you probably want to update the language file in the Device Server. Until you update the Device Server with the new language file, new text strings will appear in English.

If you are already using a language translated from an earlier version, you probably want to amend your translation. When a language file is updated, we will try to maintain the following convention:

1. New text strings will be added to the bottom of the file (not inserted into the body of the existing file).
2. Existing text strings, if altered, will be altered in sequence; that is, in their current position in the file.
3. The existing sequence of lines will be unchanged.
4. Until you have the changes translated, new text strings will appear in the Menu, CLI, or WebManager in English.

Downloading Terminal Definitions

All terminal types can be used on the Device Server. Some terminal types which are not already defined in the Device Server, however, are unable to use Full Screen mode (menus) and may not be able to page through sessions properly. When installed, the Device Server has several defined terminal types—Dumb, WYSE60, VT100, ANSI, TVI925, IBM3151, VT320, and HP700.

If you are not using, or cannot emulate, any of these terminal types, you can add up to three additional terminal definitions to the Device Server. The terminal definitions can be downloaded from a TCP/IP host.

To download terminal definitions, follow these steps:

1. Decide which TCP/IP host you are going to use. It must be a machine with enabled.
2. Configure TFTP in the Device Server as necessary.
3. Download the new terminal definition to the Device Server as **Term1**, **Term2**, or **Term3**.
4. In the **Line** configuration, select the **Terminal Type Term*x*** that you custom defined.

Creating Terminal Definition Files

To create new terminal definition files, you need to copy and edit the information from the terminfo database.

1. On a UNIX host, change directory to **`/usr/lib/terminfo/`***`x`* (where ***x*** is the first letter of the required terminal type). For a Wyse60, for example, you would enter the command **`cd /usr/lib/terminfo/w`**.
2. The termcap files are compiled, so use the command **`infocmp`** *`termfile`* to read the required file (for example: **`infocmp wy60`**).
3. Check the file for the attribute **`xmc#`***`n`* (where ***n*** is greater than or equal to 1). This attribute will corrupt menu and form displays making the terminal type unsuitable for using Menu mode.
4. If the terminal definition is suitable, change to a directory of your choice.
5. Rename and copy the file to the directory specified at step 4. using the command **`infocmp termfile > term`***`n`* where ***n*** is greater than or equal to 1; (for example, **`infocmp wy50 > term1`**). Make sure the file has global read and execute permission for its entire path.
6. Edit the file to include the following capabilities in this format:

```
term=
acsc=
bold=
civis=
clear=
cnorm=
cup=
rev=
rmacs=
rmso=
smacs=
smso=
page=
circ=
```

For example:

```
term=AT386 | at386| 386AT |386at |at/386 console
acsc=jYk?lZm@qDtCu4x3
bold=\E[1m
civis=
clear=\E[2J\E[H
cnorm=
cup=\E[%i%p1%02d;%p2%02dH
rev=\E4A
rmacs=\E[10m
rmso=\E[m
smacs=\E[12m
smso=\E[7m
page=
circ=n
```

Note: As you can see from the example, capabilities which are not defined in the terminfo file must still be included (albeit with no value). Each entry has an 80 character limit.

On some versions of UNIX, some of the capabilities are appended with a millisecond delay (of the form $<*n*>). These are ignored by the Device Server and can be left out.

The ‘acsc’ capability, if defined, contains a list of character pairs. These pairs map the characters used by the terminal for graphics characters to those of the standard (VT100) character set.

Include only the following character pairs:

jx, kx, lx, mx, qx, tx, ux and *xx*

(where *x* must be substituted by the character used by the terminal). These are the box-drawing characters used to display the forms and menus of Menu mode. They must be entered in this order.

The last two capabilities will not be found in the terminfo file. In the **page** field you must enter the escape sequence used by the terminal to change screens. The **circ** field defines whether the terminal can use **previous page** and **next page** control sequences. It must be set to **y** or **n**. These capabilities can be found in the documentation supplied with the terminal.

TFTP Configuration

Note: TFTP file transfers send via UDP packets. When the packet delivery is interrupted for any reason and a timeout occurs, that packet is resent if the retry count allows it. Therefore, if a very large file is being transferred and is interrupted, the entire file is not resent, just the part of the file and was not received.

TFTP can be configured for two unique transfer operations:

1. **Between the DeviceManager and a** Device Server**.** This configuration is accessed by selecting **Tools**, **Options** from the DeviceManager's tool bar. You can specify the number of times the DeviceManager's TFTP server retries a file transfer to a Device Server, how many seconds the TFTP process will wait (timeout) before retrying to transfer a file, and the UDP port that will be used for the file transfer between the DeviceManager and the Device Server. (DeviceManager only.)
2. **Between the** Device Server **and a host.** This configuration is accessed by selecting **Network**, **TFTP** in the DeviceManager, by typing `set server tftp` in the CLI, by selecting **Network Configuration**, **TFTP** from the Menu, by selecting **ServerInfo**, **tftpRetry** and **tftpTimeOut** in the SNMP MIB, or by selecting **Network**, **TFTP** in the WebManager. You can configure the number of times the Device Server's TFTP client retries a file transfer to a host and how many seconds the TFTP process will wait (timeout) before retrying to transfer a file.

You must have a TFTP server running on any host that you are uploading or downloading files to/from. If you are using the DeviceManager and transferring a local file to a Device Server, you still need to have a TFTP server running on your PC. When you specify the file path, the path must be relative to the default path set in your TFTP server software.

Resetting Configuration Parameters

You can reset the Device Server to its factory settings through any of the following methods:

- You can push in the recessed button at the back of the Device Server hardware for more than three seconds (pushing it in and then quickly releasing will just reboot the Device Server)
- DeviceManager, select **Tools**, **Reset to Factory Defaults**
- CLI, at the command line type, `reset factory`
- WebManager, click the **Factory Defaults** button
- Menu, select **Network Configuration**, **Reset to Factory Defaults**
- SNMP, in the **adminInfo** folder, `Set` the **adminFunction** variable to **2**

Lost Admin Password

If the Admin user password is lost, there are only two possible ways to recover it:

- reset the Device Server to the factory defaults
- have another user that has **admin** level rights, if one is already configured, reset the Admin password

DHCP/BOOTP

You can use DHCP/BOOTP to perform the following actions on a single or multiple Device Servers on bootup:

- auto-configure with minimal information; for example, only an IP address
- auto-configure with basic setup information (IP address, subnet/prefix bits, etc.)
- download a new version of firmware
- download a full configuration file

DHCP/BOOTP is particularly useful for multiple installations: you can do all the Device Server's configuration in one DHCP/BOOTP file, rather than configure each Device Server manually. Another advantage of DHCP/BOOTP is that you can connect a Device Server to the network, turn on its power and let autoconfiguration take place. All the configuration is carried out for you during the DHCP/BOOTP process.

DHCP/BOOTP Parameters

The following parameters can be set in the DHCP/BOOTP bootp file:

- **SW_FILE**—The full path, pre-fixed by hostname/IP address (IPv4 or IPv6), and file name of the firmware update.
- **CONFIG_FILE**—The full path, pre-fixed by hostname/IP address (IPv4 or IPv6), and file name of the configuration file.
- **GUI_ACCESS**—Access to the Device Server from the HTTP or HTTPS WebManager. Values are **`on`** or **`off`**.
- **AUTH_TYPE**—The authentication method(s) employed by the Device Server for all users. You can specify the primary and secondary authentication servers, separated by a comma. This uses the following numeric values for the authentication methods.
  - **0**—None (only valid for secondary authentication)
  - **1**—Local
  - **2**—RADIUS
  - **3**—Kerberos
  - **4**—LDAP
  - **5**—TACACS+
  - **6**—SECURID
  - **7**—NIS
- **SECURITY**—Restricts Device Server access to devices listed in the Device Server's host table. Values are **`yes`** or **`no`**.
- **TFTP_RETRY**—The number of TFTP retries before aborting. This is a numeric value, for example, 5.
- **TFTP_TMOUT**—The time, in seconds, before retrying a TFTP download/upload. This is a numeric value, for example, 3.
- **CUSTOM_LANG**—The full path, pre-fixed by a hostname/IP address (IPv4 or IPv6), and file name of a translated language file. For example, `192.101.34.211 /accounting/Iolan_ds_german.txt.`
- **EXTRA_TERM1**—(**EXTRA_TERM2**, **EXTRA_TERM3**) The full path, pre-fixed by a hostname/IP address (IPv4 or IPv6), and file name of a termcap file for a specific terminal type.

SLIP vs. PPP

If you require any of the features listed below, use PPP, otherwise SLIP should be sufficient.

- **IP Address Negotiation.** SLIP provides no mechanism for informing the other end of a link of its IP address, whereas PPP will do so.
- **Error Checking.** SLIP does not error check whereas PPP does. This is not necessarily a problem in SLIP since most upper layer protocols have their own error checking. Some systems exchange UDP packets with checksum disabled, which would cause problems should that part of an IP packet get corrupted.
- **Authentication.** Once SLIP has started you cannot authenticate the remote device, whereas as PPP provides the option of using security protocols PAP or CHAP.
- **Software Flow Control.** You cannot use software flow control on SLIP links since there is no way of escaping control characters from the data stream. PPP has a facility (called ACCM) which allows specific control characters to be escaped from the data stream.
- **IPv6 Support.** SLIP does not support IPv6, but PPP does.

Creating Custom Applications

You can create custom applications for the Device Server by using the Perle SDK. See the *SDK Programmer's Guide* (the SDK and guide are found on the Perle website at `www.perle.com/downloads/index.shtml`) for information about the functions that are supported. You must download the program and any ancillary files to the Device Server and set the **Line Service** to **Custom App** to run a custom application. You must also specify the program executable in the **Program Command Line** parameter.

I/O Model Features

There is a line of I/O Device Servers that can control/monitor the following types of I/O:

- Analog (Input)
- Digital Input/Output
- Relay Output
- Temperature Input

Some of the models are I/O combinations and some of the models support one I/O type, although all of the I/O models are extensions of the feature rich, extended temperature SDS Device Server.

Failsafe Timer

The Failsafe Timer is enabled on a global basis and provides a trigger mechanism that can be configured for each channel when no I/O traffic/management has occurred for the specified amount of time. A Failsafe Action can be configured for each Digital Output channel, each Serial Signal Output channel (DTR and RTS), and each Relay channel to either Activate or Deactivate the output.

Alarms

Analog and Temperature input models support an Alarm mechanism in which you can specify up to five severity levels of alarm triggers and clear levels; the alarm triggers/clear levels can activate in either increasing or decreasing severity levels.

The Digital Input supports an Alarm mechanism based on a trigger of either active input or inactive input and can be cleared either manually or automatically (when the trigger condition goes inactive or active, respectively).

Each time an alarm is triggered or cleared, you can specify any combination of the following to be initiated:

- An SNMP trap
- An email message
- A message to syslog

UDP

The I/O UDP broadcast feature periodically broadcasts the I/O channel status in a UDP message.

You can configure up to four sets of IP address entries (each entry consisting of a start and end IP address range) to broadcast I/O status data. The data depends on the I/O model (Analog, Digital, Serial Signals) and contains information for all channels.

UDP Unicast Format

In order to interpret the UDP unicast data, you must use the following tables to decipher the appropriate data. If your model does not support a data format (for example, digital data) or you do not have any channels configured for a data format, it will be included in UDP broadcast package, with a Total Length of 0 (zero) and no data following.

Version	Total Length	Analog Data	Digital Data	Serial Signal Data

Each section, with the exceptions of the Version and Total Length sections, is comprised of its own subset of bytes.

Analog Data

Each Analog channel is comprised of the following data fields (big Endian format):

Total Length	***Data Exists**	**Data**					
2 Bytes	1 Byte (in bits)	curRawValue 2 Bytes	minRawValue 2 Bytes	maxRawValue 2 Bytes	curEngValue 4 Bytes	minEngValue 4 Bytes	maxEngValue 4 Bytes

The following section describes the values in the Analog Data field:

- **curRawValue**—The current raw value received from the Analog to Digital converter.
- **minRawValue**—The minimum value received from the Analog to Digital converter until it is cleared.
- **maxRawValue**—The maximum value received from the Analog to Digital converter until it is cleared.
- **curEngValue**—The current converted value (voltage/current for Analog or Celsius/Fahrenheit for Temperature).
- **minEngValue**—The minimum converted value (voltage/current for Analog or Celsius/Fahrenheit for Temperature) until it is cleared.
- **maxEngValue**—The maximum converted value (voltage/current for Analog or Celsius/Fahrenheit for Temperature) until it is cleared.

Digital/Relay Data

The digital data is in bit format, 1 meaning On and 0 (zero) meaning Off. Each channel has its own bit, in least significant bit order.

Length	***Data Exists**	**Data** (1 Byte, one bit for each channel)							
2 Bytes	1 Byte (in bits)					Channel4	Channel3	Channel 2	Channel 1

Serial Signal Data

The serial data is in bit format, 1 meaning On and 0 (zero) meaning Off. Each channel has its own bit, in the following order.

Length	***Data Exists**	**Data** (1 Byte for each port, one bit for each signal)							
2 Bytes	1 Byte (in bits)				RTS	DTR	CTS	DCD	DSR

***Data Exits Field**

The Data Exists field is 1 byte in least significant bit order, for each channel. If data exists for a channel, the bit will be 1, if no data exists for a channel (it is not configured), the bit will be 0 (zero).

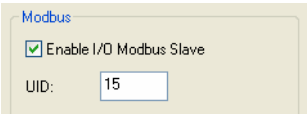
Data Exists(1 Byte, one bit for each channel)							
				Channel4	Channel3	Channel 2	Channel 1

UDP Unicast Example

For an example of the I/O UDP unicast, see the sample program, **`ioudpbcast.c`**, found on your CD-ROM.

I/O Modbus Slave

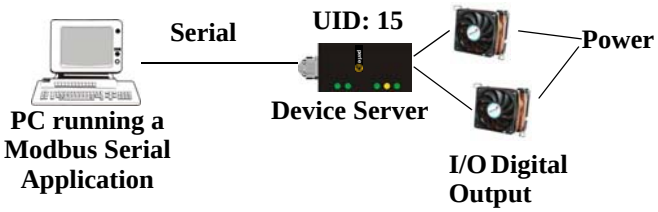
If you have a Modbus serial or TCP application, it can access I/O connected to the Device Server when the I/O Global Modbus Slave is enabled. You must supply a unique UID for the Device Server, as it will act as a Modbus Slave.



There are three ways your Modbus Application can connect to the Device Server to access I/O.

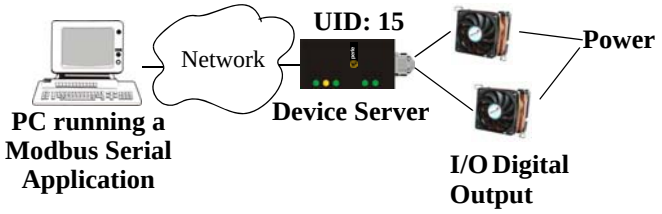
Modbus Serial Application Connected to the Serial Port

Your Modbus serial application can be connected right to the Device Server serial port to access I/O.



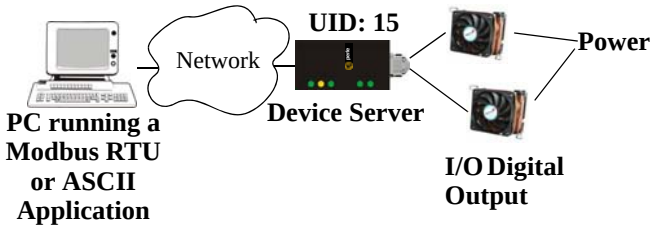
Modbus Serial Application Connected to the Network

If you want to access the I/O from a LAN connection, you can install TruePort on the PC running the Modbus serial application as described in ***TruePort*** on page 114 and connect to the Device Server over the network.



Modbus TCP Application

If you have a Modbus RTU or Modbus ASCII program, you can access the I/O by connecting to the Device Server over the network.



Modbus I/O Access

The section defines the function codes and registers you will need to access the I/O through Modbus TCP, Modbus serial, or Modbus serial/TruePort.

Function Codes

The following function codes are supported by the Device Server:

- 01 read coils
- 03 read multiple holding registers
- 04 read input registers
- 05 write coil
- 06 write single register
- 08 diagnostics (echo the request)
- 15 force multiple coils
- 16 write multiple registers

There are four Modbus data models:

Discrete Input	Not used
Coils	Digital Input (DI), Alarm state for DI, Digital Output (DO). All coils are Boolean values and are 1 byte.
Input Registers (IR)	Analog Input (AI), Alarm state for AI. All Input Registers are 2 bytes long.
Holding Registers	Status (R), Control value (R/W or W). Holding Registers with _ENG registers are 4 bytes long, all other Holding Registers are 2 bytes long.

All coil/register values are in decimal.

I/O Coil/Register Descriptions

This section contains descriptions of I/O coils:

- **MB_REG_DI_SENSOR**—Status of Digital input. 1 is Active, 0 is Inactive. If **Invert Signal** is configured **on**, 0 is Active, 1 is Inactive. If input is **Latched**, returns latched status.
- **MB_REG_DI_SENSOR_ALARM_STATE**—Indication if input is in alarm state. 1 is In Alarm state, 0 is Not in Alarm state. A write of any value clears the alarm state.
- **MB_REG_DO_SENSOR**—Status of Digital output. 1 is Active, 0 is Inactive. If **Invert Signal** is configured **on**, 0 is Active, 1 is Inactive.

This section contains descriptions of I/O holding registers:

- **MB_REG_HR_DI_SENSOR_LATCH**—The latch status of the Digital input. 1 is Latched, 0 is Not latched. A write of any value will clear the latch.
- **MB_REG_HR_DO_SENSOR_PULSE_ISW**—Inactive Signal Width. This is how long the channel will remain inactive during pulse mode in increments of 100ms. Valid values are 1-9999. The default is 1 (100 ms).
- **MB_REG_HR_DO_SENSOR_PULSE_ASW**—Active Signal Width. This is how long the channel will be active during the pulse mode in increments of 100ms. Valid values are 1-9999. The default is 1 (100 ms).
- **MB_REG_HR_DO_SENSOR_PULSE_COUNT**—The number of times the channel output will pulse. Each count consists of an active/inactive sequence. The default is 1 cycle.
- **MB_REG_HR_AI_CLEAR_ALARM_LATCH**—Used to reset a latched alarm state. A write of any value will clear the alarm latch for the specific Analog input.
- **MB_REG_HR_AI_CLEAR_MAX**—Used to reset the Analog input maximum value reached. A write of any value will reset the maximum.
- **MB_REG_HR_AI_CLEAR_MIN**—Used to reset the Analog input minimum value reached. A write of any value will reset the minimum.

This section contains descriptions of I/O input registers:

- **MB_REG_IR_CURR_ENG**—The current value of an Analog or Temperature input converted to appropriate units. For Analog, this will be in voltage or current, depending on the configuration. For the Temperature, this value will be in Celsius or Fahrenheit, depending on configuration.
- **MB_REG_IR_MIN_ENG**—The minimum converted value ever reached on this input since the Device Server was re-started or a manual clear was issued.
- **MB_REG_IR_MAX_ENG**—The maximum converted value ever reached on this input since the Device Server was re-started or a manual clear was issued.
- **MB_REG_IR_CURR_RAW**—The current raw value received from the Analog to Digital converter. This is a hexadecimal value in the range of 0 -0xFFFF.
- **MB_REG_IR_MIN_RAW**—The minimum raw value ever reached on this input since the Device Server was re-started or a manual clear was issued.
- **MB_REG_IR_MAX_RAW**—The maximum converted value ever reached on this input since the Device Server was re-started or a manual clear was issued.
- **MB_REG_IR_ALARM_LEVEL**—This gives the current alarm severity level for the corresponding Analog input. Severity levels range from 0 (not in alarm) to 5 (highest alarm severity).

Serial Port Coil/Register Descriptions

This section contains descriptions of serial port coils:

- **MB_REG_DI_DSR**—The status of the DSR input signal. 1 is Active, 0 is Inactive. If **Invert Signal** is configured **on**, 0 is Active, 1 is Inactive. If input is **Latched**, returns latched status.
- **MB_REG_DI_DSR_ALARM_STATE**—The alarm state of DSR input signal. 1 is In Alarm state, 0 is Not in Alarm state. A write of any value clears the alarm state.
- **MB_REG_DI_DCD**—The status of DCD line. 1 is Active, 0 is Inactive. If **Invert Signal** is configured **on**, 0 is Active, 1 is Inactive.
- **MB_REG_DI_DCD_ALARM_STATE**—The alarm state of DCD input signal. 1 is in Alarm state, 0 is Not in Alarm state. A write of any value clears the alarm state.
- **MB_REG_DI_CTS**—The status of CTS input signal. 1 is Active, 0 is Inactive. If **Invert Signal** is configured **on**, 0 is Active, 1 is Inactive.
- **MB_REG_DI_CTS_ALARM_STATE**—The alarm state of CTS input signal. 1 is Alarm, 0 is Not in Alarm. A write of any value clears the alarm state.
- **MB_REG_DO_DTR**—The status of DTR output signal. 1 is Active, 0 is Inactive.
- **MB_REG_DO_RTS**—The status of RTS output signal. 1 is Active, 0 is Inactive.

This section contains descriptions of serial port holding registers:

- **MB_REG_HR_DI_DSR_LATCH**—The latched status for the DSR signal. 1 is Latched, 0 is Not Latched. A write any value will clear the latch.
- **MB_REG_HR_DI_DCD_LATCH**—The latched status for the DCD signal. 1 is Latched, 0 is Not Latched. A write any value will clear the latch.
- **MB_REG_HR_DI_CTS_LATCH**—The latched status for the CTS signal. 1 is Latched, 0 is Not Latched. A write any value will clear the latch.

A4/T4 Registers

The following registers are supported by the Device Server A4 and T4 Input models:

	Data Model	**A1/T1**	**A2/T2**	**A3/T3**	**A4/T4**	**R/W**
Holding Registers:						
	MB_REG_HR_AI_CLEAR_ALARM_LATCH	2049	2050	2051	2052	W
	MB_REG_HR_AI_CLEAR_MAX	2113	2114	2115	2116	W
	MB_REG_HR_AI_CLEAR_MIN	2177	2178	2179	2180	W
Input Registers:						
	MB_REG_IR_CURR_ENG	2080	2112	2144	2176	R
	MB_REG_IR_MIN_ENG	2082	2114	2146	2178	R
	MB_REG_IR_MAX_ENG	2084	2116	2148	2180	R
	MB_REG_IR_CURR_RAW	2086	2118	2150	2182	R
	MB_REG_IR_MIN_RAW	2087	2119	2151	2183	R
	MB_REG_IR_MAX_RAW	2088	2120	2152	2184	R
	MB_REG_IR_ALARM_LEVEL	2089	2121	2153	2185	R

A4D2/A4R2 Registers

The following coils and registers are supported by the Device Server A4D2 and A4R2 I/O models:

	Data Model	A1	A2	A3	A4	D1/R1	D2/R2	R/W
Coils:								
	MB_REG_DI_SENSOR	-----	-----	-----	-----	6149	6150	R
*	MB_REG_DI_SENSOR_ALARM_STATE	-----	-----	-----	-----	6213	6214	R/W
	MB_REG_DO_SENSOR	-----	-----	-----	-----	6661	6662	R/W
Holding Registers:								
	MB_REG_HR_DI_SENSOR_LATCH	-----	-----	-----	-----	6149	6150	R/W
	MB_REG_HR_DO_SENSOR_PULSE_ISW	-----	-----	-----	-----	6213	6214	R/W
	MB_REG_HR_DO_SENSOR_PULSE_ASW	-----	-----	-----	-----	6277	6278	R/W
	MB_REG_HR_DO_SENSOR_PULSE_COUNT	-----	-----	-----	-----	6341	6342	R/W
	MB_REG_HR_AI_CLEAR_ALARM_LATCH	2049	2050	2051	2052	-----	-----	W
	MB_REG_HR_AI_CLEAR_MAX	2113	2114	2115	2116	-----	-----	W
	MB_REG_HR_AI_CLEAR_MIN	2177	2178	2179	2180	-----	-----	W
Input Registers:								
	MB_REG_IR_CURR_ENG	2080	2112	2144	2176	-----	-----	R
	MB_REG_IR_MIN_ENG	2082	2114	2146	2178	-----	-----	R
	MB_REG_IR_MAX_ENG	2084	2116	2148	2180	-----	-----	R
	MB_REG_IR_CURR_RAW	2086	2118	2150	2182	-----	-----	R
	MB_REG_IR_MIN_RAW	2087	2119	2151	2183	-----	-----	R
	MB_REG_IR_MAX_RAW	2088	2120	2152	2184	-----	-----	R
	MB_REG_IR_ALARM_LEVEL	2089	2121	2153	2185	-----	-----	R

*For DI alarm state, read will get state, write will clear alarm.

D4/D2R2 Registers

The following coils and registers are supported by the Device Server D4 and D2R2 I/O models:

	Data Model	D1	D2	D3/R1	D4/R2	R/W
Coils:						
	MB_REG_DI_SENSOR	6145	6146	6147	6148	R
*	MB_REG_DI_SENSOR_ALARM_STATE	6209	6210	6211	6212	R/W
	MB_REG_DO_SENSOR	6657	6658	6659	6660	R/W
Holding Registers:						
	MB_REG_HR_DI_SENSOR_LATCH	6145	6146	6147	6148	R/W
	MB_REG_HR_DO_SENSOR_PULSE_ISW	6209	6210	6211	6212	R/W
	MB_REG_HR_DO_SENSOR_PULSE_ASW	6273	6274	6275	6276	R/W
	MB_REG_HR_DO_SENSOR_PULSE_COUNT	6337	6338	6339	6340	R/W

*For DI alarm state, read will get state, write will clear alarm.

Serial Signals

The following coils and registers are supported by the Device Server I/O models:

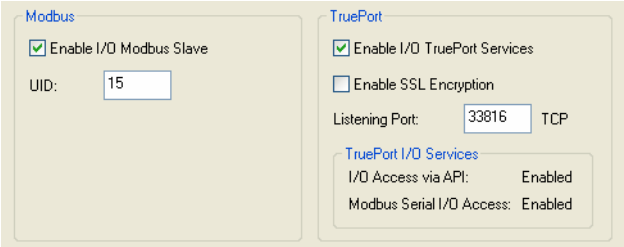
	Data Model	Pin	R/W
Coils:			
	MB_REG_DI_DSR	4225	R
	MB_REG_DI_DSR_ALARM_STATE	4289	R/W
	MB_REG_DI_DCD	4353	R
	MB_REG_DI_DCD_ALARM_STATE	4417	R/W
	MB_REG_DI_CTS	4481	R
	MB_REG_DI_CTS_ALARM_STATE	4545	R/W
	MB_REG_DO_DTR	4673	R/W
	MB_REG_DO_RTS	4737	R/W
Holding Registers:			
	MB_REG_HR_DI_DSR_LATCH	4097	R/W
	MB_REG_HR_DI_DCD_LATCH	4609	R/W
	MB_REG_HR_DI_CTS_LATCH	5121	R/W

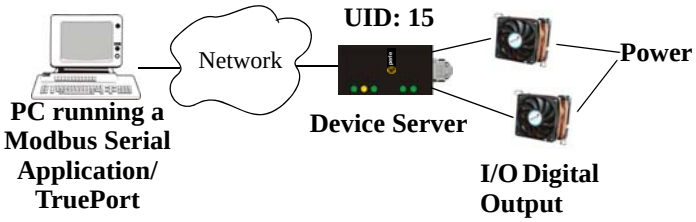
TruePort

You can see a sample API I/O over TruePort program called **`ioapiotp`**.c on the CD-ROM.

TruePort/Modbus Combination

If you have a Modbus serial application running on a PC that is connected to a network, you can use TruePort as a virtual serial connection to communicate with the Device Server over the network to access I/O data. You also have the option of enabling SSL as a security option to encrypt the data that is communicated between the Device Server and the host machine (SSL/TLS must be configured in the Server settings and on the TruePort host).

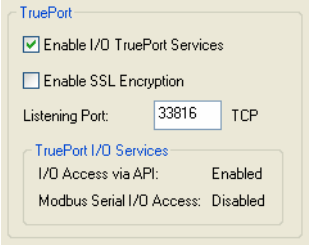


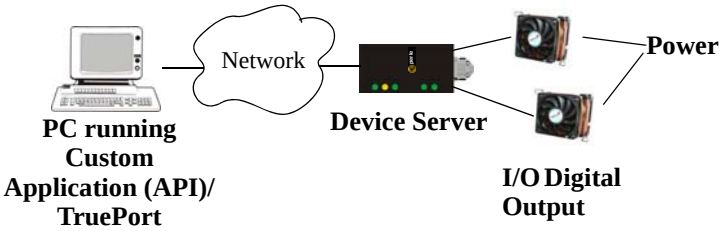


The host running TruePort must be in Modbus/ASCII or Modbus/RTU mode.

API Over TruePort Only

If you have a custom application that talks to a serial port, you can use TruePort as a virtual serial port to communicate with the Device Server over the network to access I/O data using the Perle API. You also have the option of enabling SSL as a security option to encrypt the data that is communicated between the Device Server and the host machine (SSL/TLS must be configured in the Server settings and on the TruePort host). See ***Accessing I/O Data Via TruePort*** on page 378 for more information on the API.)





The host running TruePort must be in I/O API mode.

Digital Channels

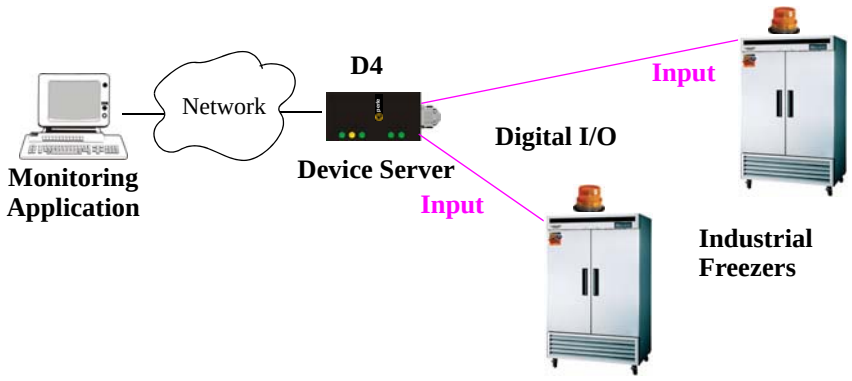
You can configure the Digital I/O channels for either input or output, making the Device Server flexible in your production environment. Jumpers must be set to correspond with the software setting of either Input or Output; see ***Digital I/O Module*** on page 48 for jumper settings.

Digital Input

The Digital input channels allow you to configure the following options:

- You can choose to remember the last state change, or latch, that occurred. Your options are to latch (remember) when the state changes from inactive to active or active to inactive.
- You can choose to invert the signal, which is useful if your sensor is wired in such a way that closed is actually inactive, whereas closed is normally considered active.
- You can also configure an alarm trigger and clear mode based on whether the Digital input is active or inactive, sending an email, syslog message, and/or SNMP trap when the alarm is triggered or cleared.

In an industrial freezer warehouse example, a D4 is used to monitor the open door sensor, so that every time a freezer door is opened, an alarm is triggered and a syslog message is sent to syslog, where the monitoring application notes the time.

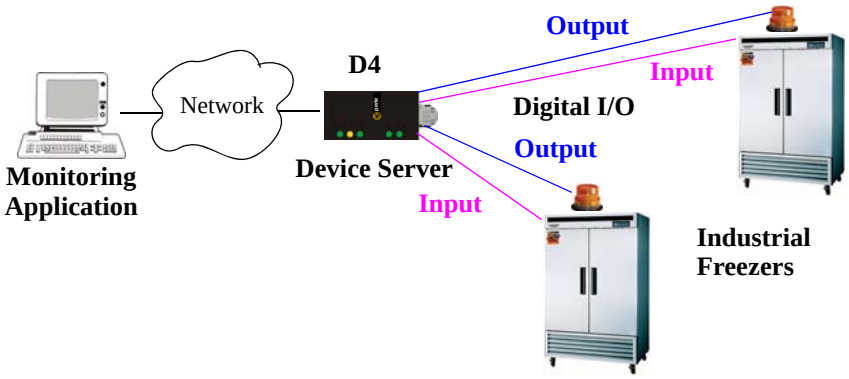


Digital Output

The Digital output channels support three types of Digital output: sink (voltage), source (ground), and sink and source (apply voltage or ground). For the output type, you can configure the following options:

- You can choose to manually activate/deactivate the Digital output.
- You can choose to manually activate/deactivate the Digital output and then specify that the Digital output will either pulse (you get to specify the active and inactive pulse times) continuously or for a specified number of pulse counts.
- You can choose to manually activate/deactivate the Digital output and then specify a delay before the output goes from inactive to active or active to inactive.
- You can also specify a failsafe action that can either activate or inactivate the Digital output when the failsafe timer is triggered (see ***Failsafe Timer*** on page 106 for more information).

When one of the industrial freezer doors are left open for more than five minutes, the Monitoring Application (using the Perle API) starts the Digital output sink, causing the strobe light on top of the offending freezer to activate.



Temperature Channels

Temperature input channels monitor RTD or thermocouple temperature sensors. You can also configure severity alarms that can send an email, a syslog message, and/or an SNMP when an alarm is triggered or cleared; See ***Alarms*** on page 107 for more information about the alarms.

RTD ranges are:

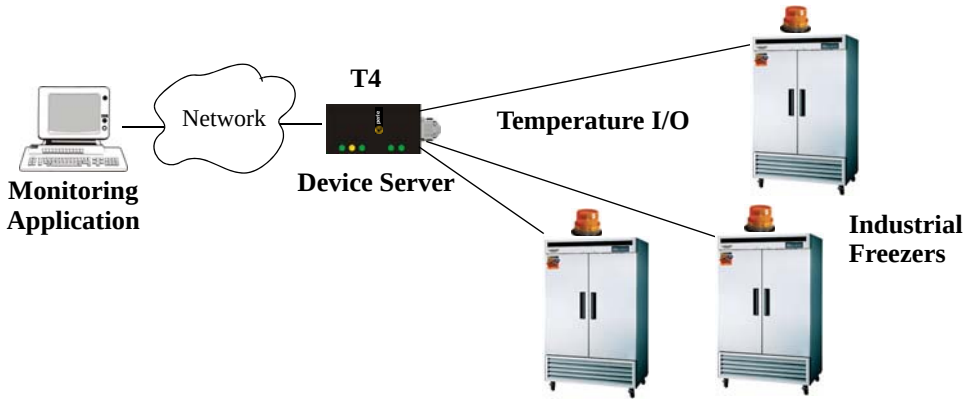
- Pt100 a=385 -50 to 150C
- Pt100 a=385 0 to 100C
- Pt100 a=385 0 to 200C
- Pt100 a=385 0 to 400C
- Pt100 a=385 -200 to 200C
- Pt100 a=392 -50 to 150C
- Pt100 a=392 0 to 100C
- Pt100 a=392 0 to 200C
- Pt100 a=392 0 to 400C
- Pt100 a=392 -200 to 200C
- Pt1000 a=385 -40 to 160C
- NiFe604 a=518 -80 to 100C
- NiFe604 a=518 0 to 100C

Note: IEC RTD 100 ohms.=0.00385
JIS RTD 100 ohms.=0.00392

Thermocouple ranges are:

- B 500 to 1800C
- E 0 to 1000C
- J 0 to 760C
- K 0 to 1370C
- R 500 to 1750C
- S 500 to 1750C
- T -100 to 400C

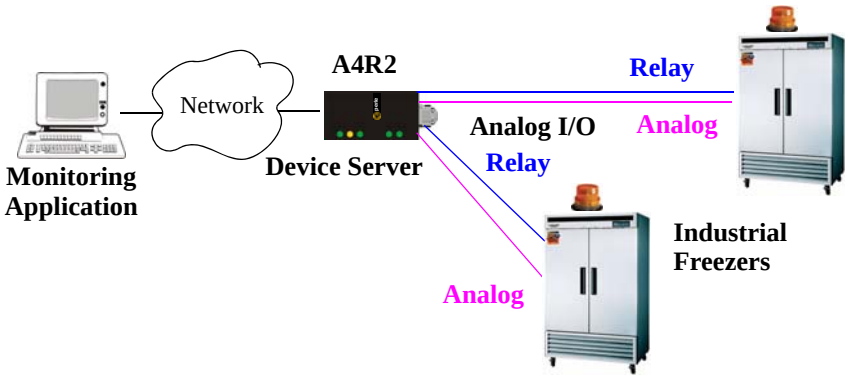
The the following example, a Temperature I/O Device Server is used to monitor industrial freezer as a temperature sensor, with an alarm set to send a syslog message if the temperature rises above 31° C.



Analog Channels

Analog input channels monitor current or voltage within a specified range and can then trigger an alarm for up to five severity levels that will send an email, SNMP trap, and/or syslog message when the alarm is triggered and/or cleared. See ***Alarms*** on page 107 for more information on alarms.

In our industrial freezer warehouse, a Device Server A4R2 is used to monitor humidity transducers, which are in place to help prevent freezer burn. If the humidity reaches a certain percentage (monitored by an Analog channel) a syslog message is sent to the Monitoring Application. The Monitoring Application then sends a command to the Device Server via the Perle API that causes the Relay channel to activate an internal freezer dehumidifier. The relay is turned off when the Analog channel sends a clear syslog message to the Monitoring Application and the Relay channel is deactivated.



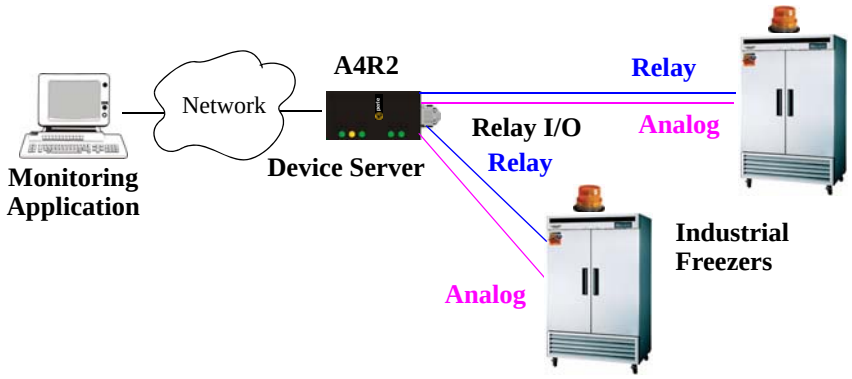
Relay Channels

Relay output channels work as a physical on/off switch, and are used to drive higher voltage devices with a lower controlling voltage.

You can configure the following Relay output channel options:

- You can choose to manually activate/deactivate the Relay output.
- You can choose to manually activate/deactivate the Relay output and then specify that the Relay output will either pulse (you get to specify the active and inactive pulse times) continuously or for a specified number of pulse counts.
- You can choose to manually activate/deactivate the Relay output and then specify a delay before the output goes from inactive to active or active to inactive.
- You can also specify a failsafe action that can either active or inactivate the Relay output when the failsafe timer is triggered (see ***Failsafe Timer*** on page 106 for more information).

In our industrial freezer warehouse, a Device Server A4R2 is used to monitor humidity transducers, which are used to help prevent freezer burn. If the humidity reaches a certain percentage (monitored by an Analog channel) a syslog message is sent to the Monitoring Application, causing the Relay channel to activate an internal freezer dehumidifier. The Relay channel is deactivated when the Analog channel sends a clear syslog message to the Monitoring Application and the Relay channel is deactivated.



Serial Signals

When the **Line Service** is set to **Signal I/O**, you will be able to configure the serial signal I/O pins. This allows you to enable/disabled each of the serial pins in the serial port that can be used as Digital inputs and outputs. The serial signal Digital I/O channels can be set for Digital Output (RTS and DTR) and/or Digital Input (DSR, DCD, and CTS).

SNMP Traps

When you enable SNMP traps for Digital and Analog inputs, a value is returned when an alarm triggers or clears. This section decodes the SNMP trap values.

Value	Alarm	Description
0	IO_DI_ALARM_SENSOR	Trap for the Digital input Trigger.
1	IO_DI_ALARM_SERIAL_DSR	Trap for the Digital input DSR serial pin Trigger.
2	IO_DI_ALARM_SERIAL_DCD	Trap for the Digital input DCD serial pin Trigger.
3	IO_DI_ALARM_SERIAL_CTS	Trap for the Digital input CTS serial pin Trigger.
4	IO_AI_ALARM_LEVEL1	Trap for Analog input Alarm Level 1.
5	IO_AI_ALARM_LEVEL2	Trap for Analog input Alarm Level 2.
6	IO_AI_ALARM_LEVEL3	Trap for Analog input Alarm Level 3.
7	IO_AI_ALARM_LEVEL4	Trap for Analog input Alarm Level 4.
8	IO_AI_ALARM_LEVEL5	Trap for Analog input Alarm Level 5.
9	IO_DI_ALARM_SENSOR_CLEAR	Trap for Digital input trigger Clear Mode.
10	IO_DI_ALARM_SERIAL_DSR_CLEAR	Trap for Digital input DSR serial pin trigger Clear Mode.
11	IO_DI_ALARM_SERIAL_DCD_CLEAR	Trap for Digital input DCD serial pin trigger Clear Mode.
12	IO_DI_ALARM_SERIAL_CTS_CLEAR	Trap for Digital input CTS serial pin trigger Clear Mode.
13	IO_AI_ALARM_LEVEL1_CLEAR	Trap for the Analog input Alarm Level 1 Clear.
14	IO_AI_ALARM_LEVEL2_CLEAR	Trap for the Analog input Alarm Level 2 Clear.
15	IO_AI_ALARM_LEVEL3_CLEAR	Trap for the Analog input Alarm Level 3 Clear.
16	IO_AI_ALARM_LEVEL4_CLEAR	Trap for the Analog input Alarm Level 4 Clear.
17	IO_AI_ALARM_LEVEL5_CLEAR	Trap for the Analog input Alarm Level 5 Clear.

Calibrating Analog Input

To calibrate an Analog input channel, read the section that applies to the type of input you are calibrating. Note that calibration will be done for the active channel configuration; for example, if Channel A1 is set to voltage, you cannot calibrate it for current. The voltage range configured for this channel will also dictate what is being calibrated. For example, if this channel is configured for a range of +/-10V, calibrating this channel will calibrate all channels which are configured for +/-10V. During the calibration process, you will be asked to apply the minimum and maximum configured range value to the channel; for example, to calibrate for voltage +/- 10V, you will be prompted to first apply -10V and then +10V to the channel.

Also, you cannot actively calibrate disabled channels (although, for Voltage, if you enable the channel and then set it for a range that has already been calibrated for another channel, it will also be calibrated).

Calibrating Voltage

When calibrating the Device Server Analog input for voltage, you will need a calibration meter that is better than .1% volts precision. When you calibrate one channel, all voltage channels are automatically calibrated for that range; if another channel is set for a different range, you will need to calibrate that channel separately, but all channels that use that range are also automatically calibrated.

Calibrating Current

When calibrating the Device Server Analog input for current, you will need a calibration meter that is better than .1% current precision. Each channel needs to be calibrated individually.

Calibrating Temperature Input

To calibrate an Analog (Temperature) input channel, read the section that applies to the type of input you are calibrating. Note that calibration will be done for the active channel configuration; for example, if Channel A1 is set to thermocouple, you cannot calibrate it for RTD. During the calibration process, you will be asked to apply the minimum and maximum range value to the channel in either mV or Ohms; for example, to calibrate for thermocouple J 0 to 760C, you will be prompted to first apply -80mV and then +80mV to the channel.

Also, you cannot actively calibrate disabled channels (although if you enable the channel and then set it for the type of thermocouple or RTD that has already been calibrated on another channel, it will also be calibrated).

Calibrating Thermocouple

When calibrating the Device Server Analog input for thermocouple, you will need a calibration meter that is better than .15% accuracy. When you calibrate one channel, all thermocouple channels are automatically calibrated for that range; if another channel is set for a different range, you will need to calibrate that channel separately, but all channels that use that range are automatically calibrated.

Calibrating RTD

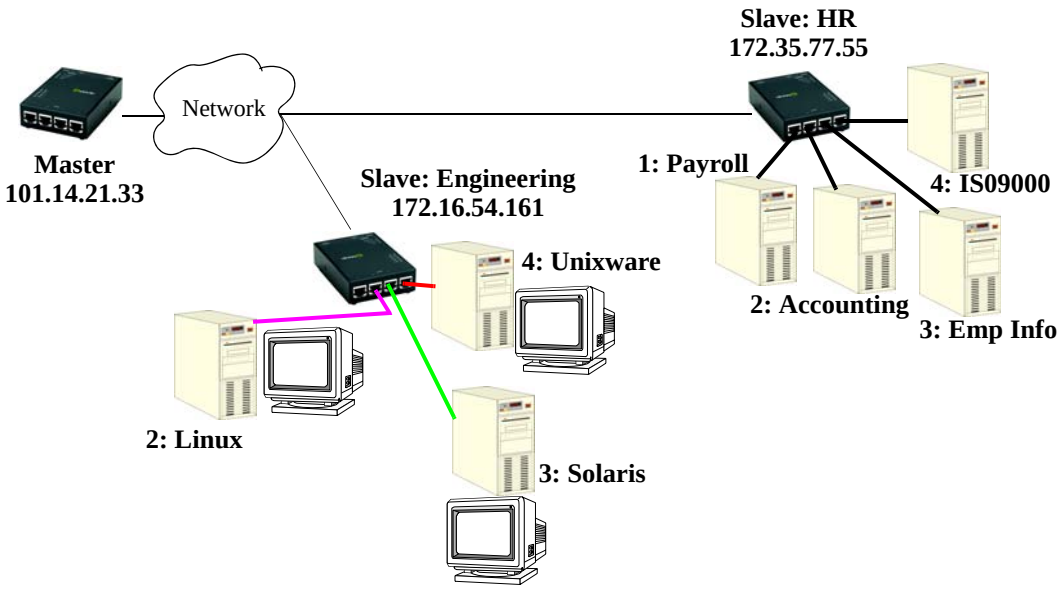
When calibrating the Device Server Analog input for RTD, you will need a resistor that is better than .05% Ohms accuracy. When you calibrate one channel, all RTD channels are automatically calibrated for that range; if another channel is set for a different range, you will need to calibrate that channel separately, but all channels that use that range are automatically calibrated.

Clustering

Clustering allows users to access Slave Device Servers through a Master Device Server via the Master Device Server's IP address and a TCP port, so users only need to remember one IP address.

Setting Up Slave Device Servers

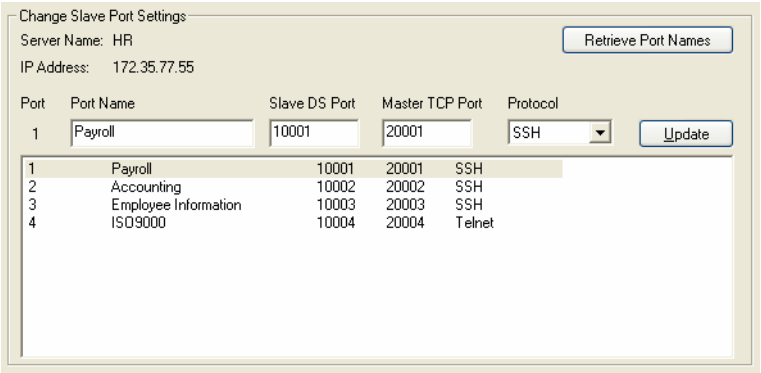
In the following example, the Master Device Server has two Slave Device Servers defined in its clustering group (Engineering and HR):



It is necessary that the Master Device Server have the ability to ping the Slave Device Servers, to verify that communication between the Device Servers exists, especially when accessing different network subnets (this can also require some Gateway configuration on the Slave Device Servers).

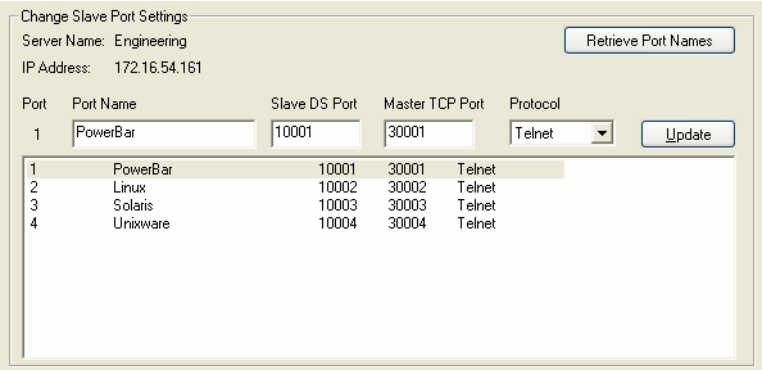
On the Master Device Server, the two Slave Device Servers have been configured as shown.

Slave Device Server: HR



Users can access serial devices connected to the Slave Device Servers through the Master Device Server's IP address and the Master TCP Port number for the serial port. So, a user who wants to access the Payroll server would need to open an SSH session to the Master Device Server's IP address 101.14.21.33 and port 20001.

Slave Device Server: Engineering

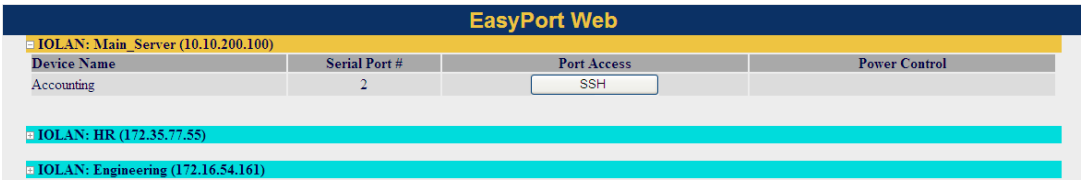


A user who wants to access the Linux server would need to open a Telnet session to the Master Device Server's IP address 101.14.21.33 and port 30002.

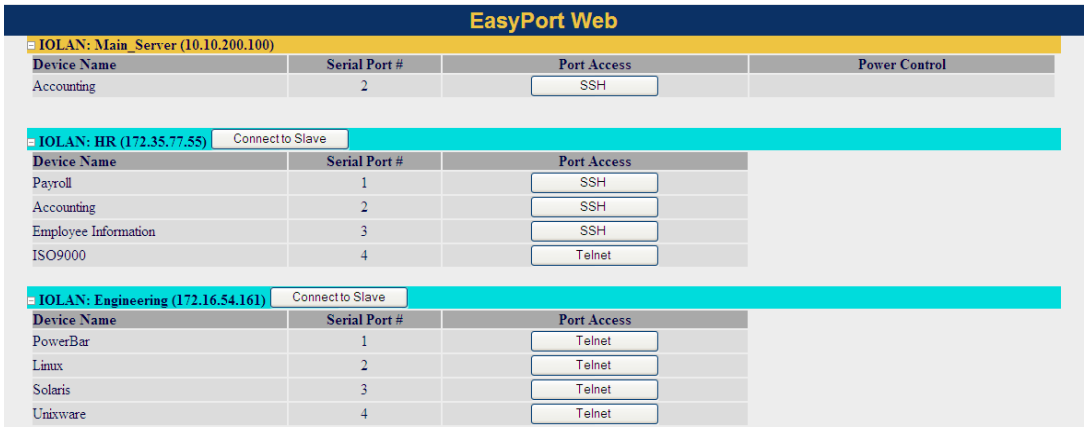
Accessing Slave Device Servers

One of the easiest ways to access any of the Slave Device Servers is through WebManager's EasyPort Web. Any user (except users with **`admin`** level privileges, who can access EasyPort Web by clicking on the **EasyPort Web** button in WebManager) who accesses the Master Device Server through a web browser will automatically see all the Slave Device Servers that they can access. All a user needs to do is click a button to start either a Telnet or SSH session to a Slave Device Server.

When a user accesses the Master Device Server through a web browser by the Master Device Server's IP address, EasyPort Web is displayed:



You can expand the Slave Device Servers as shown:



Click the **Port Access** button, **Telnet** or **SSH**, for the device you want to access and a java applet is launched to connect to the device.

Wireless WAN (SCS only)

SCS Device Server models support a wireless WAN card that can be installed to permit access to the Device Server via the internet or other WAN network. When the PCI card type has been configured to be a **Wireless WAN** card, verify that the **PCI Card Line** is set to **PPP**. No other PPP configuration is typically required. The wireless WAN card will establish a GPRS data connection over the service provider's GSM network. The service provider will assign an IP address to your wireless connection. This address may be public or private and it may be dynamically or statically assigned, depending on the type of account established with the service provider. If a static, public IP address has been assigned, the Device Server will be directly accessible via that IP address. If a dynamic, public IP address has been assigned, you may access your Device Server with the assistance of a dynamic DNS service provider. These service providers provide a method of accessing your device server using a standard URL (for example, yourcompany.dyndns.org), when the IP address assigned by the Wireless provider is dynamic. The IOLAN SCS supports dynamic DNS updates to DynDNS.com (see www.DynDNS.com for more information).

Dynamic DNS

Dynamic DNS Service providers enable users to access a server connected to the internet that has been assigned a dynamic IP address. The Device Server product line has built-in support for the DynDNS.com service provider. When the Device Server is assigned a dynamic IP address, it will inform the DynDNS.com service provider of its new IP address. Users may then use DynDNS.com as a DNS service to get the IP address of the Device Server. In order to take advantage of this service the following steps need to be taken.

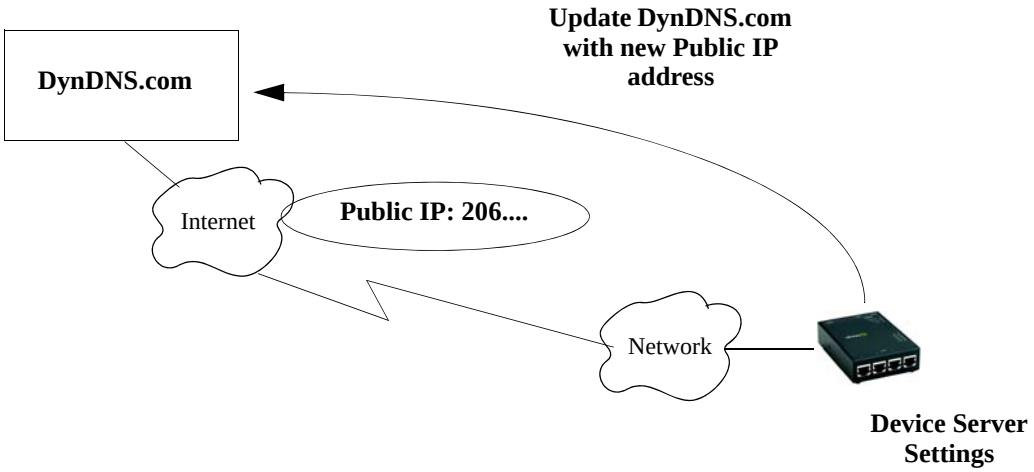
1. Create an account with DynDNS.com and configure the name your Device Server will be known by on the internet (the **Host** name). For example, create a host name such as yourcomapnySCS.DynDNS.org.
2. Enable the **Server Dynamic DNS** feature and configure the Device Server's dynamic DNS parameters to match the **Host**'s configuration on the DynDNS.com server. Every time the Device Server gets assigned a new IP address, it will update DynDNS.com with the new IP address.
3. Users accessing the Device Server via the internet can now access it via its fully qualified host name. For example, **`telnet yourcompanySCS.DynDNS.org`**.

Dynamic DNS Update

When the **Server Dynamic DNS** feature is enabled and the DynDNS.com account information configured, the Device Server will automatically update the DynDNS.com server with the public IP address assigned by the internet service provider (ISP). In the example below, an public IP address of 206.xx.xx.xx is assigned to the Device Server by the ISP. The ISP should also provide the following:

- The Device Server will need to have the Default Gateway configured so IP packets can be routed to the internet.
- You will also need to verify that a valid DNS entry (in the Network settings) has been created, since the DynDNS.com server is accessed via its Domain Name or URL.

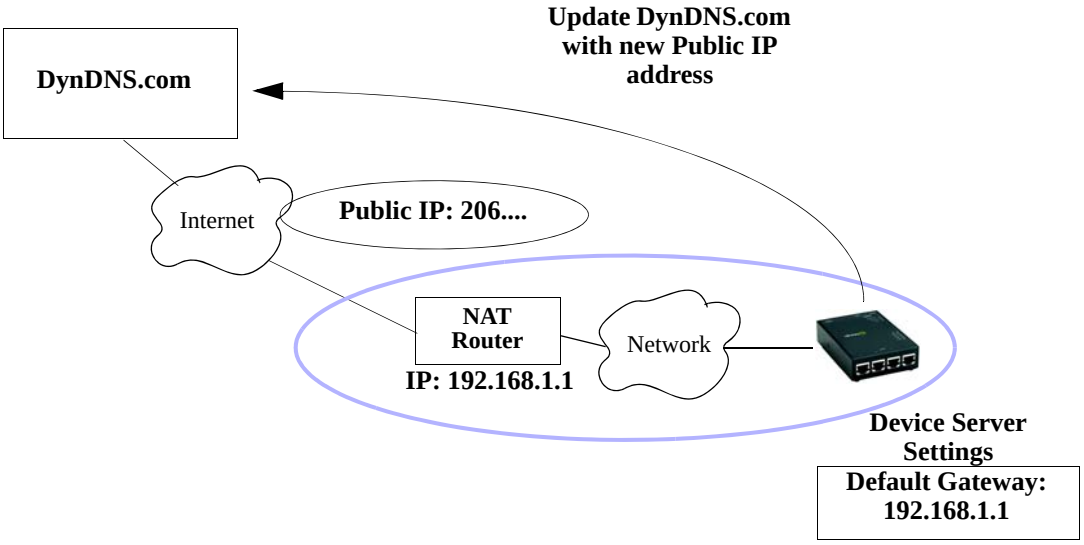
If the internet service provider changes the Device Server's IP address and Dynamic DNS is enabled and properly configured, the Device Server will automatically send an update message to DynDNS.com to update it with the newly assigned IP address.



Using Dynamic DNS Behind a NAT Router

If the Device Server is installed on a private network and has access to the internet via a router that performs NAT (Network Address Translation), this feature will still operate correctly. The Device Server determines its internet facing (public) IP address by sending a special request to the DynDNS.com server. This is the IP address that is used to update the DynDNS.com server. If setting up this type of configuration, verify that:

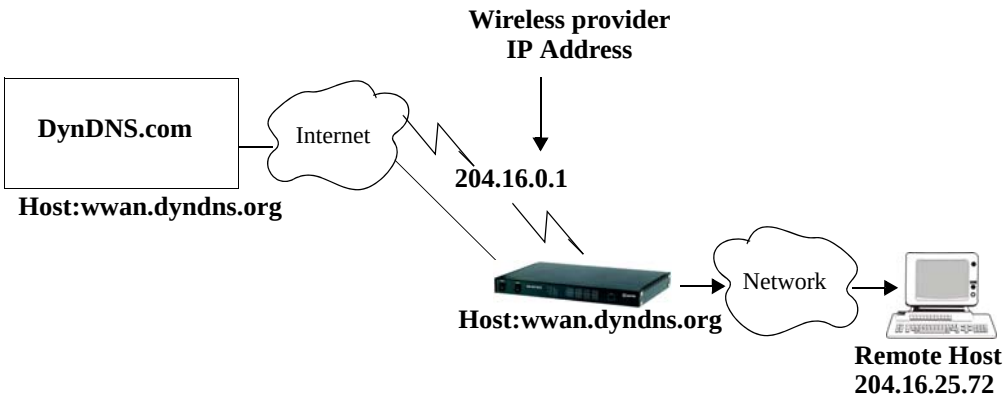
- The NAT router is identified on the Device Server as the Default Gateway.
- A valid DNS server is defined in the Device Server's network settings.
- You may need to setup Port Forwarding on the router to ensure that IP packets for sessions initiated on the internet can be routed to Device Serverr.



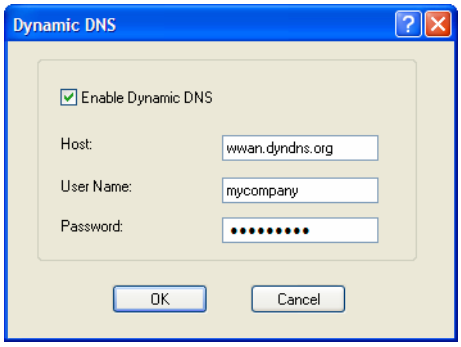
Dynamic DNS with Wireless WAN (SCS Only)

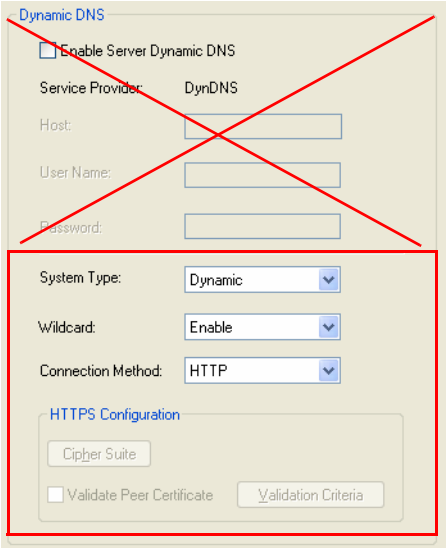
If the Device Server has been setup to establish an internet connection over the Wireless WAN, it is likely that the internet service provider will have assigned a dynamic IP address to the connection. If you want users to be able to access the Device Server via the wireless WAN connection, using an internet URL instead of an IP address, you can use the Dynamic DNS feature. Once this feature is enabled, the Device Server will update the DynDNS.com server with the IP address negotiated for the wireless WAN session. You will be required to create an account with DynDNS.com, and select an internet URL (Host name).

In the example below, the host name **`wwan.dyndns.org`** is registered with DynDNS.com. When the wireless WAN card connects to the wireless provider, the wireless provider assigns the IP address of 204.16.0.1 for the session. When Dynamic DNS is enabled and configured, the Device Server sends a message to DynDNS.com to update it host entry, **`wwan.dyndns.org`** with the assigned IP address.



PPP is used to create the connection between the wireless WAN card and the wireless provider, so the **PCI card** line (the last configurable line in your SCS model) **Service** must be set to **PPP** with **IP Address Negotiation** enabled. You must also enable **Dynamic DNS** associated with the **PPP IP Address Negotiation** and configure the dynamic DNS parameters. Also, you need to verify that the dynamic DNS parameters configured for the server match the your DynDNS.org account configuration.



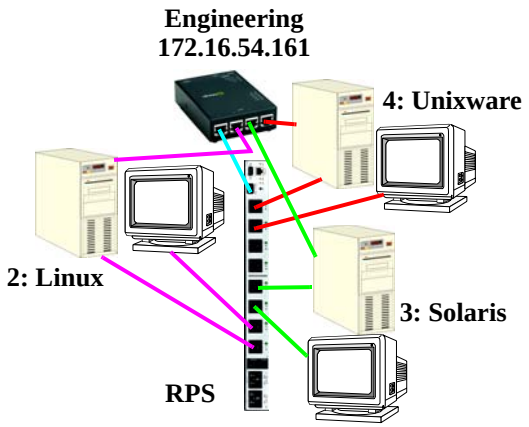


Power Management

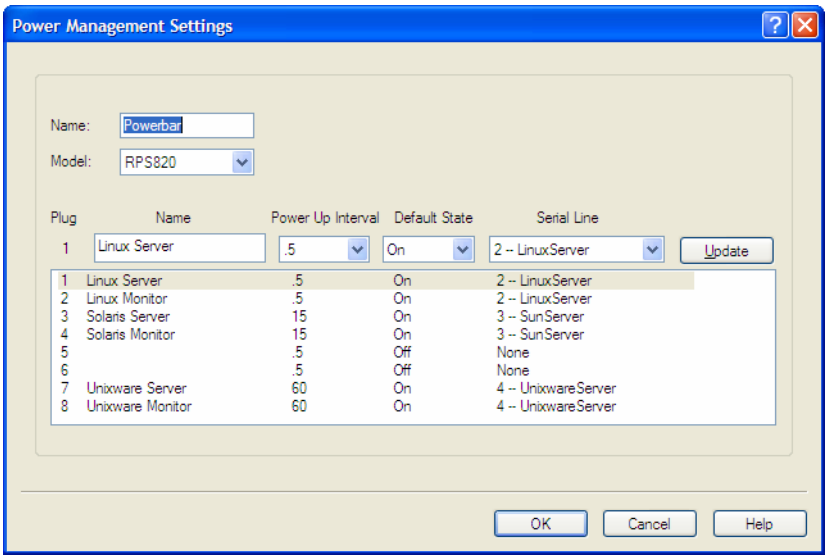
Setting Up the Device Server

If you have purchased a Perle RPS (Remote Power Switch) and have it connected to a Device Server's serial port, you can manage the plugs on the RPS through the DeviceManager, CLI, Menu, SNMP, or EasyPort Web.

In the following example, the Perle RPS is connected to serial port 1 and there are various other servers connected to the other serial ports. Each Unix server and its monitor is plugged into the RPS so that they can be managed through the power switch if, for example, the server should become remotely inaccessible.



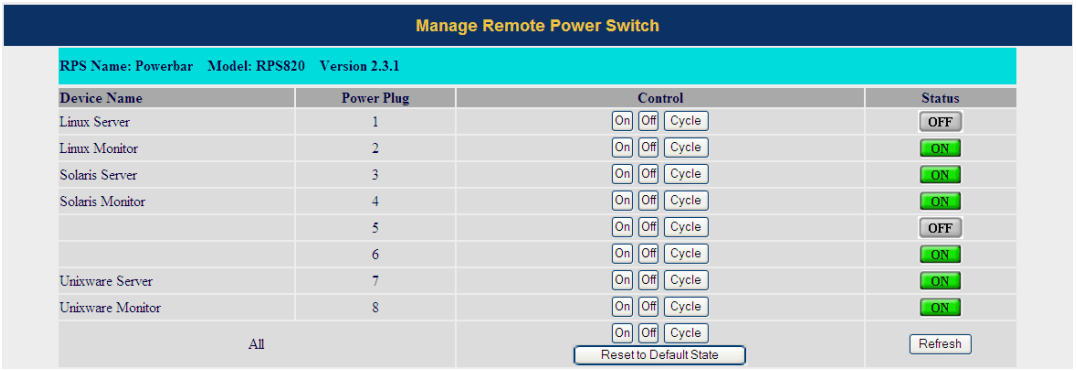
The **Line** settings for serial line 1 are set to **Service Power Management**. The Power Management settings are configured to reflect the device (by device name) plugged into each RPS plug and its associated serial line (this allows a user to connect directly to a port and manage the plugs for all the devices associated with that port).



Accessing the RPS Through EasyPort Web

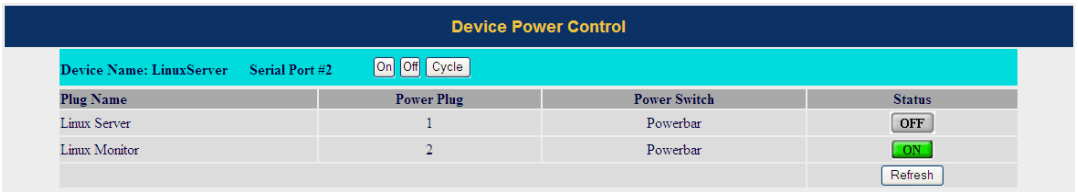
Any user can access and control all the plugs in an RPS connected to a Device Server by typing the Device Server's IP address into a web browser's **Address** field and entering their **User Name** and **Password**. The admin user and users who have **Admin Level** access rights can open EasyPort Web through the WebManager by clicking the **EasyPort Web** button in the navigation pane. All other users will automatically get EasyPort Web as shown:

From EasyPort Web in this example, a user can either manage the entire RPS unit by clicking the **Manage RPS** button for **Serial Port # 1** to control the plugs individually:



Or a user can mange the all plugs associated with a serial line by clicking on the **Device Power** button for **Serial Port # 2**:

The user can now control all the plugs associated with the serial port (individual plug control is not available).



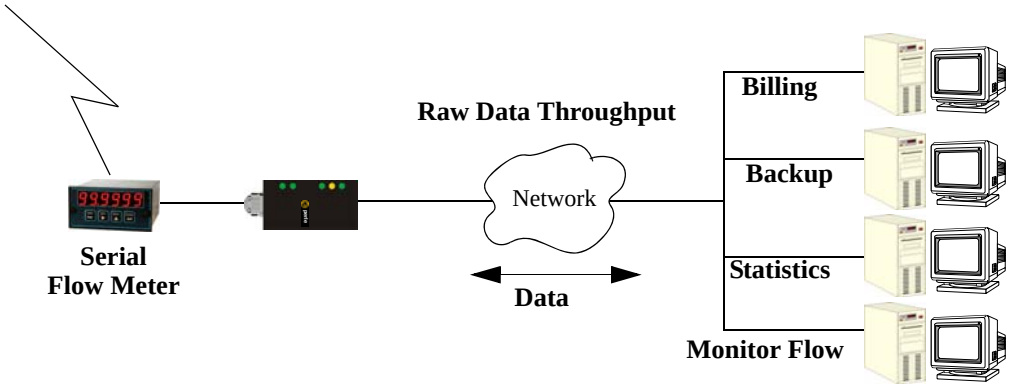
Configuring Multiple Hosts

A serial device connected to a Device Server can send data to several hosts on the network or several hosts on the network can send data to a serial device connected to a Device Server using the Multihost option. The Multihost option is supported on **Silent Raw**, **Reverse Raw**, and **TruePort Line Services**. Connections to the hosts are attempted as soon as the line is active (when the Device Server powers up, the line is enabled, or a kill line is performed). When a host is unavailable or the connection to a host drops, the Device Server will attempt to re-establish the connection every three minutes. If you are using the primary/backup host schema, if the primary host is unavailable or the connection drops, the Device Server will connect to the backup host, but will continue to re-establish the connection to the primary host every three minutes. Once the communication between the serial device and the network host is successfully established, data can be exchanged in both directions between both parties.

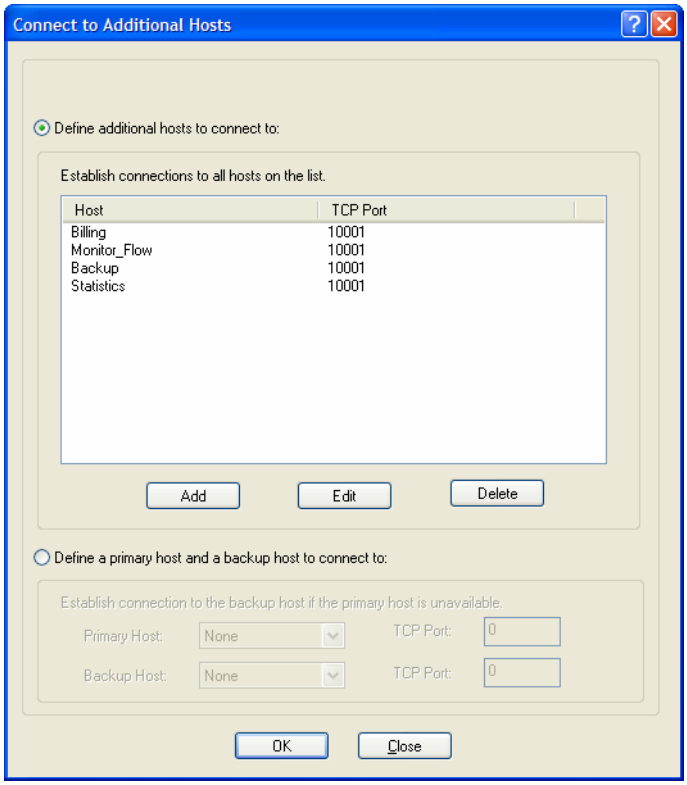
Using the Silent Raw Line Service

Connecting to Multiple Hosts

In the following example, a serial flow meter collects flow data that is communicated to several systems that track the data in several different applications.



Since the Flow Meter is sending data to the network hosts, the **Line Service** is set to Silent Raw (**Sil Raw**), the **Connect to Multiple Hosts** option is enabled, and the hosts that run the applications that require the flow meter data are added to the multihost list.

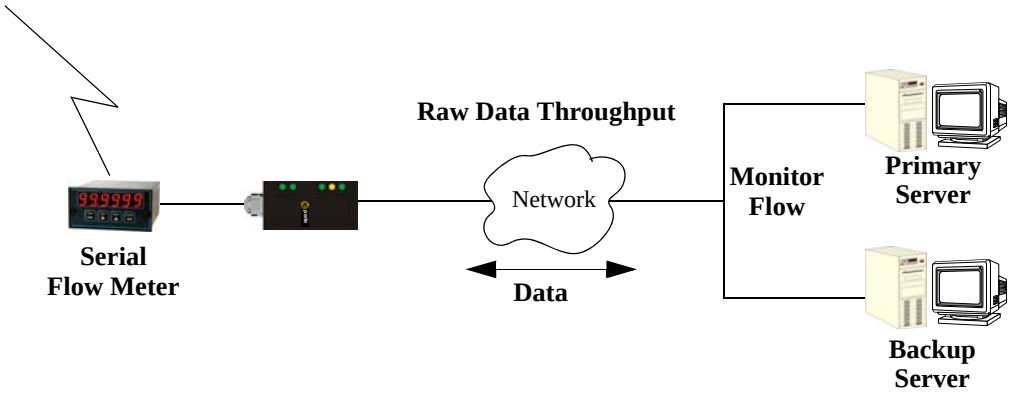


When the Device Server receives data from the Flow Meter, it will automatically be sent to all the hosts in the multihost list.

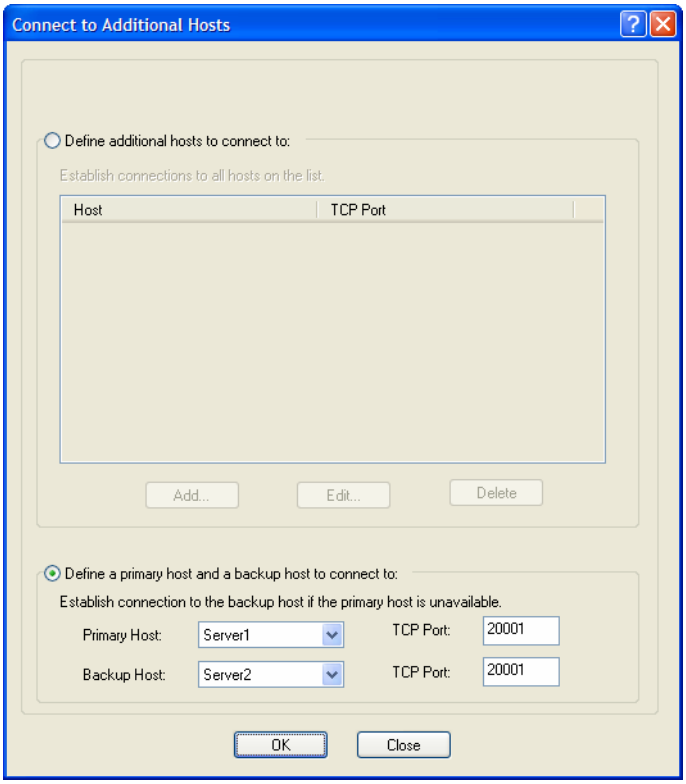
If you have a serial device that requires data from hosts on the network, you would set the **Line Service** to **Reverse Raw** and enable the **Allow Multiple Hosts to Connect** option.

Connecting to a Primary/Backup Host

In the following example, a serial flow meter collects flow data that is communicated to a system that tracks the flow data. Because the flow data is critical to the function of machinery, a backup server is maintained should the primary server become unavailable.



Since the Flow Meter is sending data to the primary server, the **Line Service** is set to Silent Raw (**Sil Raw**) and the **Connect to Multiple Hosts** option is enabled. In the **Connect to Additional Hosts** window, the **Define a primary host and backup host to connect to** option is enabled.

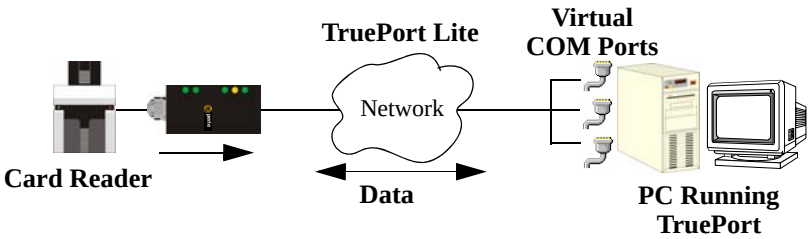


The primary server and backup server are configured. Should the primary server become unavailable, the Device Server will connect to the backup server, but will try to re-establish the connection to the primary server every three minutes.

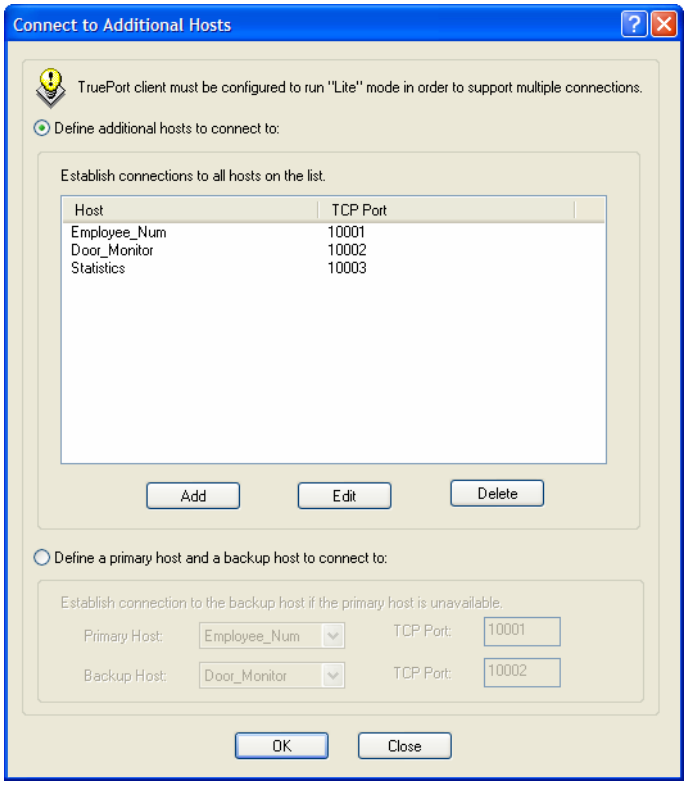
Using the TruePort Line Service

Server-Initiated

In the following example, a Card Reader sends data to a host that is running multiple serial applications that require the Card Reader data. TruePort is installed (and must be configured for TruePort Lite) on the host and a virtual COM port is created for each serial application. The Device Server's serial port is configured for **Line Service TruePort**.



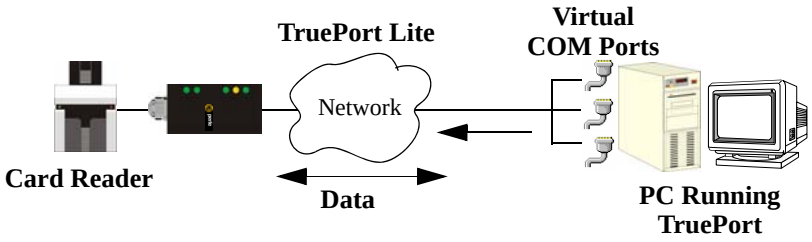
TruePort is configured on the Device Server to **Connect to** multiple hosts. In this example, the IP address of the host is the same (although you can use unique host names for the same IP address to identify the corresponding application), but the TCP Port corresponds to the virtual COM port configured on the TruePort host for each serial application.



When the Device Server receives data from the Card Reader, it will send the data to every host (or in this case, equivalent virtual COM port) configured in the multihost table.

Client-Initiated

In the following example, several serial applications poll a Card Reader for data. TruePort is installed (and must be configured for TruePort Lite) on the host and a virtual COM port is created for each serial application. The Device Server's serial port is configured for **Line Service TruePort**.



TruePort is configured on the Device Server to **Listen for Connections** and the **Allow Multiple Hosts to Connect** option is enabled. Each serial application can now access the serial device connected to the Device Server's serial port. Up to 1024 hosts can connect to this serial port.

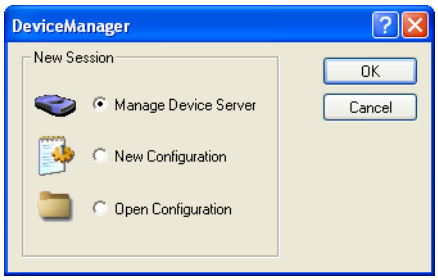
5 Using the DeviceManager

Introduction

This chapter provides information about configuring/managing the Device Server using the DeviceManager. It is assumed that the DeviceManager has already been installed; if you still need to install the DeviceManager, see ***Using DeviceManager*** on page 54.

Starting a New Session

When you start the DeviceManager application, the New Session window is displayed.

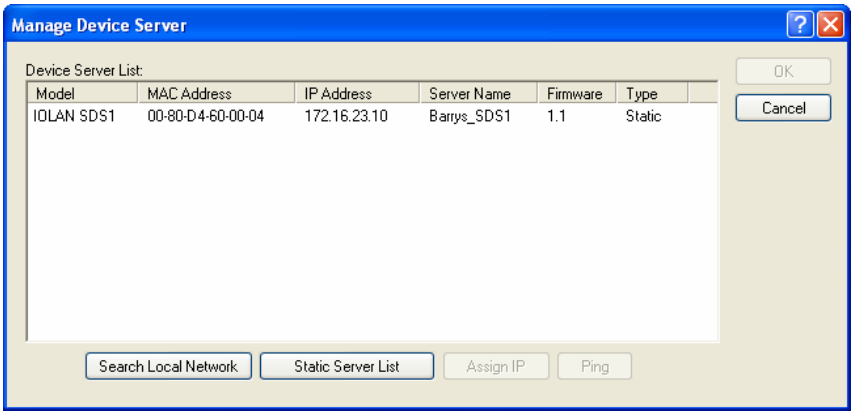


You can choose:

- **Manage Device Server**—Connect to a Device Server to manage/view it.
- **New Configuration**—Create a new Device Server configuration.
- **Open Configuration**—Open an existing configuration file.

Managing a Device Server

You can connect to Device Servers or assign an IP address (temporary or permanent) to a new Device Server. Whenever you connect to a Device Server through the DeviceManager, you connect as the Admin user and must supply the password for the Admin user.



If you want to connect to a Device Server to manage/configure it, or assign an IP address to a Device Server, select **File**, **New Session** and the **Manage Device Server** radio button. If you want to create a new or edit an existing configuration file, select **File**, **New Session** and the **New Configuration** or **Open Configuration** radio button, respectively.

If you have a Device Server on the network that has an IP address that is invalid through the router, for example, the Device Server has an IP address of 171.16.25.45 and the local network is 12.12.0.0, the DeviceManager can find the Device Server (the router must have multicast enabled) and you can assign a valid IP address by clicking the **Assign IP** button.

Populating the Device Server List

The first time you start the DeviceManager, the **Manage Device Server** window will be empty. To add Device Servers to the Device Server **List**, you can do either of the following:

- Click the **Search Local Network** button. This searches the local network segment and automatically displays any Device Servers it finds. Any Device Servers found by this method will be displayed in **Type** column as **Dynamic**. Once you close the DeviceManager, any Device Servers that were displayed as **Dynamic** will not be there until you click the **Search Local Network** button again.
- Click the **Static Server List** button to add Device Servers to the Device Server **List** permanently. This also allows you to add Device Servers that are not found on the local network segment with the **Search Local Network** button. To connect to a Device Server that is not in the Device Server List and resides outside the local network, see ***Adding/Deleting Static Device Servers*** on page 138.

For more information about managing a Device Server, see ***Managing a Device Server*** on page 139.

Assigning a Temporary IP Address to a New Device Server

You can temporarily assign an IP address to a Device Server that is connected to your local network segment, for the purpose of connecting to it and downloading a configuration file (containing a permanent IP address). To temporarily assign an IP address to a Device Server, do the following:

1. Click the **Search Local Network** button. The Device Server will be displayed in the **IP Address** column as **Not Configured**.
2. Select the new Device Server and click the **Assign IP** button. The following window is displayed:

Version 3.1 or lower

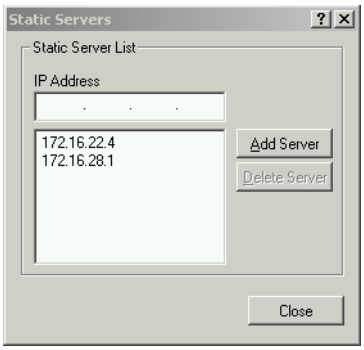
Version 3.2 or higher

3. Type a valid temporary IP address into the address field or, in version 3.2 or higher, enable the **Have the Device Server automatically get a temporary IP address**. If you enable the temporary IP address, the Device Server will enable DHCP/BOOTP on your Device Server and attempt to get an IP address from the DHCP/BOOTP server (this will permanently enable DHCP/BOOTP in your Device Server's configuration, until you change it). If your network does not have a DHCP/BOOTP server, the Device Server will temporarily assign an IP address in the range of **`169.254.0.1`**-**`169.254.255.255`** (this IP address is only assigned for the duration of the DeviceManager/Device Server connection).
4. Click the **Assign IP** button.
5. Double-click the Device Server in the Device Server **List**. If this is the first time you are accessing the Device Server, type in the factory default Admin password, **`superuser`**, and click **OK**. The DeviceManager will display a window indicating that it is trying to authenticate and connect you on the Device Server.
6. If the authentication and connection are successful, the Server Info window is displayed. You are now ready to configure the Device Server. If authentication was unsuccessful, try to connect to the Device Server again; you probably mistyped the password for the Admin user.

For more information about managing a Device Server, see ***Managing a Device Server*** on page 139.

Adding/Deleting Static Device Servers

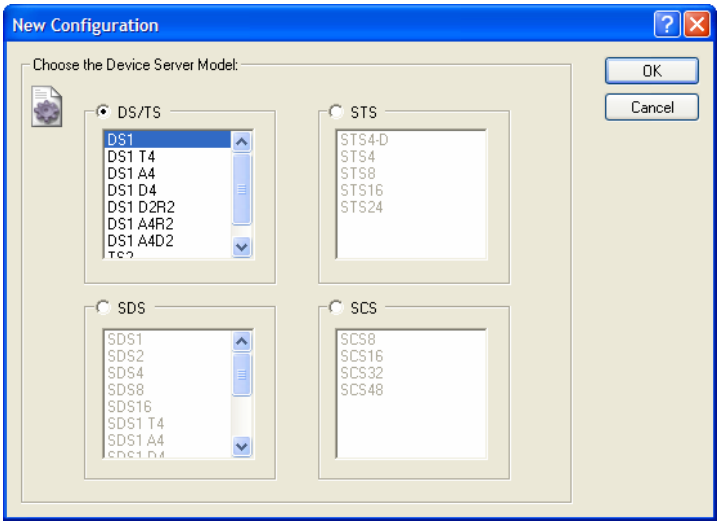
To permanently add or delete a Device Server to/from the Device Server **List**, select the **Static Server List** button. The following window is displayed:



To permanently add a Device Server to the Device Server **List**, type in the IP address of the Device Server and click the **Add Server** button. To permanently delete a Device Server from the Device Server **List**, select the Device Server's IP address and click the **Delete Server** button.

Creating a New Device Server Configuration

If you selected the **New Configuration** radio button, the New Configuration window is displayed.



Select the Device Server model for which you want to create a new configuration file.

Opening an Existing Configuration File

If you selected the **Open Configuration** radio button, a browse window is opened so you can select the configuration file you want to edit. Device Server configuration files can be in the Device Server-native binary format (**`.dme`**) or as a text file (**`.txt`**), which can be edited with a text editor. Either configuration version can be imported into the DeviceManager.

Connecting to a Device Server

To connect to a Device Server, double-click on the Device Server in the **Device Server List**. You will be prompted for the Admin Password.

If the authentication and connection are successful, the Device Server's **Server Info** window is displayed.

If you cannot connect to a Device Server, you can highlight the Device Server and click the **Ping** button to verify that the DeviceManager can communicate with the Device Server's IP Address. If the ping times out, then you might need to set up a Gateway in your Device Server or verify that your network is communicating correctly.

Managing a Device Server

Once you are connected to a Device Server, you can edit its configuration, download a new configuration, save the configuration to file, perform administrative tasks, and view statistics about the Device Server and its network environment.

DeviceManager Work Flow

When you connect to a Device Server, the Device Server's configuration is automatically uploaded to the DeviceManager. Before you make any changes to the configuration, you probably want to save the configuration locally, to make a backup file of the configuration. Use the navigation panel to select the feature that you want to edit. After you make all your changes to a configuration window, you must click the **Apply** button to submit those changes. When you have completed all of your configuration edits, select **Tools**, **Download Configuration to Unit**. If you want your changes to take effect at this point, select **Tools**, **Reboot Server**.

Creating/Editing Configuration Files

You can create and edit Device Server configuration files. When you create a new configuration file, the configuration file contains the Device Server's factory default settings.

Working With the Device Server Configuration

When you connect to a Device Server, the configuration that is saved to FLASH is automatically uploaded to the DeviceManager. It is suggested that you save the working configuration to a file as a backup precaution by selecting **Tools**, **Save Configuration to File**. You can then make any edits to the configuration and download it back to the Device Server by selecting **Tools**, **Download Configuration to Unit**. The downloaded configuration does not take effect until you reboot the Device Server by selecting **Tools**, **Reboot Server**. If you want to continue managing/configuring the Device Server, you can reconnect to the Device Server after it has been rebooted.

Working With a Local Configuration File

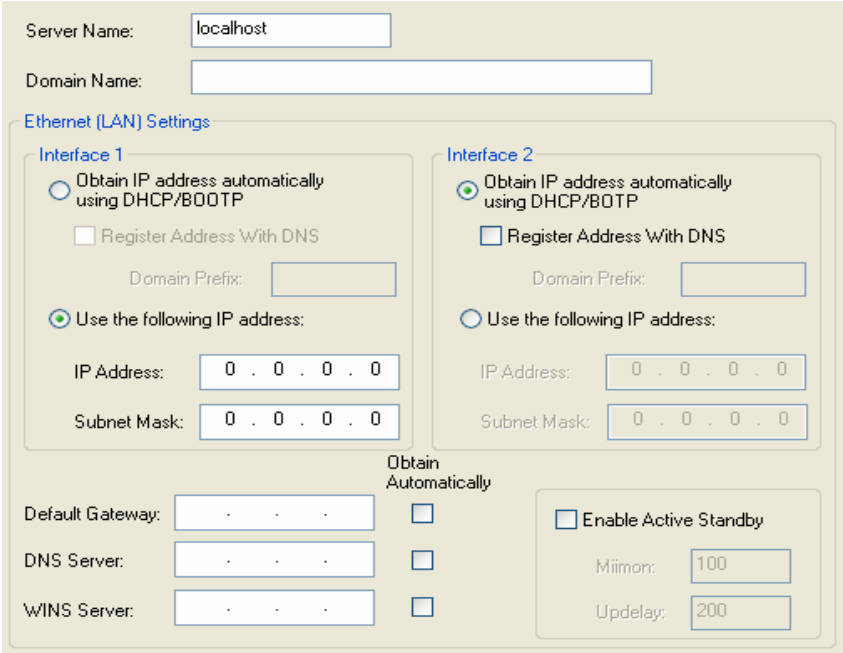
You can also connect to a Device Server and open a saved configuration file by selecting **Tools**, **Get Configuration**, **Import from File**. This configuration can then be edited or just downloaded right to the Device Server by selecting **Tools**, **Download Configuration to Unit**. The downloaded configuration does not take effect until you reboot the Device Server by selecting **Tools**, **Reboot Server**. If you want to continue managing/configuring the Device Server, you can reconnect to the Device Server after it has been rebooted.

Configuring the Server

The following sections describe how to configure the Device Server's server parameters. In each of the sections, the SCS8 model is being used; this means that what you configure can differ from what's shown, although all of your configuration parameters will be explained.

Configuring the Main Server Window

When you select **Server Configuration**, **Server** from the navigation panel, the following Server window is displayed.



Enter values in the Device Server parameters that identify the Device Server to your network.

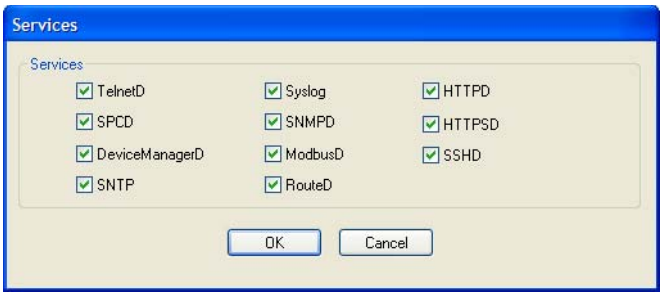
Server

Server Name	You must supply a name for the Device Server.
Domain Name	Unique name for your domain, your location in the global network. Like Hostname, it is a symbolic, rather than a numerical, identifier. See ***IPv6 Network*** on page 57 for information on how to determine your IPv6 address.
Obtain IP address automatically using DHCP/BOOTP	Enables the DHCP/BOOTP client process in the Device Server. By default, this is disabled/off. If this is enabled, the server IP address parameter is disabled.

Register Address With DNS	The DHCP server will update the DNS server when the Device Server requests a DHCP IP address (the communication between the DNS server and the DHCP server must already be set up in your network).
Domain Prefix	(SCS models only) A domain prefix to uniquely identify the Ethernet interface to the DNS when the Device Server has two Ethernet interfaces. The format of the Ethernet interface will take the form of *<Server Name>.<Domain Prefix>.<Domain Name>* or *<Server Name>.<Domain Prefix>*, depending on what is configured.
Use the following IP Address	When you select this option, you must supply an IP Address and Subnet Mask for the Device Server. By default, this is enabled.
IP Address	The Device Server's unique IPv4 network IP address for the second Ethernet connection. If you are using the Device Server in an IPv6 network, this field can be left blank. SCS models support two IP addresses, one for each Ethernet connection.
Subnet Mask	The network subnet mask. For example, 255.255.0.0.
Enable Active Standby	(SCS only) Enables/disables the feature of automatically assigning the Ethernet 1 IP address to Ethernet 2 if Ethernet 1 should fail to communicate to the network.
Default Gateway	You can specify the Default Gateway IP address to provide general access beyond your local network.
Default Gateway Obtain Automatically	When DHCP/BOOTP is enabled, you can enable this option to have the Device Server receive the Default Gateway IP address from the DHCP/BOOTP server.
DNS Server	You can specify the IP addresses of a DNS (Domain Name Server) host in your network.
DNS Server Obtain Automatically	When DHCP/BOOTP is enabled, you can enable this option to have the Device Server receive the DNS IP address from the DHCP/BOOTP server.
WINS Server	You can specify the IP addresses of a WINS (Windows Internet Naming Service) host in your network.
WINS Server Obtain Automatically	When DHCP/BOOTP is enabled, you can enable this option to have the Device Server receive the WINS IP address from the DHCP/BOOTP server.
Miimon	(SCS only) The interval in which the active interface is checked to see if it is still communicating. The default is 100 ms.
Updelay	(SCS only) The time that the Device Server will wait to make the secondary interface (Ethernet 2) active after it has been detected as up.

Services

Services are either daemon or client processes that run on the Device Server. You can disable any of the services for security reasons. If you disable the DeviceManagerD service, you will not be able to use DeviceManager to connect to a Device Server.

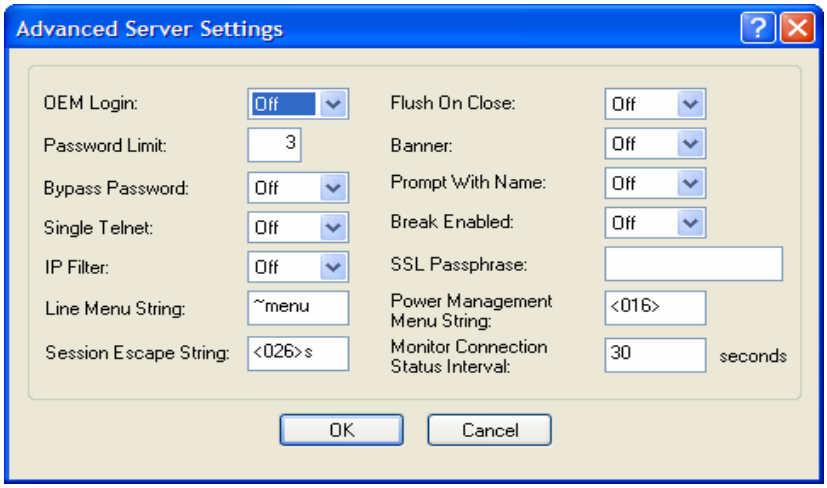


Configure the appropriate parameters:

TelnetD	Telnet daemon process in the Device Server on port 23.
SPCD	SPC (TruePort) daemon process in the Device Server that supports TruePort Full Mode on UDP port 668. You can still communicate with the Device Server in Light Mode when this service is disabled.
DeviceManagerD	DeviceManager daemon process in the Device Server. If you disable this service, you will not be able to connect to the Device Server with the DeviceManager application. DeviceManagerD listens on port 33812 and sends on port 33813.
SNTP	SNTP client process in the Device Server.
Syslog	Syslog client process in the Device Server.
SNMPD	SNMP daemon process in the Device Server on port 161.
MODBUSD	Modbus daemon process in the Device Server on port 502.
RouteD	Route daemon process in the Device Server on port 520.
HTTPD	HTTP daemon process in the Device Server on port 80.
HTTPSD	HTTPS daemon process in the Device Server on port 443.
SSHD	SSH daemon process in the Device Server on port 22.

Configuring Advanced Server Settings

In the Server window, the following window is displayed when you click the Advanced button.



Configure the appropriate parameters:

OEM Login

When set, and a custom language file is in use, the login prompt will use the string defined in the language file as the login prompt instead of the default prompt, `login:`.

Password Limit

The number of authentication attempts a user is allowed for a serial port connection (this applies to **Line Service DSLogin** and Console mode connections). If this limit is exceeded, the port is disabled for 5 minutes. A user with Admin level rights can restart the port, bypassing the timeout, by issuing a kill on the disabled port. The default value is **3**.

Bypass Password

When set, authorised users who do not have a password set, with the exception of the Admin user, WILL NOT be prompted for a password at login with **Local Authentication**.

Single Telnet

Sets all reverse connections (raw, SSH, and telnet) to a one connection at a time mode. Server-side applications will get a (socket) connection refused until:

- All data from previous connections on that serial port has drained
- There are no other connections
- Up to a 1 second interconnection poll timer has expired

This also enables a per-connection keepalive TCP keepalive feature. After approximately 3 minutes of network connection idle time, the connection will send a gratuitous ACK to the network peer, thus either ensuring the connection stays active OR causing a dropped connection condition to be recognised by the reverse service (all connections).

Applications using Single Telnet need to be aware that there can be some considerable delay between a network disconnection and the port being available for the next connection attempt; this is to allow any data sent on prior connections to be transmitted out of the serial port. Application network retry logic needs to accommodate this feature. The default value is **Off**.

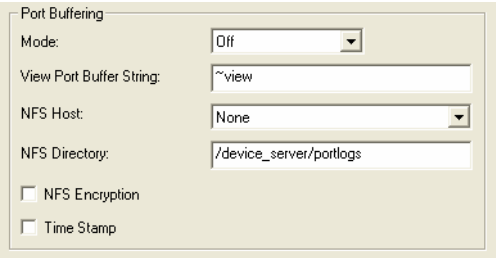
IP Filter

A security feature that when enabled, the Device Server will only accept data from hosts configured in the Device Server's **Host Table** with an IP address (hosts configured with a Fully Qualified Domain Name, FQDN, will not be able to access the Device Server when this option is enabled). The default value is **Off**.

Line Menu String	The string used to access to the Easy Port Access menu without disconnecting the initial reverse SSH or reverse Telnet session.The default string is **~menu**.
Session Escape String	A configurable string that allows access to a port to view the multisession screen options, allowing the various options while accessing the particular port on the Device Server. You can specify control (unprintable) codes by putting the decimal value in angle brackets < > (for example, **ESC-b** is **<027>b**). The default value is **Ctrl-z s** (**<026>s** in decimal).
Flush On Close	When enabled, deletes any pending data when a port is closed; as opposed to maintaining the port to send pending data. The default value is **Off**.
Banner	This parameter concerns the banner information (product name/software version). This banner information is presented to a user with a login prompt. For security reasons, you can turn off the display of this information. The default is **Off**.
Prompt With Name	Displays the **Server Name** field value instead of default product name. When enabled, the **Server Name** is displayed in the Device Server login prompt, CLI prompt, WebManager login screen, and the heading of the Menu. The default value is **Off**.
Break Enabled	Enables/disables proprietary inband SSH break signal processing as well as the existing Reverse Telnet break signal. This parameter can also enable/disable the out-of-band break signals for TruePort. The default value is **Off**.
SSL Passphrase	This is the SSL/TLS passphrase used to generate an encrypted RSA/DSA private key. This private key and passphrase are required for both HTTPS and SSL/TLS connections, unless an unencrpyted private key was generated, then the SSL passphrase is not required. Make sure that you download the SSL private key and certificate if you are using the secure HTTP option (HTTPS) or SSL/TLS. If both RSA and DSA private keys are downloaded to the Device Server, they need to be generated using the same SSL passphrase for both to work.
Power Management Menu String	Users accessing the Device Server through reverse sessions can enter the string to bring up a power bar management menu. This is a decimal value. The default value is **<016>** or **Ctrl-p** on the keyboard.
Monitor Connection Status Interval	Specify how often, in seconds, the Device Server will send a TCP Keepalive to services that support TCP Keepalive. The default is 30 seconds.

Configuring Port Buffering

Port buffering displays or logs data received on the Device Server serial port.



Configure the following parameters:

Mode

Specifies where the port buffer log is kept, either **Off**, **Local**, **Remote**, or **Both**. If **Remote** or **Both** is selected, you must specify an NFS server location for the port buffer log.

View Port Buffering String

The string (up to 8 characters) used by a a session connected to a serial port to display the port buffer for that particular serial port. You can specify control (unprintable) codes by putting the decimal value in angle brackets < > (for example, **Escape b** is **<027>b**). The default is **~view**.

NFS Host

The NFS host that the Device Server will use for its **Remote Port Buffering** feature. The Device Server will open a file on the NFS host for each reverse SSH or reverse Telnet line, and send any port data to be written to those files. The default is **None**. This field is required when **Mode** is set to **Remote** or **Both**.

NFS Directory

The directory and/or subdirectories where the **Remote Port Buffering** files will be created. This field is used when Port Buffering **Mode** is set to **Remote** or **Both**. For multiple Device Servers using the same NFS host, it is recommended that each Device Server have its own unique directory to house the remote port log files. The default is **`/device_server/portlogs`**.

NFS Encryption

Determines if the data sent to the NFS host is sent encrypted or in the clear across the LAN. The default is set of **Off**.

NOTE: When NFS encryption is enabled, the Decoder utility software is required to be installed on the NFS host for decrypting the data to a readable format. The Decoder utility software can be found on the installation CD-ROM and on the **`www.perle.com`** website.

Time Stamp

Enable/disable time stamping of the port buffer data.

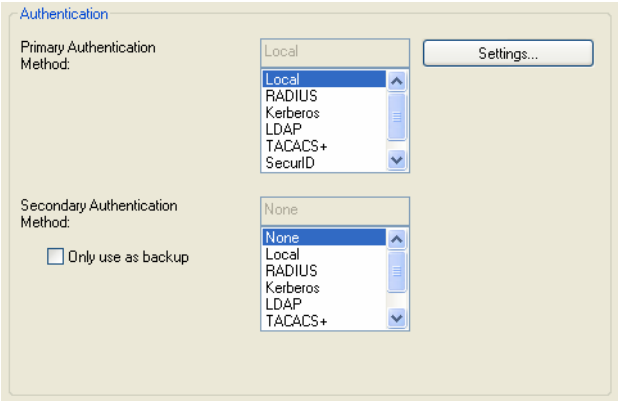
Configuring TruePort Baud

The TruePort Baud configuration window allows you to map the baud rate configured on the UNIX system running the TruePort client to another baud rate that will run between the Device Server and the serial device. See Appendix E, ***Utilities*** on page 377 for more information about TruePort.

Configuring Authentication

The Device Server can authenticate a user locally or through an external authentication server, based on **User Name** and **Password**, or locally by just a **Password** when the **Guest** option is enabled. This is different from authorization, which can restrict a user's access to the network (although this can be done through the concept of creating sessions for a user, see ***Sessions*** on page 94 for more information on user sessions). All authentication does is ensure that the user is defined within the authentication database—unless you are using RADIUS or TACACS+, which can also send back **User** and **Line** parameters (see Appendix A, ***RADIUS*** on page 353 for more information about RADIUS or Appendix B, ***TACACS+*** on page 361 for more information about TACACS+).

When you select **Configuration**, **Server Configuration**, **Authentication** from the navigation panel, the Authentication window is displayed.



You can select a **Primary Authentication Method**, which is the first method the Device Server will use to authenticate the user. If that authentication method fails (due to connection problems or the user authentication fails), then the Device Server will attempt to authenticate the user by the **Secondary Authentication Method**, if one is selected and configured. The user will be prompted to enter a password for each authentication method tried. You can choose to use the **Secondary Authentication Method** as a backup only (enable the **Only use as backup** option), in which case the secondary authentication method will be tried only when the Device Server cannot communicate with the primary authentication host (if the user fails to be authenticated by the primary authentication host, the user will be denied access to the Device Server).

The next sections describe the parameters that must be configured for each authentication method.

Local

When **Local** authentication is selected, the user must either be configured in the Device Server's **User List** or you must enable **Guest** users.

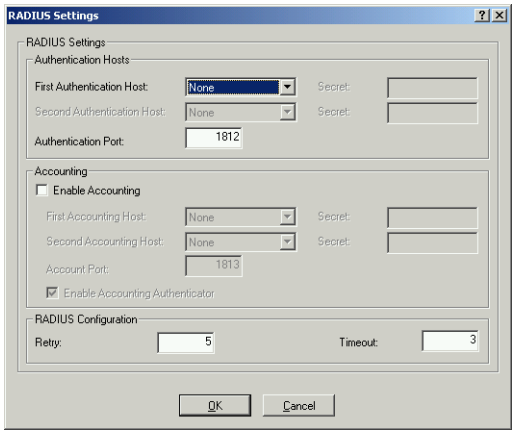
Configure the following parameters:

Enable Guest Mode	Allow users who are not defined in the **User** database to log into the Device Server with any user ID and the specified password. **Guest** users inherit their settings from the **Default User**'s configuration.
Guest Password	The password that **Guest** users must use to log into the Device Server.
Confirm Password	Type the **Guest Password** in again to verify that it is correct.

RADIUS

RADIUS is an authentication method that the Device Server supports that can send back **User** information; see Appendix A, ***RADIUS*** on page 353 for more information on the **User** parameters that can be sent back by RADIUS.

The RADIUS configuration window is displayed when you click on **RADIUS Settings** button.



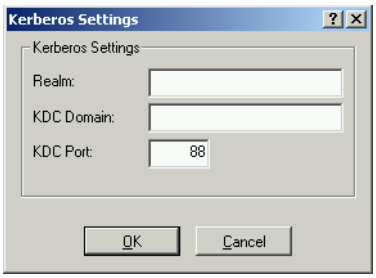
Configure the following parameters:

First Authentication Host	Name of the primary RADIUS authentication host.
Second Authentication Host	Name of the secondary RADIUS authentication host.
Secret	The secret (password) shared between the Device Server and the RADIUS authentication host.
Authentication Port	The port that the RADIUS host listens to for authentication requests. The default port is 1812.
Enable Accounting	Enables/disables RADIUS accounting.

First Accounting Host	Name of the primary RADIUS accounting host.
Second Accounting Host	Name of the secondary RADIUS accounting host.
Secret	The secret (password) shared between the Device Server and the RADIUS accounting host.
Account Port	The port that the RADIUS host listens to for accounting requests. The default port is 1813.
Enable Accounting Authenticator	Enables/disables whether or not the Device Server validates the RADIUS accounting response.
Retry	The number of times the Device Server tries to connect to the RADIUS server before erroring out. Valid values are 0-255. The default is **5**.
Timeout	The time, in seconds, that the Device Server waits to receive a reply after sending out a request to a RADIUS accounting or authentication host. If no reply is received before the timeout period expires, the Device Server will retry the same host up to and including the number of retry attempts. Valid values are 1-255. The default is **3** seconds.

Kerberos

The Kerberos configuration window is displayed when you click on **Kerberos Settings** button.

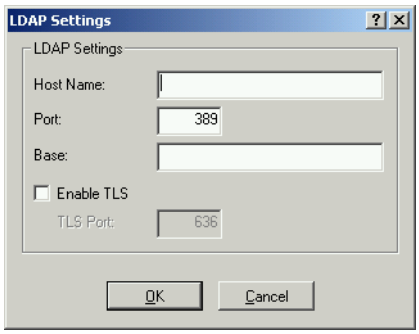


Configure the following parameters:

Realm	The Kerberos realm is the Kerberos host domain name, in upper-case letters.
KDC Domain	The name of a host running the KDC (Key Distribution Center) for the specified realm. The host name that you specify must either be defined in the Device Server's **Host Table** before the last reboot or be resolved by DNS.
KDC Port	The port that the Kerberos server listens to for authentication requests. If no port is specified, the default port 88 is used.

LDAP

The LDAP configuration window is displayed when you click on **LDAP Settings** button. If you are using LDAP with **TLS**, you need to download an SSL/TLS certificate to the Device Server by selecting **Tools**, **Keys and Certificates**. See ***Keys and Certificates*** on page 98 for more information on the LDAP certificate.

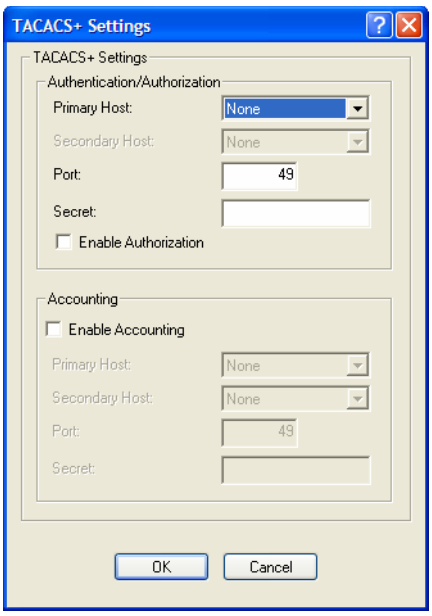


Configure the following parameters:

Host Name	The name or IP address of the LDAP host. If you use a host name, that host must either have been defined in the Device Server's **Host Table** before the last reboot or be resolved by DNS. If you are using **TLS**, you must enter the same string you used to create the LDAP certificate that resides on your LDAP server.
Port	The port that the LDAP host listens to for authentication requests. The default port is 389.
Base	The domain component (dc) that is the starting point for the search for user authentication.
Enable TLS	Enables/disables the Transport Layer Security (TLS) with the LDAP host.
TLS Port	Specify the port number that LDAP will use for **TLS**. The default is port 636.

TACACS+

TACACS+ is an authentication method that the Device Server supports that can send back **User** information; see Appendix B, ***TACACS+*** on page 361 for more information on the **User** parameters that can be sent back by TACACS+.The TACACS+ configuration window is displayed when you click on **TACACS+ Settings** button.



Configure the following parameters:

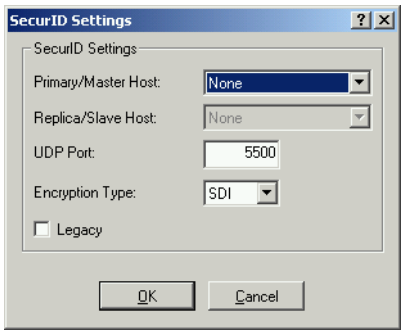
Authentication/ Authorization Primary Host	The primary TACACS+ host that is used for authentication.
Authentication/ Authorization Secondary Host	The secondary TACACS+ host that is used for authentication, should the primary TACACS+ host fail to respond.
Authentication/ Authorization Port	The port number that TACACS+ listens to for authentication requests. The default port number is 49.
Authentication/ Authorization Secret	The TACACS+ shared secret is used to encrypt/decrypt TACACS+ packets in communications between two devices. The shared secret may be any alphanumeric string. Each shared secret must be configured on both client and server sides.
Enable Authorization	Enables authorization on the TACACS+ host, meaning that Device Server-specific parameters set in the TACACS+ configuration file can be passed to the Device Server after authentication.
Enable Accounting	Enables/disables TACACS+ accounting.
Accounting Primary Host	The primary TACACS+ host that is used for accounting.
Accounting Secondary Host	The secondary TACACS+ host that is used for accounting, should the primary accounting TACACS+ host fail to respond.

Accounting Port The port number that TACACS+ listens to for accounting requests. The default port number is 49.

Accounting Secret The TACACS+ shared secret is used to encrypt/decrypt TACACS+ packets in communications between two devices. The shared secret may be any alphanumeric string. Each shared secret must be configured on both client and server sides.

SecurID

The SecurID configuration window is displayed when you click on **SecurID Settings** button. If you need to reset the SecurID secret, select **Tools**, **Reset SecurID Node Secret**.



Configure the following parameters:

Primary/Master Host The first SecurID server that is tried for user authentication.

Replica/Slave Host If the first SecurID server does not respond to an authentication request, this is the next SecurID server that is tried for user authentication.

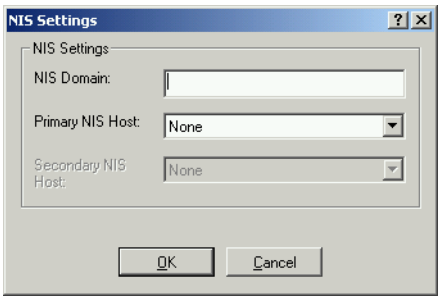
UDP Port The port number that SecurID listens to for authentication requests. The default port number is 5500.

Encryption Type You can specify either **SDI** or **DES** encryption for SecurID server communication. The default is **SDI** encryption.

Legacy If you are running SecurID 3.x or 4.x, you need to run in **Legacy Mode**. If you are running SecurID 5.x or above, do not select **Legacy Mode**.

NIS

The NIS configuration window is displayed when you click on **NIS Settings** button.



Configure the following parameters:

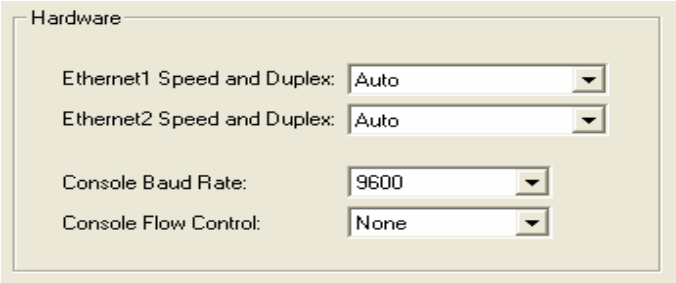
NIS Domain The NIS domain name.

Primary NIS Host The primary NIS host that is used for authentication.

Secondary NIS Host The secondary NIS host that is used for authentication, should the primary NIS host fail to respond.

Configuring the Hardware

You need to configure the Ethernet interface that you are using to connect the Device Server to the LAN.



Select the appropriate option:

Ethernet1 Speed and Duplex Define the Ethernet connection speed at one of the following (desktop models don't support 1000 Mbps):

- **auto**—automatically detects the Ethernet interface speed and duplex
- **10 Mbps Half Duplex**
- **10 Mbps Full Duplex**
- **100 Mbps Half Duplex**
- **100 Mbps Full Duplex**
- **1000 Mbps Half Duplex**
- **1000 Mbps Full Duplex**

Ethernet2 Speed and Duplex Define the Ethernet connection speed at one of the following (Available on SCS models only):

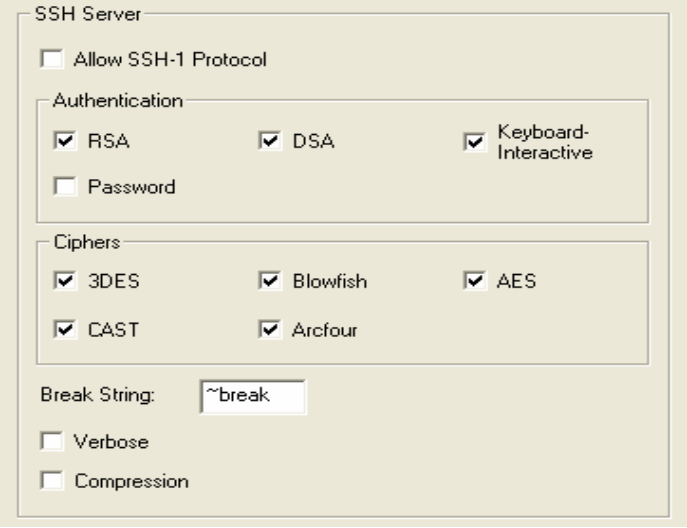
- **auto**—automatically detects the Ethernet interface speed and duplex
- **10 Mbps Half Duplex**
- **10 Mbps Full Duplex**
- **100 Mbps Half Duplex**
- **100 Mbps Full Duplex**
- **1000 Mbps Half Duplex**
- **1000 Mbps Full Duplex**

Console Baud Rate For Device Server models that have a dedicated console port, specifies the baud rate of the line connected to the console port.

Console Flow Control For Device Server models that have a dedicated console port, defines whether the data flow is handled by using software (**Soft**), hardware (**Hard**), or no (**None**) flow control.

Configuring the SSH Server

The Device Server contains SSH Server software that you need to configure if the Device Server is going to be accessed via SSH. If you specify more than one **Authentication** method and/or **Cipher**, the Device Server will negotiate with the client and use the first authentication method and cipher that is compatible with both systems.



Configure the following parameters:

Allow SSH-1 Protocol Allows the user's client to negotiate an SSH-1 connection, in addition to SSH-2.

RSA When a client SSH session requests RSA authentication, the Device Server's SSH server will authenticate the user via RSA.

DSA When a client SSH session requests DSA authentication, the Device Server's SSH server will authenticate the user via DSA.

Keyboard-Interactive The user types in a password for authentication.

Password The user types in a password for authentication.

3DES The Device Server SSH server's 3DES encryption is enabled/disabled.

CAST The Device Server SSH server's CAST encryption is enabled/disabled.

Blowfish The Device Server SSH server's Blowfish encryption is enabled/disabled.

Arcfour The Device Server SSH server's Arcfour encryption is enabled/disabled.

AES The Device Server SSH server's AES encryption is enabled/disabled.

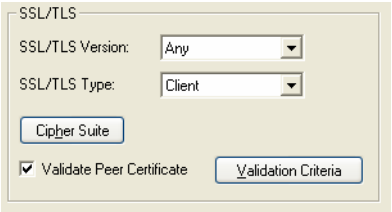
Breakstring The break string used for inband SSH break signal processing. A break signal is generated on a specific serial port only when the server's break option is enabled and the user currently connected using reverse SSH has typed the break string exactly. The default is set to **~break**, where **~** is tilde; the break string can be up to eight characters.

Verbose Displays debug messages on the terminal.

Compression Requests compression of all data. Compression is desirable on modem lines and other slow connections, but will only slow down things on fast networks.

SSL/TLS Settings

When you configure the **SSL/TLS** settings in the **Server** section, you are actually configuring the default SSL/TLS settings; you are not configuring an SSL/TLS server.



Configure the following parameters:

Enable SSL/TLS Activates the SSL/TLS settings for the line.

User Server Settings Uses the SSL/TLS server configuration for the line.

SSL/TLS Version Specify whether you want to use:

- **Any**—The Device Server will try a TLSv1 connection first. If that fails, it will try an SSLv3 connection. If that fails, it will try an SSLv2 connection.
- **TLSv1**—The connection will use only TLSv1.
- **SSLv3**—The connection will use only SSLv3.

The default is **Any**.

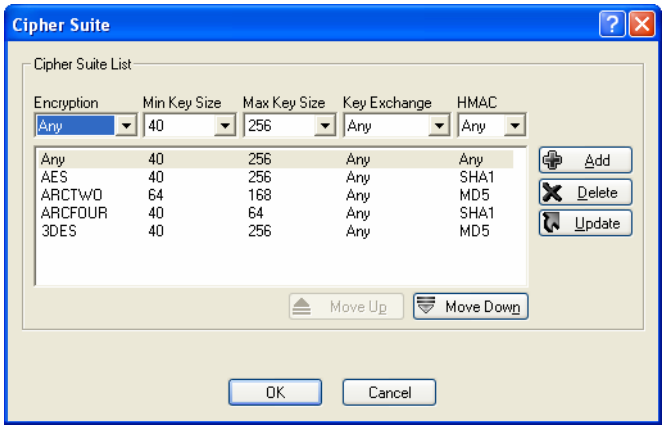
SSL/TLS Type Specify whether the Device Server will act as an SSL/TLS client or server. The default is **Client**.

Validate Peer Certificate Enable this option when you want the Validation Criteria to match the Peer Certificate for authentication to pass. If you enable this option, you need to download an SSL/TLS certificate authority (CA) list file to the Device Server.

For more information, see ***Keys and Certificates*** on page 98.

Cipher Suite

You can set up cipher rules to govern the encryption that will be used for the SSL/TLS connection.



Configure the following parameters:

Encryption

Select the type of encryption that will be used for the SSL connection:

- Any—Will use the first encryption format that can be negotiated.
- AES
- 3DES
- DES
- ARCFOUR
- ARCTWO

The default value is **Any**.

Min Key Size

The minimum key size value that will be used for the specified encryption type. The default is **40**.

Max Key Size

The maximum key size value that will be used for the specified encryption type. The default is **256**.

Key Exchange

The type of key to exchange for the encryption format:

- **Any**—Any key exchange that is valid is used (this does not, however, include ADH keys).
- **RSA**—This is an RSA key exchange using an RSA key and certificate.
- **EDH-RSA**—This is an EDH key exchange using an RSA key and certificate.
- **EDH-DSS**—This is an EDH key exchange using a DSA key and certificate.
- **ADH**—This is an anonymous key exchange which does not require a private key or certificate. Choose this key if you do not want to authenticate the peer device, but you want the data encrypted on the SSL/TLS connection.

The default is **Any**.

HMAC

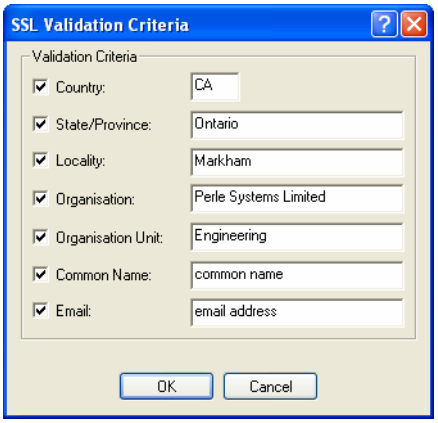
Select the key-hashing for message authentication method for your encryption type:

- Any
- MD5
- SHA1

The default is **Any**.

Validation Criteria

If you choose to configure validation criteria, then the information in the peer SSL/TLS certificate must match exactly the information configured in this window in order to pass peer authentication and create a valid SSL/TLS connection.



Configure the following parameters:

Country — A two character country code; for example, US. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

State/Province — Up to a 128 character entry for the state/province; for example, IL. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Locality — Up to a 128 character entry for the location; for example, a city. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Organisation — Up to a 64 character entry for the organisation; for example, Accounting. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

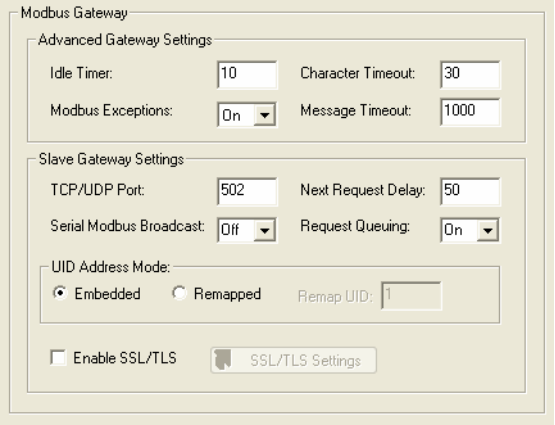
Oraganisation Unit — Up to a 64 character entry for the unit in the organisation; for example, Payroll. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Common Name — Up to a 64 character entry for common name; for example, the host name or fully qualified domain name. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Email — Up to a 64 character entry for an email address; for example, acct@anycompany.com. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Configuring the Modbus Gateway

The Advanced Gateway Settings are global to both Master and Slave Modbus Gateways. There are no parameters that are specific to only the Modbus Master Gateway, as there is for the Modbus Slave Gateway. For more information on Modbus Gateways, see ***Modbus Gateway Settings*** on page 80.



Configure the following parameters:

Idle Timer Specifies the number of seconds that must elapse without any network or serial traffic before a connection is dropped. If this parameter is set to 0 (zero), a connection will not be dropped (with the following exceptions: the TCP KeepAlive causes the connection to be dropped or the Modbus device drops the connection). The default is **10** seconds.

Modbus Exceptions When enabled, an exception message is generated and sent to the initiating Modbus device when any of the following conditions are encountered: there is an invalid UID, the UID is not configured in the Gateway, there is no free network connection, there is an invalid message, or the target device is not answering the connection attempt. The default is **On**.

Character Timeout Used in conjunction with the Modbus RTU protocol, specifies how long to wait, in milliseconds, after a character to determine the end of frame. The default is **30** ms.

Message Timeout Time to wait, in milliseconds, for a response message from a Modbus TCP or serial slave (depending if the Modbus Gateway is a Master Gateway or Slave Gateway, respectively) before sending a Modbus exception. The default is **1000** ms.

TCP/UDP Port The network port number that the Slave Gateway will listen on for both TCP and UDP messages. The default is **502**.

Serial Modbus Broadcast When enabled, a UID of 0 (zero) indicates that the message will be broadcast to all Modbus Slaves. The default is **Off**.

Next Request Delay A delay, in milliseconds, to allow serial slave(s) to re-enable receivers before issuing next Modbus Master request. The default is **50** ms.

Request Queuing When enabled, allows multiple, simultaneous messages to be queued and processed in order of reception. The default is **On**.

Embedded When this option is selected, the address of the slave Modbus device is embedded in the message header.

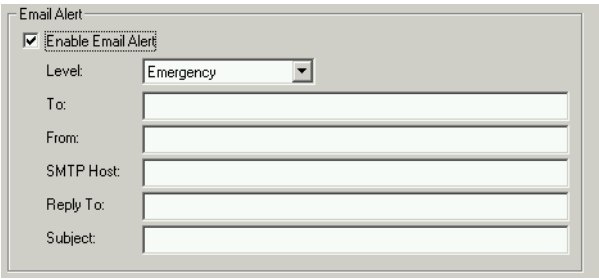
Remapped Used for single device/port operation. Older Modbus devices may not include a UID in their transmission header. When this option is selected, you can specify the UID that will be inserted into the message header for the Modbus slave device. This feature supersedes the Broadcast feature.

Remap UID Specify the UID that will be inserted into the message header for the Slave Modbus serial device. Valid values are 1-247.

Enable SSL/TLS When enabled, Modbus Slave Gateway messages to remote TCP Modbus Masters are encrypted via SSL/TLS.

Configuring Server Email Alerts

You can configure server email alerts, emails that are sent to specified recipients when an event occurs at the specified level.



Configure the following parameters:

Enable Email Alert Determines whether or not email notification is turned on. Default is **Off**.

Level Choose the event level that triggers an email notification:

- **Emergency**
- **Alert**
- **Critical**
- **Error**
- **Warning**
- **Notice**
- **Info**
- **Debug**

You are selecting the lowest notification level; therefore, when you select **Debug**, you will get an email notification for all events that trigger a message.

To An email address or list of email addresses that will receive the email notification.

From This field can contain an email address that might identify the Device Server name or some other value.

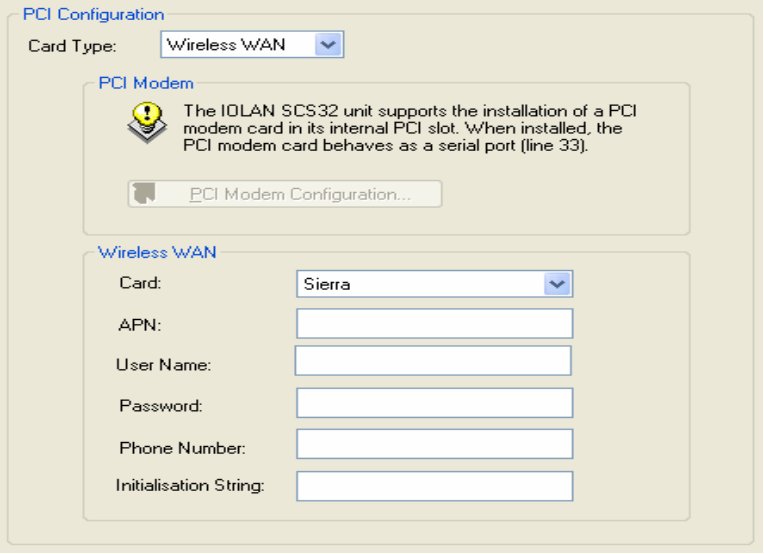
SMTP Host The SMTP host (email server) that will process the email notification request. This can be either a host name defined in the Device Server host table or the SMTP host IP address.

Reply To The email address to whom all replies to the email notification should go.

Subject A text string, which can contain spaces, that will display in the **Subject** field of the email notification.

PCI Configuration

If you have an SCS model, there is an optional internal PCI modem card (see ***Configuring Lines*** on page 163 to find out more about the configuration options) or a wireless WAN card that can be installed.



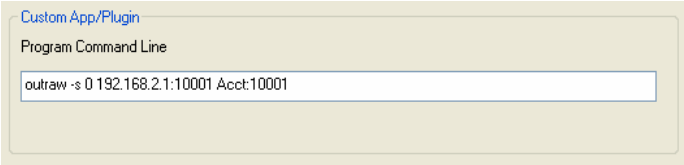
Click the **PCI Modem Configuration** button to be taken to the **Line** configuration window for the PCI modem line.

If you are using a wireless WAN card, configure the following parameters:

Card	Specify the wireless WAN card you are using. If the wireless WAN card you are using is not listed, try the standard driver. If that does not work, look at the Perle website for a custom driver.
APN	Specify the APN required by your internet provider to access their network. See the internet provider documentation for more information.
User Name	Specify the name required by your internet provider to access their network.
Password	Specify the password required by your internet provider to access their network.
Phone Number	Specify the phone number provided by your service provider to access their wireless network. The phone number will probably take a format similar to **`*99***1#`**.
Initialisation String	Specify the initialisation string required by your internet service provider for your wireless WAN card.

Custom App/Plugin

You can create a custom application or use a custom plugin that can run on the Device Server using the Perle SDK.



Configure the following parameter:

Program Command Line

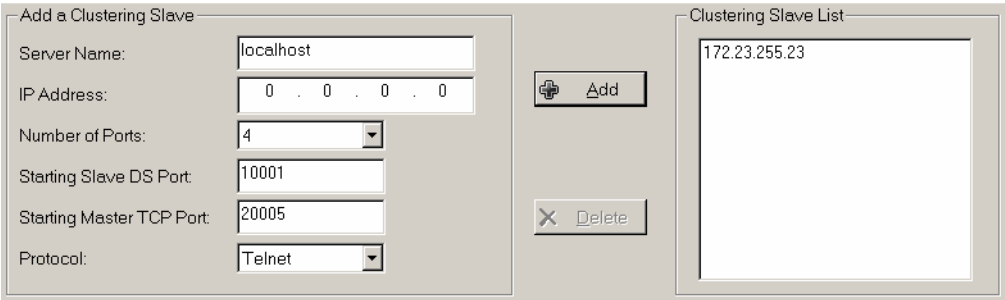
The name of the SDK program executable that has been already been downloaded to the Device Server, plus any parameters you want to pass to the program. Maximum of 80 characters. Use the **`shell`** CLI command as described in the *SDK Programmer's Guide* to manage the files that you have downloaded to the Device Server. For example, using sample **`outraw`** program, you would type:

`outraw -s 0 192.168.2.1:10001 Acct:10001`

if you were starting the application on the Server (notice the **`-s 0`** parameter specifies Line 1).

Clustering

Users can access Slave Device Servers through the Master Device Server by specifying the Master Device Server's IP address and Master TCP Port (which is mapped to the Slave Device Server's DS Port).



Add a Clustering Slave

Configure the following parameters:

Server Name

Specify a name for the Slave Device Server in the clustering group. This name does not have to correspond to the proper host name, as it is just used within the Device Server.

IP Address

Specify the IP address of the Slave Device Server in the clustering group. The IP address must be in a valid IPv4 format.

Number of Ports

Specify the number of ports in the Slave Device Server that you are adding to the clustering group.

Starting Slave DS Port

Specify the first DS Port number (as specified in the Slave Device Server's Line configuration) on the slave host. By default, this is 10001 and incriments by one for each line/port.

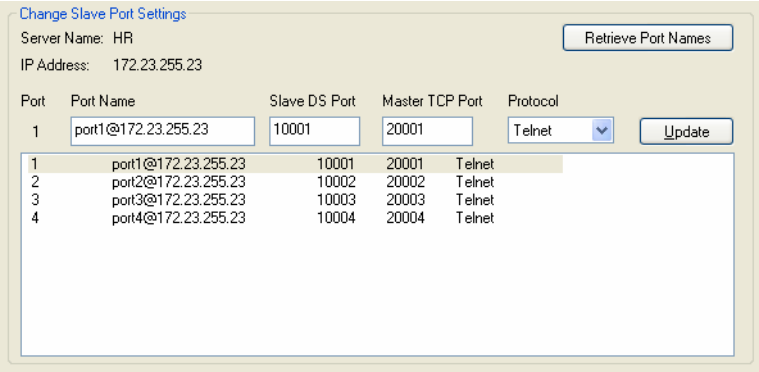
Starting Master TCP Port Specify the TCP port number you want to map the first Slave Device Server DS Port number to. This number should not be a port number that is already in use by the Master Device Server.

Protocol Specify the protocol that will be used to access the Slave Device Server port, SSH or Telnet.

Clustering Slave List Displays a list of the configured Slave Device Servers in the clustering group.

Change Slave Port Settings

After the Slave Device Server is added to the clustering group, you can configure each port individually.



Click on the Slave Device Server's port number and configure the following parameters:

Retrieve Port Names Gets the Port (Line) names from the Slave Device Server as they were configured on the Slave Device Server.

Port Name Specify a name for the port. The defualt name is a combination of the port number, the at symbol, and the IP address; for example, **`port1@172.22.23.101`**.

Slave DS Port Specify the DS Port number configured on the Slave Device Server that is associated to the port number you are configuring.

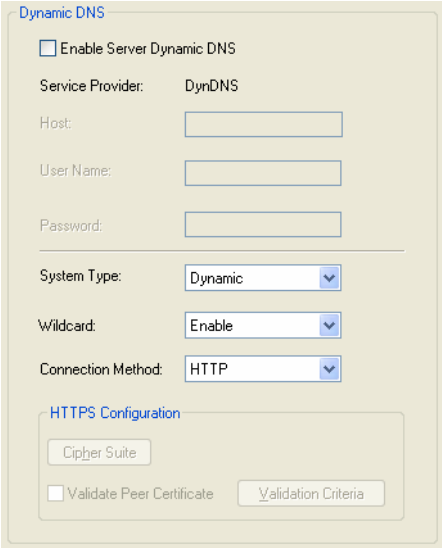
Master TCP Port Specify the TCP port number you want to map to the Slave Device Server DS Port. User's will use this TCP port number to access the Slave Device Server's port.

Protocol Specify the protocol that will be used to access the port, SSH, Telnet, or Not Used.

Update Button Updates the changes you have made to the Slave Device Server port.

Dynamic DNS

The Dynamic DNS feature will update DynDNS.org if the Device Server's IP address changes (the account must already be set up and the Device Server's host name must already be registered). See ***Dynamic DNS*** on page 124 for an explanation of how dynamic DNS works.



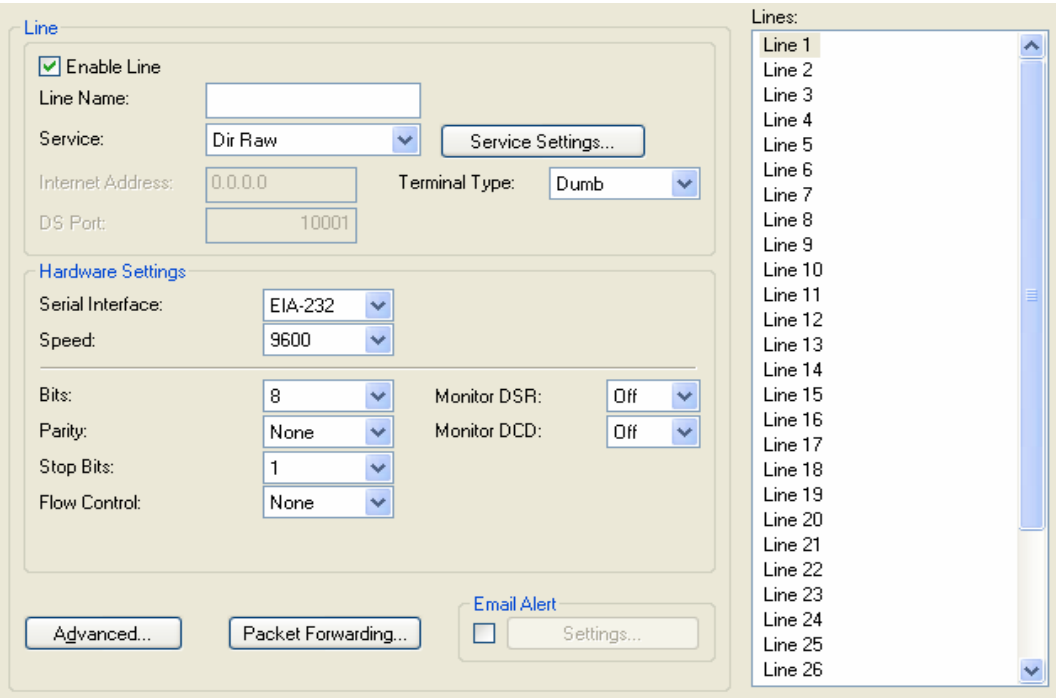
Configure the following parameters:

Enable Dynamic DNS	Enables/disables the dynamic DNS feature. When **Dynamic DNS** is enabled, the Device Server will automatically update its IP address with DynDNS.org if it changes.
Host	Specify the registered hostname with DynDNS.org that will be updated with the Device Server's IP address should it change. Put in the full name; for example, mydeviceserver.dyndns.org.
User Name	Specify the user name used to access the DynDNS.org server.
Password	Specify the password used to access the DynDNS.org server.
System Type	Specify how your account was set up with DynDNS.org, using a Dynamic, Static, or Custom IP address schema.
Wildcard	Adds an alias to **`*.yourhost.ourdomain.ext`** pointing to the same IP address as entered for **`yourhost.ourdomain.ext`**.
Connection Method	Specify how the Device Server is going to connect to the DynDNS.org server, via HTTP, HTTP through Port 8245, or HTTPS.

If you are using **HTTPS** as your **Connection Method**, see ***Cipher Suite*** on page 155 and/or ***Validation Criteria*** on page 156 for SSL/TLS configuration information.

Configuring Lines

When you configure the Device Server **Line**, you are specifying how the port will be used and accessed. You can always make changes to **Line** parameters by clicking the **Save Line Configuration** button, selecting **Tools**, **Download Configuration to Unit**, and then selecting **Tools**, **Kill Line** to have your line changes take effect permanently without having to reboot the Device Server. The example below shows a 4-port model; the 1-port and 2-port models do not have the **Save & Copy Line Configuration** button.



Configure the appropriate parameters:

Enable Line

Enables/disables a line (available only on 2-port+ models). The default is enabled.

Line Name

Provide a name for the line so it can be easily identified. The **Remote Port Buffering** logging feature uses the **Line Name** when creating a file on the remote NFS server.

Service

Defines the **Line Service**, which determines how the line will be used.

See ***Service Settings*** on page 168 for more information about configuring each **Line Service**.

Internet Address

Used with reverse sessions, users can access serial devices connected to the Device Server by the specified Internet Address (or host name that can be resolved to the Internet Address in a DNS network). You must reboot the Device Server for the **Internet Address** to take affect (the kill line option does not apply to this parameter). This parameter must be in IPv4 format.

DS Port

The Device Server port number.

Terminal Type

Specifies the type of terminal connected to the line:

- **Dumb**
- **WYSE60**
- **VT100**
- **ANSI**
- **TVI925**
- **IBM3151TE**
- **VT320** (specifically supporting VT320-7)
- **HP700** (specifically supporting HP700/44)
- **Term1, Term2, Term3** (user-defined terminals)

Serial Interface

Specifies the type of line that is being used with the Device Server. Select either **EIA-232**, **EIA-422**, or **EIA-485**. The SCS/STS models support only **EIA-232**.

Speed

Specifies the baud rate of the line; keep in mind that speed is affected by the length of the cable. You can also specify a custom baud rate; valid values are 50-230400.

Bits

Specifies the number of bits in a byte. The default is **8**.

Parity

Specifies if you are using **Even**, **Odd**, or **No parity** on the line. If you want to force a parity type, you can specify **Mark** for 1or **Space** for 0.

Stop Bits

Specifies the number of stop bits that follow a byte. The 1.5 option is only available on the 1-port and 2-port models, but not on the modem line (Line 2) of the SDS1M model.

Flow Control

Defines whether the data flow is handled by the software (**Soft**), hardware (**Hard**), **Both**, or **None**. If you are using **SLIP**, set to **Hard** only. If you are using **PPP**, set to either **Soft** or **Hard** (**Hard** is recommended). If you select **Soft** with **PPP**, you must set the **ACCM** parameter when you configure **PPP** for the **Line**.

Duplex

Specify whether the line is **Full Duplex** (communication both ways at the same time) or **Half Duplex** (communication in one direction at a time).

TX Driver Control

Used with a **EIA-485** serial interface, if your application supports **RTS** (Request To Send), select this option. Otherwise, select **Auto**. Default is **Auto**.

Echo Suppression

This parameter applies only to **EIA-485 Half Duplex** mode. All characters will be echoed to the user and transmitted across the serial ports. Some EIA-485 applications require local echo to be enabled in order to monitor the loopback data to determine that line contention has occurred. If your application cannot handle loopback data, echo suppression should be **On**. The default is echo suppression **Off**.

Monitor DSR

Specifies whether the RS-232 signal DSR (data set ready) should be monitored. This is used with modems or any device that sends a DSR signal. When it is monitored and the Device Server detects a DSR signal, the line service is started. Default is **Off**. If both **Monitor DCD** and **Monitor DSR** are enabled, both signals must be detected before the line service is started.

Monitor DCD

Specifies whether the RS-232 signal DCD (Data Carrier Detect) should be monitored. This is used with modems or any other device that sends a DCD signal. When it is monitored and the Device Server detects a DCD signal, the line service is started. Default is **Off**. If both **Monitor DCD** and **Monitor DSR** are enabled, both signals must be detected before the line service is started.

Line Termination

Used with **EIA-422** and **EIA-485** on SDS 8-port+ Device Server models, specifies whether or not the line is terminated; use this option when the line is connected to a device at the end of the EIA network.

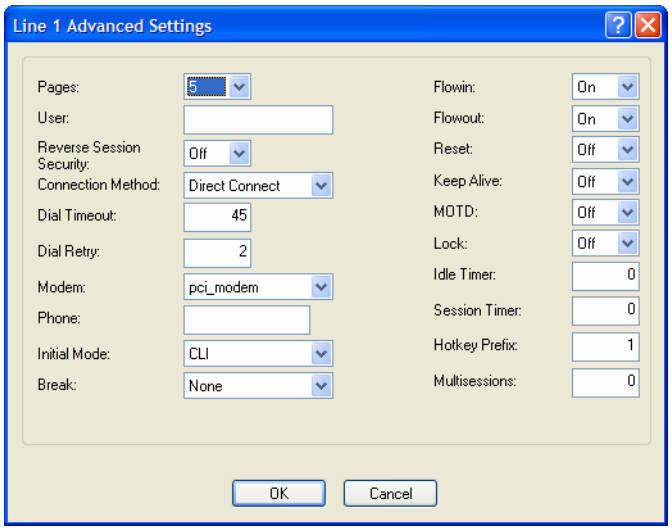
Email Alert

Determines whether or not email notification is turned on. Default is **Off**.

Click the **Settings** button to configure the email alert for the line. See ***Configuring Line Email Alerts*** on page 199 for parameter descriptions.

Advanced Line Settings

You can configure these advanced settings for a line.



Configure the appropriate parameters:

Pages

For **DSLogin** line service, this is the number of video pages the terminal supports. Valid values are 1-7. The default is **5** pages.

User

For **DSLogin** line service, makes this a line that is dedicated to the specified user. Only this user will be able to log in on this line and they won't need to enter their login name - just their password. When the **Line Service** is set to **Direct** or **Silent Rlogin**, the **User** parameter is used as the Rlogin user name (since Rlogin will not prompt you for a user name).

Reverse Session Security

Enables/disables login/password authentication, locally or externally, on reverse Telnet connections. The default is **Off**.

Connection Method Determines how a modem will work on the line. Select from the following options:

- **Direct Connect**—Indicates that there is not a modem on the line. This is the default.
- **Dial In**—Specify this option when a user is remote and will be dialing in via modem or ISDN TA.
- **Dial Out**—Specify this option when a modem is attached to the serial port and is being used to dial out.
- **Dial In/Out**—Specify this option when the Device Server is being used as a router (depending on which end of the link your Device Server is situated and how you want to initiate the communication).
- **MS Direct-Host**—Specify this option when the serial port is connected to a Microsoft Guest device. **Line Service** must be set to **PPP** for this option.
- **MS Direct-Guest**—Specify this option when the serial port is connected to a Microsoft Host device. **Line Service** must be set to **PPP** for this option.

Dial Timeout The number of seconds the Device Server will wait to establish a connection to a remote modem. The default value is **45** seconds.

Dial Retry The number of times the Device Server will attempt to re-establish a connection with a remote modem. The default value is **2**.

Modem The name of the predefined modem that is used on this line.

Phone The phone number to use when **Connection Method** is set to **Dial Out**.

Initial Mode Specifies the initial interface a user navigates when logging into the line; either the **Menu** or a prompt for the **CLI**. The default is **CLI**.

Break Specifies how a break is interpreted:

- **None**—The Device Server ignores the break key completely and it is not passed through to the host. This is the default setting.
- **Local**—The Device Server deals with the break locally. If the user is in a session, the break key has the same effect as a hot key.
- **Remote**—When the break key is pressed, the Device Server translates this into a telnet break signal which it sends to the host machine.
- **Brkintr**—On some systems such as SunOS, XENIX, and AIX, a break received from the peripheral is not passed to the client properly. If the client wishes to make the break act like an interrupt key (for example, when the stty options **`-ignbrk`** and **`brkintr`** are set).

Map CR to CRLF When **Line Service Printer** is selected, defines the default end-of-line terminator as CR-LF (ASCII carriage-return line-feed) when enabled. Default is **Off**.

Flowin Determines if input flow control is to be used. Default is **On**. This is active only when **Line Flow Control** is set to **Soft**, **Hard**, or **Both**.

Flowout Determines if output flow control is to be used. Default is **On**. This is active only when **Line Flow Control** is set to **Soft**, **Hard**, or **Both**.

Reset Resets the terminal type connected to the line when a user logs out.

Keep Alive

Enables a per-connection TCP keepalive feature; after approximately 3 minutes of network connection idle time, the connection will send a gratuitous ACK to the network peer, either ensuring the connection stays active OR causing a dropped connection condition to be recognised by the reverse raw service.

Applications using this feature need to be aware that there might be some considerable delay between a network disconnection and the port being available for the next connection attempt; this is to allow any data sent on prior connections to be transmitted out of the serial port buffer. Application network retry logic needs to accommodate this feature.

MOTD

Enables/disables the message of the day on the line.

Lock

When enabled, the user can lock his terminal with a password using the **Hotkey Prefix** (default Ctrl-a) **^a l** (lowercase L). The Device Server prompts the user for a password and a confirmation.

Idle Timer

Enter a time period, in seconds, for which the **Idle Timer** will run. Use this timer to close a connection because of inactivity. When the **Idle Timer** expires, the Device Server will end the connection. The maximum value is 4294967 seconds (about 49 days). The default value of **0** (zero) means the **Idle Timer** will not expire, so the connection is permanently open.

Session Timer

Enter a time, in seconds, for which the **Session Timer** will run. Use this timer to forcibly close the session (connection). When the **Session Timer** expires, the Device Server will end the connection. The default value is **0** seconds so the port will never timeout. The maximum value is 4294967 seconds (about 49 days).

Hotkey Prefix

The prefix that a user types to lock a line or redraw the Menu. The default value is **hex 01**, which corresponds to **Ctrl-a** (**^a**) (hex value 02 would be Ctrl-b (^b), etc.):

- **^a l**—(Lowercase L) Locks the line until the user unlocks it. The user is prompted for a password (any password, excluding spaces) and locks the line. Next, the user must retype the password to unlock the line.
- **^r**—When you switch from a session back to the Menu, the screen may not be redrawn correctly. If this happens, use this command to redraw it properly. This is always **Ctrl R**, regardless of the **Hotkey Prefix**.

You can use the **Hotkey Prefix** key to lock a line only when the **Line Lock** parameter is **On**.

Multisessions

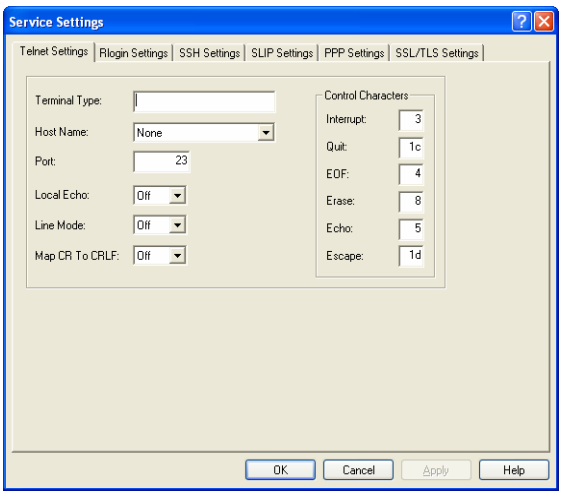
The number of extra reverse sessions available on a line (available only on 2 port+ models), in addition to the single session that is always available on the line. You can specify **0**-**7** multisessions per line. The default is **0** (zero). Total sessions available for the Device Server are 1-8 for the 2-/4-port models and 2x the number of ports for all other models.

Service Settings

Line Services determine how a line is defined. As a rule, when you are accessing a serial device through the Device Server, coming from the Ethernet side, you want to set the **Line Service** to **Reverse Raw**, **Reverse SSH**, or **Reverse Telnet**.

DSLogin

When the **Line Service** is set to **DSLogin**, any user accessing the Device Server will have to log into the Device Server. What happens after the user successfully logs into the Device Server is based on how the user is configured. For example, if after a successful login, the user is set to telnet to a specific host, you will want to set the **Telnet** parameters that will be used by the user for the telnet session (any parameters that are also available in the user's configuration are overridden by the user's definitions).

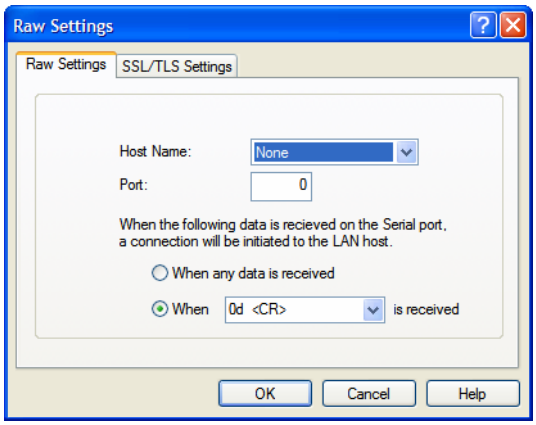


Configure the following parameters:

Telnet Settings	See ***Telnet Settings*** on page 172 for parameter definitions.
Rlogin Settings	See ***Rlogin Settings*** on page 173 for parameter definitions.
SSH Settings	See ***SSH Client Settings*** on page 182 for parameter definitions.
SLIP Settings	See ***SLIP Settings*** on page 174 for parameter definitions.
PPP Settings	See ***PPP Settings*** on page 176 for parameter definitions.
SSL Settings	See ***SSL/TLS Settings*** on page 154 for parameter definitions.

Direct Raw Settings

When the **Line Service** is set to **Direct Raw**, data is sent through the connection in its original format. This raw TCP/IP connection is initiated from the Device Server to the configured host.

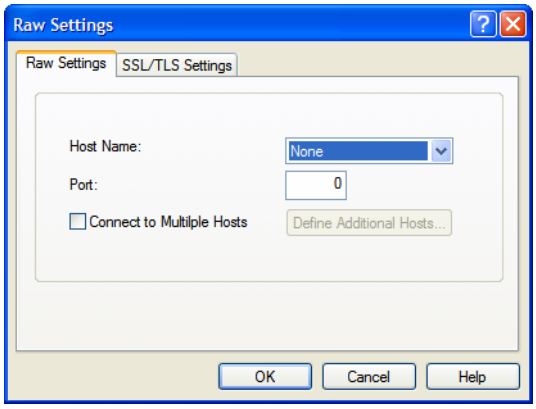


Configure the following parameters:

Host Name	The name of the target host.
Port	The port number the target host is listening on for incoming connections.
When any data is received	Initiates a Raw connection to the specified host when any data is received by the serial port.
When <data> is received	Initiates a Raw connection to the specified host only when the specified character is received by the serial port.

Silent Raw Settings

When the **Line Service** is set to **Silent Raw**, data is sent through the connection in its original format. This raw TCP/IP connection is initiated from the Device Server to the configured host.

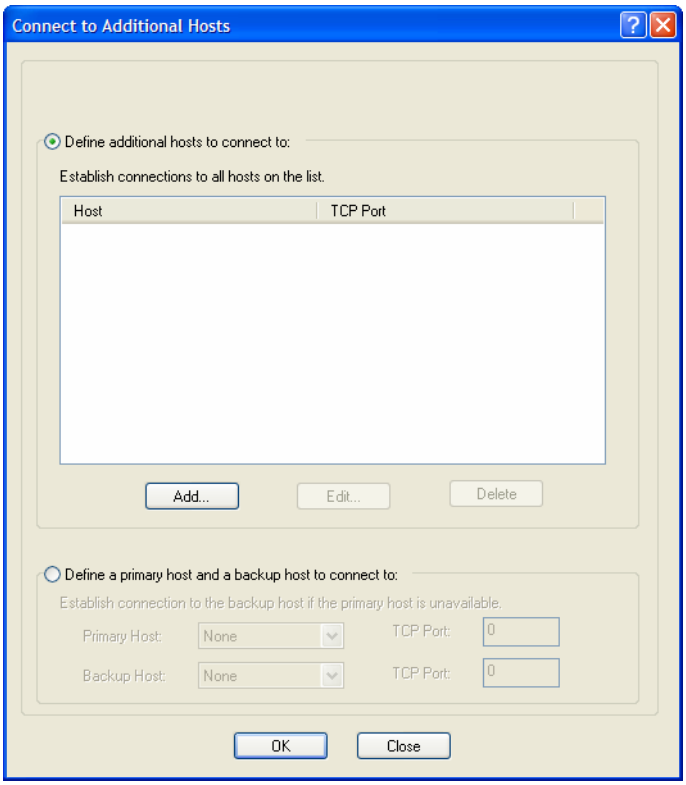


Configure the following parameters:

Host Name	The name of the target host.
Port	The port number the target host is listening on for incoming connections.
Connect to Multiple Hosts	When enabled, allows a serial device connected to this serial port to communicate to multiple hosts.
Define Additional Hosts Button	Click this button to define the hosts that this serial port will connect to.

Silent Raw Multihost

You can define a list of hosts that the serial device will communicate to or a primary/backup host.



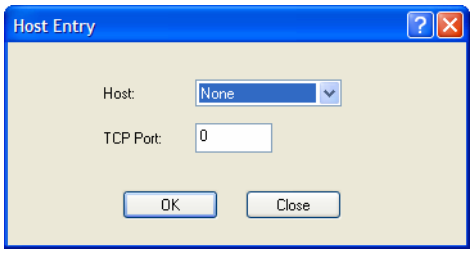
Configure the following parameters:

Define additional hosts to connect to	When this option is enabled, you can define up to 49 hosts that the serial device connected to this serial port will attempt communicate to.
Add Button	Click the **Add** button to add a host to the list of hosts that will be receiving communication from the serial device connected to the Device Server.
Edit Button	Highlight an existing host and click the **Edit** button to edit a host in the list of hosts that will be receiving communication from the serial device connected to the Device Server.
Delete Button	Click the **Delete** button to delete a host to the list of hosts that will be receiving communication from the serial device connected to the Device Server.
Define a primary host and backup...	When this option is enabled, you need to define a primary host that the serial device connected to this serial port will communicate to and a backup host, in the event that the Device Server looses communication to the primary host.
Primary Host	Specify a preconfigured host that the serial device will communicate to through the Device Server.
TCP Port	Specify the TCP port that the Device Server will use to communicate to the **Primary Host**.
Backup Host	Specify a preconfigured host that the serial device will communicate to through the Device Server if the Device Server cannot communicate with the **Primary Host.**

TCP Port Specify the TCP port that the Device Server will use to communicate to the **Backup Host**.

Adding/Editing a Multihost Entry

When you click the **Add** or **Edit** button, the Host Entry window appears. The hosts in the multihost list must already be defined (see ***Configuring Hosts*** on page 219) If you add a host that was defined with its fully qualified domain name (FQDN), it must be resolvable by your configured DNS server.



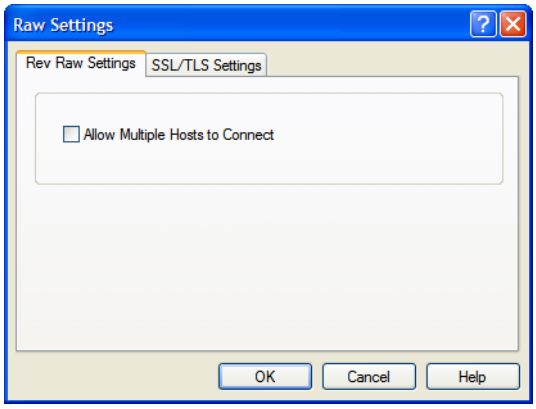
Configure the following parameters:

Host Name Enter the name of the target host.

TCP Port The port number the target host is listening on for incoming connections.

Reverse Raw Settings

When the **Line Service** is set to **Reverse Raw**, data is received through the connection in its original format. This raw TCP/IP connection is initiated from the Device Server to the configured host.

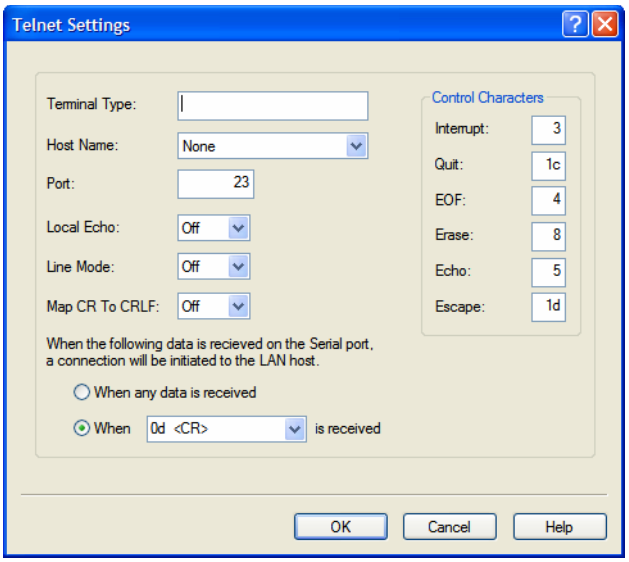


Configure the following option:

Allow Multiple Hosts to Connect When this option is enabled, multiple hosts can connect to a serial device that is connected to this serial port.

Telnet Settings

When the **Line Service** is set to **Direct** or **Silent Telnet** or **DSLogin**, data is sent through the connection in a telnet session. This telnet session is initiated from the Device Server to the configured host.



Configure the following parameters:

Terminal Type — Type of terminal attached to this line; for example, ANSI or WYSE60.

Host Name — The name of the target host.

Port — The port number the target host is listening on for incoming connections.

Local Echo — Toggles between local echo of entered characters and suppressing local echo. Local echo is used for normal processing, while suppressing the echo is convenient for entering text that should not be displayed on the screen, such as passwords. This parameter can only be used when **Line Mode** is **On**. Default is **Off**.

Line Mode — When **On**, keyboard input is not sent to the remote host until **Enter** is pressed, otherwise input is sent every time a key is pressed. Default is **Off**.

Map CR to CRLF — Maps carriage returns (CR) to carriage return line feed (CRLF). The default value is **Off**.

Interrupt — Defines the interrupt character. Typing the interrupt character interrupts the current process. This value is in hexadecimal with a default value of **3** (ASCII value **^C**).

Quit — Defines the quit character. Typing the quit character closes and exits the current telnet session. This value is in hexadecimal with a default value of **1c** (ASCII value **FS**).

EOF — Defines the end-of-file character. When **Line Mode** is **On**, entering the EOF character as the first character on a line sends the character to the remote host. This value is in hexadecimal with a default value of **4** (ASCII value **^D**).

Erase — Defines the erase character. When **Line Mode** is **Off**, typing the erase character erases one character. This value is in hexadecimal with a default value of **8** (ASCII value **^H**).

Echo Defines the echo character. When **Line Mode** is **On**, typing the echo character echoes the text locally and sends only completed lines to the host. This value is in hexadecimal with a default value of **5** (ASCII value **^E**).

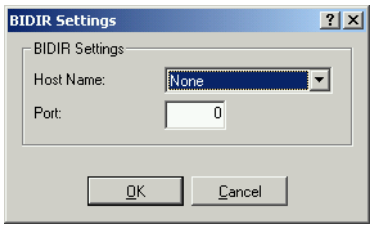
Escape Defines the escape character. Returns you to the command line mode. This value is in hexadecimal with a default value of **1d** (ASCII value **GS**).

When any data is received Initiates a Telnet connection to the specified host when any data is received by the serial port (direct Telnet only).

When <data> is received Initiates a Telnet connection to the specified host only when the specified character is received by the serial port (direct Telnet only).

BIDIR Settings

When the **Line Service** is set to **BIDIR**, a bidirectional connection is created, with data flowing in both directions in its original format. This raw TCP/IP connection can be initiated from either the Device Server or the configured host. The Device Server initiates TCP connections to the configured host and port and listens for TCP connections on the **DS Port** configured for the **Line**.



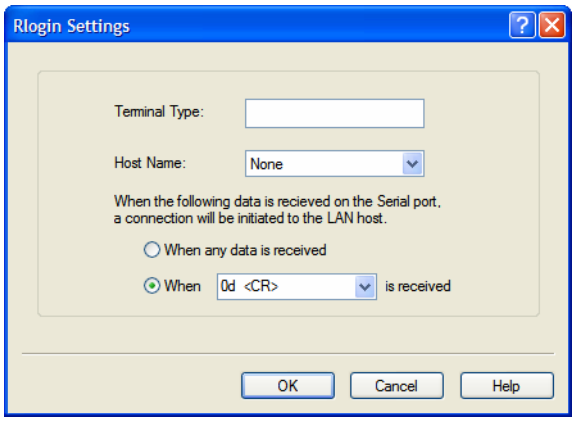
Configure the following parameters:

Host Name The name of the target host.

Port The port number the target host is listening on for incoming connections.

Rlogin Settings

When the **Line Service** is set to **Direct** or **Silent Rlogin** or **DSLogin**, data is sent in its original format. This rlogin session is initiated from the Device Server to the configured host.



Configure the following parameters:

Terminal Type Type of terminal attached to this line; for example, ansi or wyse60.

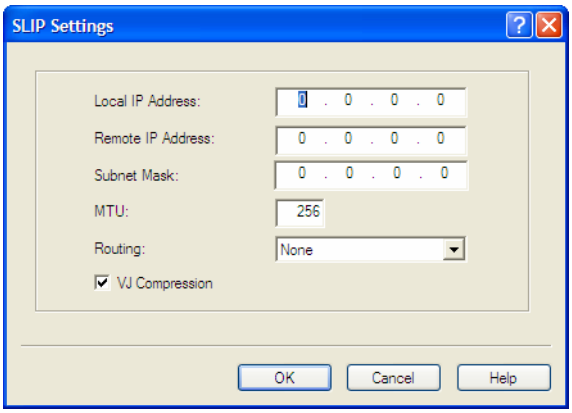
Host Name The name of the target host.

When any data is received Initiates a Rlogin connection to the specified host when any data is received by the serial port (direct Rlogin only).

When <data> is received Initiates a Rlogin connection to the specified host only when the specified character is received by the serial port (direct Rlogin only).

SLIP Settings

When the **Line Service** is set to **SLIP** or **DSLogin**, a SLIP connection is established between the Device Server and a remote user. This connection can be initiated by either the Device Server or the remote user.



Configure the following parameters:

Local IP Address The IPv4 address of the Device Server end of the SLIP link. For routing to work you must enter an IP address in this field. Choose an address that is part of the same network or subnetwork as the remote end; for example, if the remote end is address 192.101.34.146, your local IP address can be 192.101.34.145. Do not use the Device Server's (main) IP address in this field; if you do so, routing will not take place correctly.

Remote IP Address The IPv4 address of the remote end of the SLIP link. Choose an address that is part of the same network or subnetwork as the Device Server. If your user is authenticated by the Device Server, this remote IP address will be overridden if you have set a **Framed IP Address** for the user. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Address** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

Subnet Mask The network subnet mask. For example, 255.255.0.0. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Netmask** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

MTU The Maximum Transmission Unit (MTU) parameter restricts the size of individual SLIP packets being sent by the Device Server. Enter a value between 256 and 1006 bytes; for example, 512. The default value is **256**. If your user is authenticated by the Device Server, this MTU value will be overridden when you have set a **Framed MTU** value for the user. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-MTU** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

Routing

Determines the routing mode (RIP, Routing Information Protocol) used on the **SLIP** interface as one of the following options:

- **None**—Disables RIP over the SLIP interface.
- **Send**—Sends RIP over the SLIP interface.
- **Listen**—Listens for RIP over the SLIP interface.
- **Send and Listen**—Sends RIP and listens for RIP over the SLIP interface.

This is the same function as the **Framed-Routing** attribute for RADIUS authenticated users. Default is **None**.

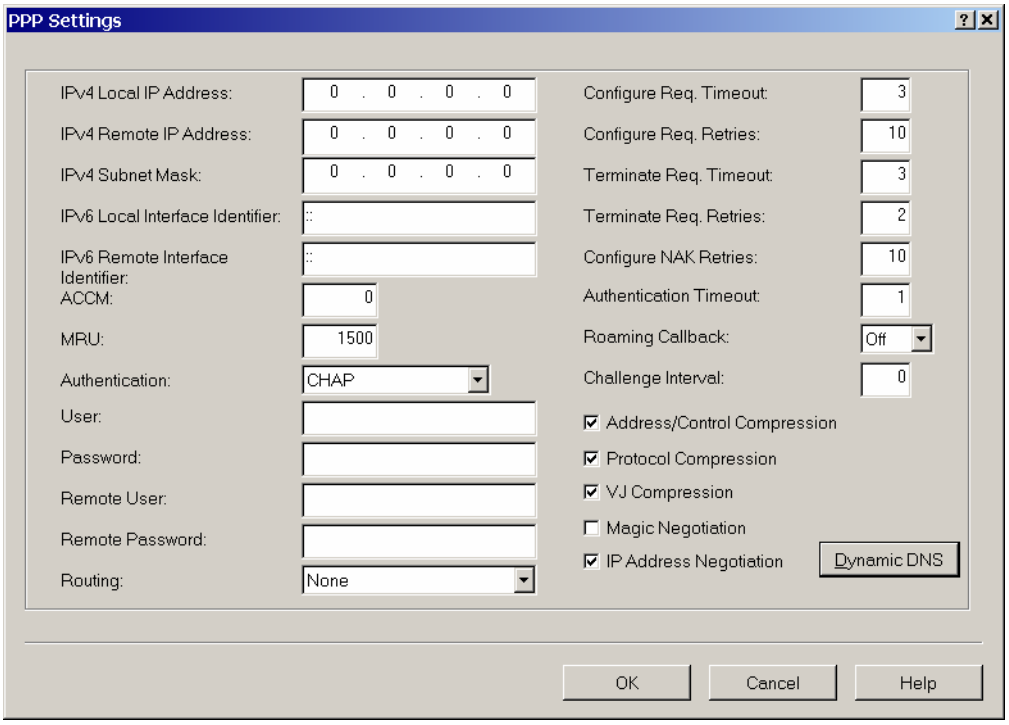
VJ Compression

This determines whether Van Jacobson compression is used on this link; that is, whether you are using SLIP or C-SLIP (compressed SLIP). The choices are **On** (C-SLIP) or **Off** (SLIP). The default is **On**. C-SLIP greatly improves the performance of interactive traffic, such as Telnet or Rlogin.

If your user is authenticated by the Device Server, this VJ compression value will be overridden if you have set a **Framed Compression** value for a user. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Compression** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

PPP Settings

When the **Line Service** is set to **PPP** or **DSLogin**, a PPP connection is established between the Device Server and a remote user. This connection can be initiated by either the Device Server or the remote user.



Configure the following parameters:

IPv4 Local IP Address

The IPV4 IP address of the Device Server end of the PPP link. For routing to work, you must enter a local IP address. Choose an address that is part of the same network or subnetwork as the remote end; for example, if the remote end is address 192.101.34.146, your local IP address can be 192.101.34.145. Do not use the Device Server's (main) IP address in this field; if you do so, routing will not take place correctly.

IPv4 Remote IP Address

The IPV4 IP address of the remote end of the PPP link. Choose an address that is part of the same network or subnetwork as the Device Server. If you set the PPP parameter IP Address Negotiation to On, the Device Server will ignore the remote IP address value you enter here and will allow the remote end to specify its IP address. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Address** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here. The exception to this rule is a **Framed-Address** value in the RADIUS file of **255.255.255.254**; this value allows the Device Server to use the remote IP address value configured here.

IPv4 Subnet Mask

The network subnet mask. For example, 255.255.0.0. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Netmask** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

IPv6 Local Interface Identifier

The local IPv6 interface identifier of the Device Server end of the PPP link. For routing to work, you must enter a local IP address. Choose an address that is part of the same network or subnetwork as the remote end. Do not use the Device Server's (main) IP address in this field; if you do so, routing will not take place correctly. The first 64 bits of the Interface Identifier must be zero, therefore, ::abcd:abcd:abcd:abcd is the expected format.

IPv6 Remote Interface Identifier

The remote IPv6 interface identifier of the remote end of the PPP link. Choose an address that is part of the same network or subnetwork as the Device Server. If you set the **PPP** parameter **IP Address Negotiation** to **On**, the Device Server will ignore the remote IP address value you enter here and will allow the remote end to specify its IP address. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Interface-ID** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here. The first 64 bits of the Interface Identifier must be zero, therefore, ::abcd:abcd:abcd:abcd is the expected format.

ACCM

Specifies the ACCM (Asynchronous Control Character Map) characters that should be escaped from the data stream. This is entered as a 32-bit hexadecimal number with each bit specifying whether or not the corresponding character should be escaped. The bits are specified as the most significant bit first and are numbered 31-0. Thus if bit 17 is set, the 17th character should be escaped, that is, 0x11 (XON). So entering the value 000a0000 will cause the control characters 0x11 (XON) and 0x13 (XOFF) to be escaped on the link, thus allowing the use of XON/XOFF (software) flow control. If you have selected **Soft Flow Control** on the **Line**, you must enter a value of at least **000a0000** for the **ACCM**. The default value is **00000000**, which means no characters will be escaped.

MRU

The Maximum Receive Unit (MRU) parameter specifies the maximum size of PPP packets that the Device Server's port will accept. Enter a value between 64 and 1500 bytes; for example, 512. The default value is **1500**. If your user is authenticated by the Device Server, the **MRU** value will be overridden if you have set a **Framed MTU** value for the user. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-MTU** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

Authentication

The type of authentication that will be done on the link: **None**, **PAP**, or **CHAP**. The default is **CHAP**. You can use **PAP** or **CHAP** to authenticate a port or user on the Device Server, from a remote location, or authenticate a remote client/device, from the Device Server (not commonly used for **Dial Out**).

PAP is a one time challenge of a client/device requiring that it respond with a valid username and password. A timer operates during which successful authentication must take place. If the timer expires before the remote end has been authenticated successfully, the link will be terminated.

CHAP challenges a client/device at regular intervals to validate itself with a username and a response, based on a hash of the secret (password). A timer operates during which successful authentication must take place. If the timer expires before the remote end has been authenticated successfully, the link will be terminated.

When setting either **PAP** and **CHAP**, make sure the Device Server and the remote client/device have the same setting. For example, if the Device Server is set to **PAP**, but the remote end is set to **CHAP**, the connection will be refused.

User

Complete this field only if you have specified **PAP** or **CHAP** (security protocols) in the **Security** field, *and*

- you wish to dedicate this line to a single remote user, who will be authenticated by the Device Server, *or*
- you are using the Device Server as a router (back-to-back with another Device Server).

When **Connection Method** is set to **Out** or **Both**, the **User** is the name the remote device will use to authenticate a port on this Device Server. The remote device will only authenticate your Device Server's port when **PAP** or **CHAP** are operating. You can enter a maximum of sixteen alphanumeric characters; for example, tracy201. When connecting together two networks, enter a dummy user name; for example, DS_HQ.

Note If you want a reasonable level of security, the user name and password should not be similar to a user name or password used regularly to login to the Device Server. External authentication can not be used for this user.

Password

Complete this field only if you have specified **PAP** or **CHAP** (security protocols) in the **Security** field and:

- you wish to dedicate this line to a single remote user, who will be authenticated by the Device Server, *or*
- you are using the Device Server as a router (back-to-back with another Device Server)

Password means the following:

- When **PAP** is specified, this is the password the remote device will use to authenticate the port on this Device Server.
- When **CHAP** is specified, this is the secret (password) known to both ends of the link upon which responses to challenges shall be based.

In either case, you can enter a maximum of 16 alphanumeric characters.

Remote User

Complete this field only if you have specified **PAP** or **CHAP** (security protocols) in the **Security** field, *and*

- you wish to dedicate this line to a single remote user, who will be authenticated by the Device Server, *or*
- you are using the Device Server as a router (back-to-back with another Device Server)

When **Connection Method** is set to **In** or **Both**, the **Remote User** is the name the Device Server will use to authenticate the port on the remote device. Your Device Server will only authenticate the port on the remote device when **PAP** or **CHAP** are operating. You can enter a maximum of sixteen alphanumeric characters. When connecting together two networks, enter a dummy user name; for example, DS_SALES.

Note If you want a reasonable level of security, the user name and password should not be similar to a user name or password used regularly to login to the Device Server. This option does not work with external authentication.

Remote Password

Complete this field only if you have specified **PAP** or **CHAP** (security protocols) in the **Security** field, *and*

- you wish to dedicate this line to a single remote user, and this user will be authenticated by the Device Server, *or*
- you are using the Device Server as a router (back-to-back with another Device Server)

Remote password means the following:

- When **PAP** is specified, this is the password the Device Server will use to authenticate the remote device.
- When **CHAP** is specified, this is the secret (password) known to both ends of the link upon which responses to challenges will be based.

Remote Password is the opposite of the parameter **Password**. Your Device Server will only authenticate the remote device when **PAP** or **CHAP** is operating. In either case, you can enter a maximum of sixteen alphanumeric characters.

Routing

Determines the routing mode (RIP, Routing Information Protocol) used on the **PPP** interface as one of the following options:

- **None**—Disables RIP over the PPP interface.
- **Send**—Sends RIP over the PPP interface.
- **Listen**—Listens for RIP over the PPP interface.
- **Send and Listen**—Sends RIP and listens for RIP over the PPP interface.

This is the same function as the **Framed-Routing** attribute for RADIUS authenticated users. Default is **None**.

Configure Req. Timeout

The maximum time, in seconds, that LCP (Link Control Protocol) will wait before it considers a **`configure request`** packet to have been lost.

Configure Req. Retries

The maximum number of times a **`configure request`** packet will be re-sent before the link is terminated.

Terminate Req. Timeout

The maximum time, in seconds, that LCP (Link Control Protocol) will wait before it considers a **`terminate request`** packet to have been lost.

Terminate Req. Retries

The maximum number of times a **`terminate request`** packet will be re-sent before the link is terminated.

Configure NAK Retries

The maximum number of times a **`configure NAK`** packet will be re-sent before the link is terminated.

Authentication Timeout

The timeout, in minutes, during which successful PAP or CHAP authentication must take place (when **PAP** or **CHAP** is turned **On**). If the timer expires before the remote end has been authenticated successfully, the link will be terminated.

Roaming Callback

A user can enter a telephone number that the Device Server will use to callback him/her. This feature is particularly useful for a mobile user. Roaming callback can only work when the **User Callback** parameter is set to **On**. Roaming callback therefore overrides (fixed) **User Callback**.To use **Roaming Callback**, the remote end must be a Microsoft Windows OS that supports Microsoft's Callback Control Protocol (CBCP). The user is allowed 30 seconds to enter a telephone number after which the Device Server ends the call. The default is **Off**.

Challenge Interval

The interval, in minutes, for which the Device Server will issue a CHAP re-challenge to the remote end. During CHAP authentication, an initial CHAP challenge takes place, and is unrelated to CHAP re-challenges. The initial challenge takes place even if re-challenges are disabled. Some PPP client software does *not* work with CHAP re-challenges, so you might want to leave the parameter disabled in the Device Server. The default value is **0** (zero), meaning CHAP re-challenge is disabled.

Address/Control Compression

This determines whether compression of the **PPP Address** and **Control** fields take place on the link. The default is **On**. For most applications this should be enabled.

Protocol Compression

This determines whether compression of the PPP Protocol field takes place on this link. The default is **On**.

VJ Compression

This determines whether Van Jacobson Compression is used on this link. The default is **On**. If your user is authenticated by the Device Server, this VJ compression value will be overridden if you have set the **User Framed Compression On**. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Compression** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

Magic Negotiation

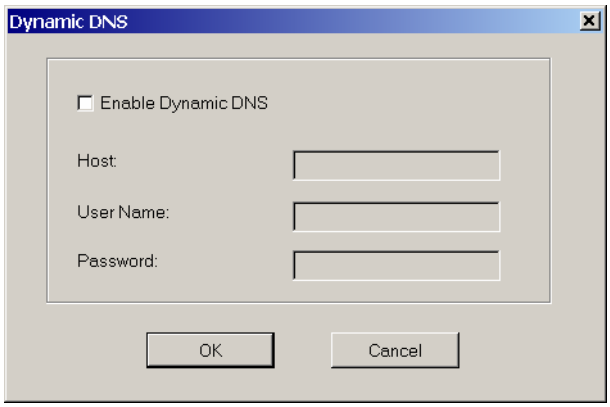
Determines if a line is looping back. If enabled (**On**), random numbers are sent on the link. The random numbers should be different, unless the link loops back. The default is **Off**.

IP Address Negotiation

Specifies whether or not IP address negotiation will take place. IP address negotiation is where the Device Server allows the remote end to specify its IP address. The default value is **Off**. When **On**, the IP address specified by the remote end will be used in preference to the **Remote IP Address** set for a **Line**. When **Off**, the **Remote IP Address** set for the **Line** will be used.

PPP Dynamic DNS Settings

The PPP Dynamic DNS settings are to be used in conjunction with a wireless WAN card and DynDNS.org account.

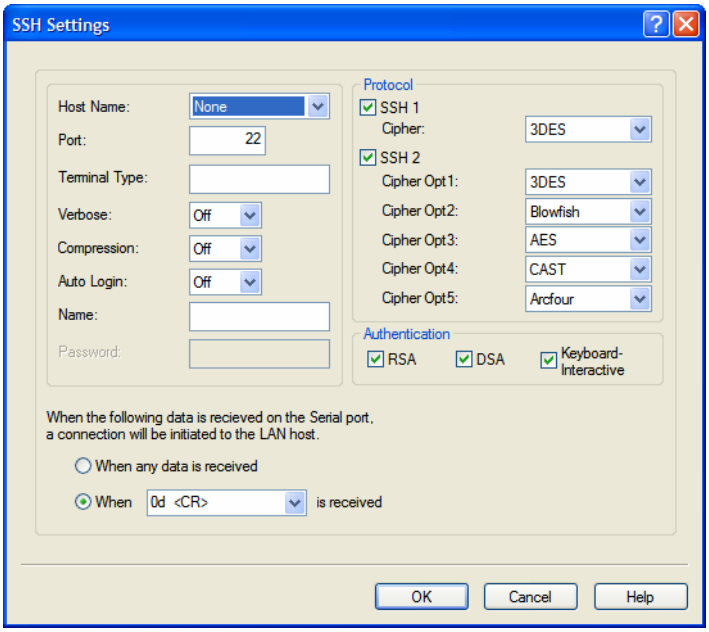


Configure the following parameters:

Enable Dynamic DNS	Enables/disables the ability to register a new IP address with the DynDNS.org server.
Host	Specify the host name that will be updated with the PPP session's IP address on the DynDNS.org server.
User Name	Specify the user name used to access the DynDNS.org server.
Password	Specify the password used to access the DynDNS.org server.

SSH Client Settings

When the **Line Service** is set to **Direct** or **Silent SSH** or **DSLogin**, the data will be sent through the connection in an SSH session. This session will be initiated by the Device Server to the configured host.



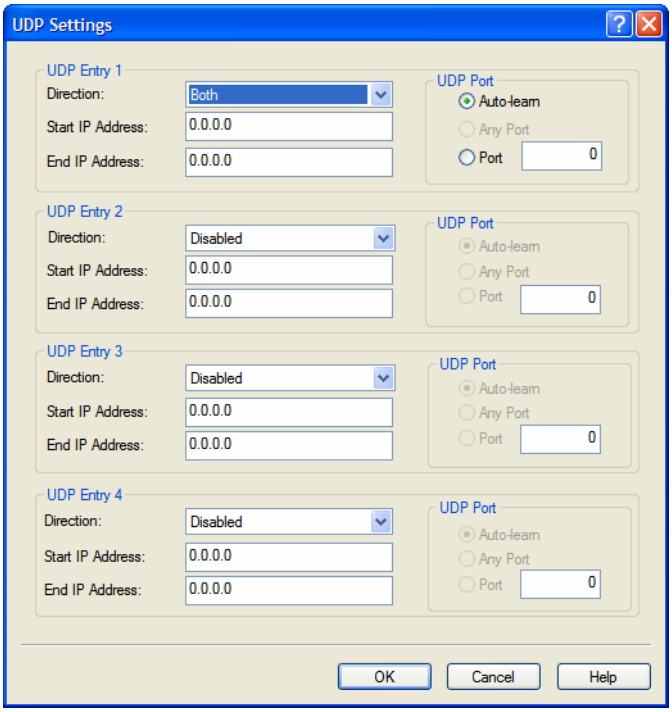
Configure the following parameters:

Host Name	The name of the target host.
Port	The port number the target host is listening on for incoming connections. The default is port 22.
Terminal Type	The type of terminal that will connecting via SSH.
Verbose	Displays debug messages on the terminal.
Compression	Requests compression of all data. Compression is desirable on modem lines and other slow connections, but will only slow down things on fast networks.
Auto Login	Creates an automatic SSH login, using the **Name** and **Password** values.
Name	The name of the user logging into the SSH session.
Password	The user's password when **Auto Login** is enabled.
SSH 1	Selects an SSH version 1 connection.
SSH 1 Cipher	Select the encryption method (cipher) that you want to use for your SSH version 1 connection: • **3DES** • **Blowfish**
SSH2	Selects an SSH version 2 connection. If both SSH 1 and SSH 2 are selected, the Device Server will attempt to make an SSH 2 connection first. If that connection fails, it will attempt to connect to the specified host using SSH 1.

SSH 2 Ciphers	Select the order of negotiation for the encryption method (ciphers) that the Device Server will use for the SSH version 2 connection: • **3DES** • **Blowfish** • **AES** • **Arcfour** • **CAST**
RSA	An authentication method used by SSH version 1 and 2. Use RSA authentication for the SSH session.
DSA	An authentication method used by SSH version 2. Use DSA authentication for the SSH session.
Keyboard Interactive	The user types in a password for authentication.
When any data is received	Initiates a SSH connection to the specified host when any data is received by the serial port (direct SSH only).
When <data> is received	Initiates a SSH connection to the specified host only when the specified character is received by the serial port (direct SSH only).

UDP Settings

When the **Line Service** is set to **UDP**, the Device Server processes UDP packets according to the UDP settings.



Configure the following parameters:

Direction

The direction in which information is received or relayed:

- **Disabled**—UDP service not enabled.
- **LAN to Serial—UDP Port** can be set to **Auto-learn** or **Port**. The Device Server will listen on port value configured in the **DS Port** parameter for messages coming from the learned or configured port.
- **Serial to LAN—UDP Port** can be set to **Port** only. The Device Server will listen on the port value configured in the **DS Port** parameter and will send to the configured port.
- **Both**—Messages are relayed both directions. UDP Port can be set to **Auto-learn** or **Port**. For messages coming from the LAN to the serial device, Device Server will listen on port value configured in the **DS Port** parameter for messages coming from the learned or configured port. For messages going from the serial device to the LAN, the Device Server will listen on the port value configured in the **DS Port** parameter and will send to the configured or learned (if **Auto-learn** is enabled, the Device Server must receive a UDP message before it can send one, since the port must first be 'learned') port.

Start IP Address

The first host IP address in the range of IP addresses (for IPV4 or IPV6) that the Device Server will listen for messages from and/or send messages to.

End IP Address

The last host IP address in the range of IP addresses (for IPV4, not required for IPV6) that the Device Server will listen for messages from and/or send messages to.

Auto-learn

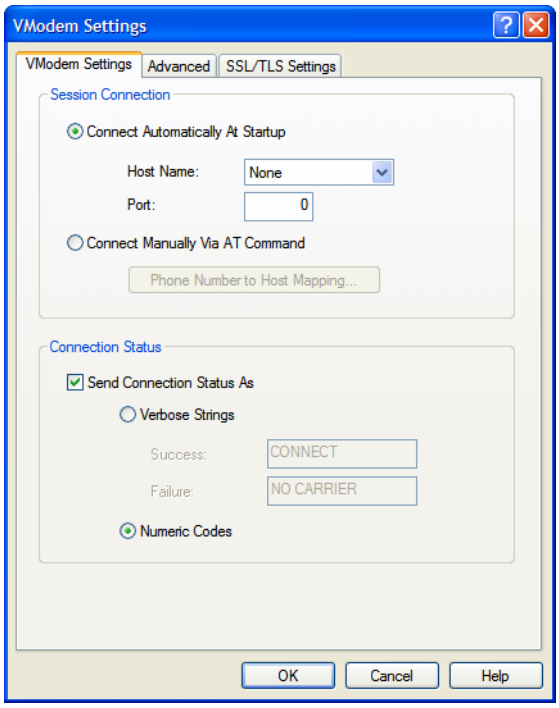
The Device Server will only listen to the first port that it receives a UDP packet from. Applicable when **Direction** is set to **LAN to Serial** or **Both**.

Any Port The Device Server will receive messages from any port sending UDP packets. Applicable when **Direction** is set to **LAN to Serial**.

Port The port that the Device Server will use to relay messages to servers/hosts. This option works with any **Direction** except **Disabled**. The Device Server will listen for UDP packets on the port configured by the **DS Port** parameter.

VModem Settings

When the **Line Service** is set to **VModem**, the Device Server acts as a virtual modem. After a virtual modem connection is established, data will flow in both directions in its original format.



Configure the following parameters:

Connect Automatically at Startup When enabled, automatically establishes the vmodem connection when the serial port becomes active.

Host Name The target host name.

Port The port number the target host is listening on for messages.

Connect Manually Via AT Command When enabled, vmodem requires an AT command before it establishes a connection. Specify this option when your modem application sends a phone number or other AT command to a modem.

Phone Number Mapping Button When your modem application sends a phone number or AT command string, you can map that phone number or AT command string to the receiving Device Server vmodem port.

Send Connection Status As When enabled, the connection success/failure indication strings are sent to the connected device, otherwise these indications are suppressed. The default is disabled.

Verbose Strings The connection status is sent by return codes (strings) to the connected device.

Success String that is sent to the serial device when a connection succeeds. If no string is entered, then the string **`CONNECT`** will be sent with the connecting speed, for example **`CONNECT 9600`**.

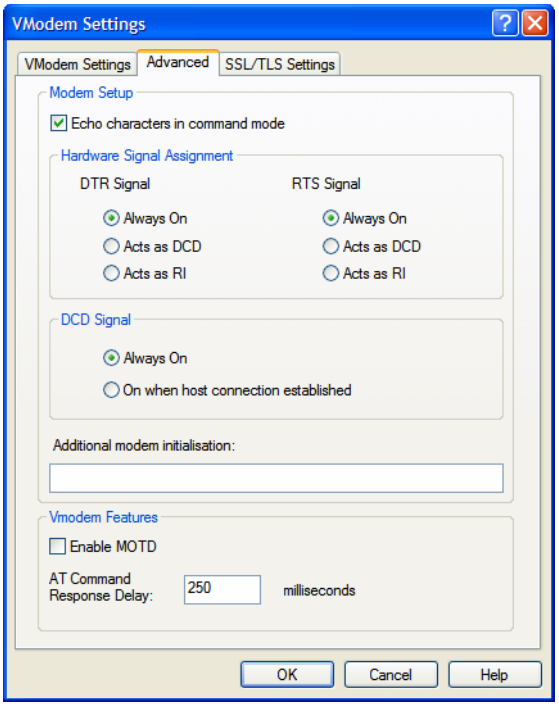
Failure String that is sent to the serial device when a connection fails. If no string is entered, then the string **`NO CARRIER`** will be sent.

Numeric The connection status is sent to the connected device using the following numeric codes:

- **1** Successfully Connected
- **2** Failed to Connect
- **4** Error

VModem Advanced Settings

When the **Line Service** is set to **VModem**, the Device Server acts as a virtual modem. After a virtual modem connection is established, data will flow in both directions in its original format.



Configure the following parameters:

Echo characters in command mode When enabled, echoes back characters that are typed in (equivalent to ATE0/ATE1 commands). Disabled by default.

DTR Signal Always On When you configure the DTR or RTS signal pin to act as a DCD signal pin, enable this option to make the DCD signal pin to always stay on. This is the DCD signal pin default.

DTR Signal Acts as DCD Specify this option to make the DTR signal always act as a DCD signal.

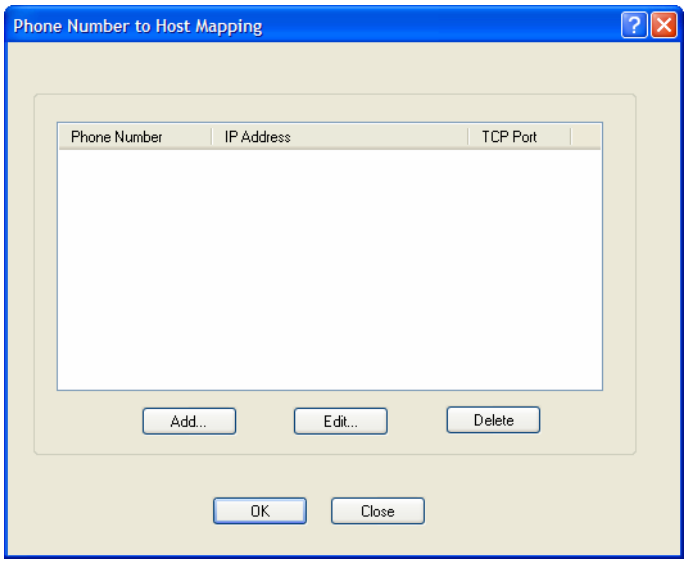
DTR Signal Acts as RI Specify this option to make the DTR signal always act as a RI signal.

RTS Signal Always On Specify this option to make the RTS signal always act as a RTS signal. This is the default.

RTS Signal Acts as DCD	Specify this option to make the RTS signal always act as a DCD signal.
RTS Signal Acts as RI	Specify this option to make the RTS signal always act as a RI signal.
Additional modem initialisation	You can specify additional vmodem commands that will affect how vmodem starts. The following commands are supported: ATQn, ATVn, ATEn, +++ATH, ATA, ATI0, ATI3, ATS0, AT&Z1, AT&Sn, AT&Rn, AT&Cn, AT&F, ATS2, ATS12, ATO (ATD with no phone number), and ATDS1. See ***VModem Initialisation Commands*** on page 87 for a more detailed explanation of the support initialisation commands.
Enable MOTD	When enabled, displays the Message of the Day (MOTD) when a successful vmodem connection is made. Disabled by default.
AT Command Response Delay	The amount of time, in milliseconds, before an AT response is sent to the requesting device. The default is 250 ms.

VModem Phone Number to Host Mapping

If your modem application requires a phone number or AT command, you can add an entry in the Phone Number to Host Mapping window that can be accessed by all VModem configured serial ports. You need to enter the phone number or AT command required by your modem application and the Device Server IP address and TCP Port that will be receiving the 'call.' 1-port models support up to 4 entries, all other desktop models support up to 8 entries, and rack-mount models support up to 48 entries.

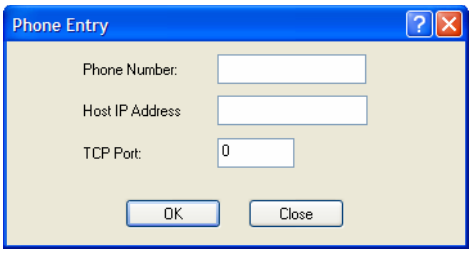


The following buttons are available:

Add Button	Click the **Add** button to display a window that allows you to configure the phone number or AT command your modem application sends and the Device Server's IP address and TCP port number that is receiving the call.
Edit Button	Click on a phone number entry and click the **Edit** button to change any values configured for the phone number.
Delete Button	Click on a phone number entry and click the **Delete** button to remove it from the phone number list.

VModem Phone Number Entry

Create an entry in the Phone Number to Host Mapping window.



Configure the following parameters:

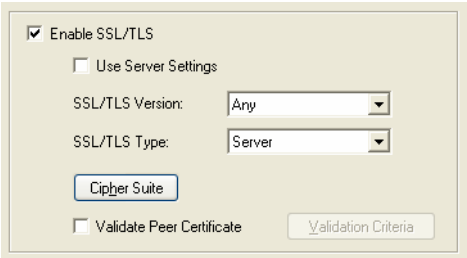
Phone Number Specify the phone number or AT command that your modem application sends to the modem.

Host IP Address Specify the IP address of the Device Server that is receiving the vmodem connection.

TCP Port Specify the TCP Port on the Device Server that is set to receive the vmodem connection.

SSL/TLS Settings

SSL/TLS can be configured for any service that uses a raw connection, such as Dir/Sil/Rev Raw, Vmodem, and Bidir. SSL/TLS can also be configured for DSLogin, which is used when the **User Service SSL-Raw** is configured. The **Server Tunnel Line Service** requires no configuration unless you want to send the data encrypted using SSL/TLS; for more information about the **Server Tunnel Line Service**, see ***Serial Tunnel Settings*** on page 93.



Configure the following parameters:

Enable SSL/TLS Activates the SSL/TLS settings for the line.

User Server Settings Uses the SSL/TLS server configuration for the line.

SSL/TLS Version Specify whether you want to use:

- **Any**—The Device Server will try a TLSv1 connection first. If that fails, it will try an SSLv3 connection. If that fails, it will try an SSLv2 connection.
- **TLSv1**—The connection will use only TLSv1.
- **SSLv3**—The connection will use only SSLv3.

The default is **Any**.

SSL/TLS Type Specify whether the Device Server will act as an SSL/TLS client or server. The default is **Client**.

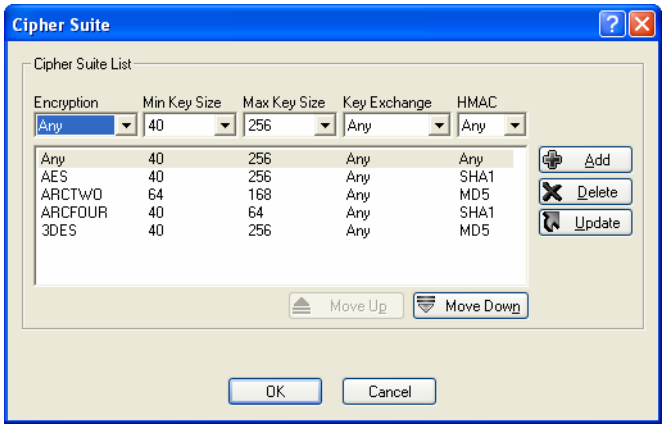
Validate Peer Certificate

Enable this option when you want the Validation Criteria to match the Peer Certificate for authentication to pass. If you enable this option, you need to download an SSL/TLS certificate authority (CA) list file to the Device Server.

For more information, see ***Keys and Certificates*** on page 98.

Cipher Suite

You can set up cipher rules to govern the encryption that will be used for the SSL/TLS connection.



Configure the following parameters:

Encryption

Select the type of encryption that will be used for the SSL connection:

- Any—Will use the first encryption format that can be negotiated.
- AES
- 3DES
- DES
- ARCFOUR
- ARCTWO

The default value is **Any**.

Min Key Size

The minimum key size value that will be used for the specified encryption type. The default is **40**.

Max Key Size

The maximum key size value that will be used for the specified encryption type. The default is **256**.

Key Exchange

The type of key to exchange for the encryption format:

- **Any**—Any key exchange that is valid is used (this does not, however, include ADH keys).
- **RSA**—This is an RSA key exchange using an RSA key and certificate.
- **EDH-RSA**—This is an EDH key exchange using an RSA key and certificate.
- **EDH-DSS**—This is an EDH key exchange using a DSA key and certificate.
- **ADH**—This is an anonymous key exchange which does not require a private key or certificate. Choose this key if you do not want to authenticate the peer device, but you want the data encrypted on the SSL/TLS connection.

The default is **Any**.

HMAC

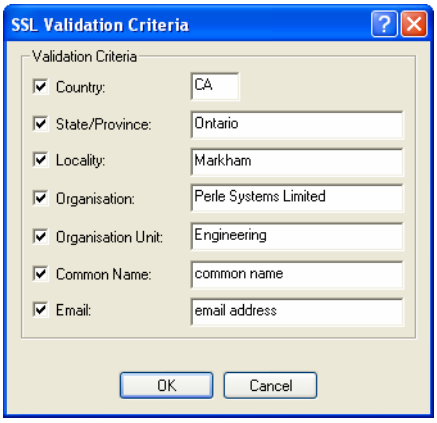
Select the key-hashing for message authentication method for your encryption type:

- Any
- MD5
- SHA1

The default is **Any**.

Validation Criteria

If you choose to configure validation criteria, then the information in the peer SSL/TLS certificate must match exactly the information configured in this window in order to pass peer authentication and create a valid SSL/TLS connection.



Configure the following parameters:

Country

A two character country code; for example, US. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

State/Province

Up to a 128 character entry for the state/province; for example, IL. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Locality

Up to a 128 character entry for the location; for example, a city. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Organisation

Up to a 64 character entry for the organisation; for example, Accounting. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Oraganisation Unit

Up to a 64 character entry for the unit in the organisation; for example, Payroll. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Common Name

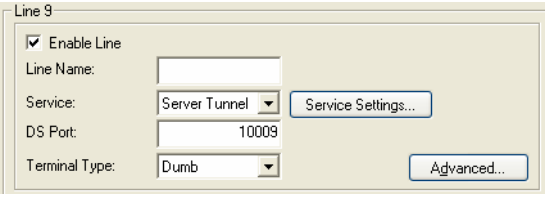
Up to a 64 character entry for common name; for example, the host name or fully qualified domain name. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Email

Up to a 64 character entry for an email address; for example, acct@anycompany.com. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Server Tunnel Settings

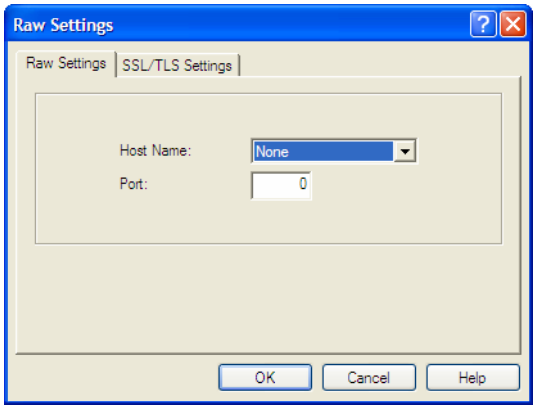
The purpose of the serial **Line Service Client/Server Tunnel** is to allow two Device Servers that are connected back-to-back over Ethernet to virtually link two serial ports. The serial device that initiates the connection is the **Client Tunnel** and the recipient is the **Server Tunnel**, although once the communication tunnel has been successfully established, the communication tunnel will stay connected and can go both ways. The Server Tunnel will support Telnet Com Port Control protocol as detailed in RFC 2217. See ***Serial Tunnel Settings*** on page 93 for more information about how to configure the Device Server for a serial tunnelling.



It is important that the **Client Tunnel Port** parameter reflect the **DS Port** set for the Line when the Device Servers are being used back-to-back over Ethernet.

Client Tunnel Settings

The purpose of the serial **Line Service Client/Server Tunnel** is to allow two Device Servers that are connected back-to-back over Ethernet to virtually link two serial ports. The serial device that initiates the connection is the **Client Tunnel** and the recipient is the **Server Tunnel**, although once the communication tunnel has been successfully established, the communication tunnel will stay connected and can go both ways. See ***Serial Tunnel Settings*** on page 93 for more information about how to configure the Device Server for a serial tunnelling.

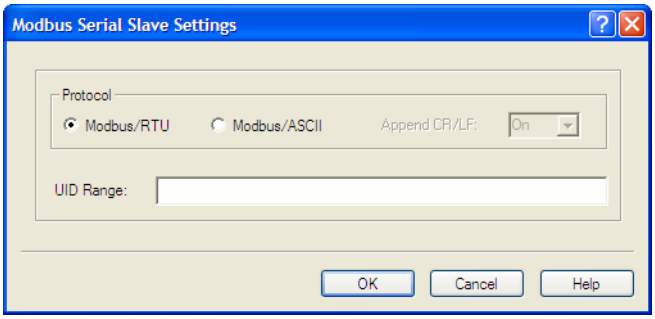


Configure the following parameters:

Host Name	The name of the Device Server that is connected to the serial device, acting as the Server Tunnel.
Port	The **DS Port** of the Device Server that is connected to the serial device.

Modbus Slave Settings

This window configures the parameters for Modbus Slaves residing on the serial side of the Device Server. See ***Modbus Configuration*** on page 79 for more information about how to configure the Device Server for a Modbus environment.



Configure the following parameters:

Modbus/RTU Select this option if the Modbus Master is configured using the Modbus/RTU protocol.

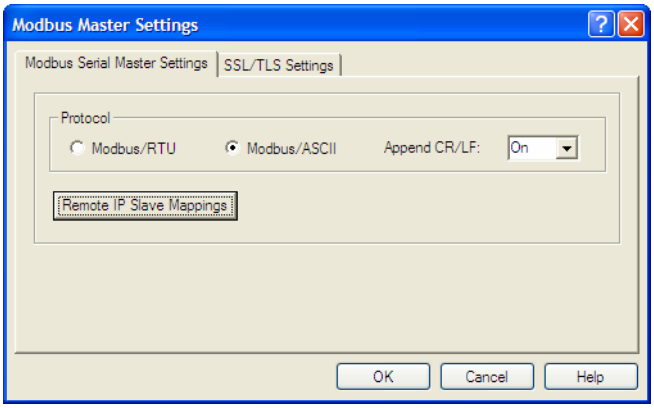
Modbus/ASCII Select this option if the Modbus Master is configured using the Modbus/ASCII protocol.

Append CR/LF When **Modbus/ASCII** is selected, adds a CR/LF to the end of the transmission; most Modbus devices require this option. The default is **On**.

UID Range You can specify a range of UIDs (1-247), in addition to individual UIDs. The format is comma delimited; for example, 2-35, 50, 100-103.

Modbus Master Settings

This window configures the parameters for Modbus Masters on the serial side of the Device Server. You can also choose to transmit the Modbus Master data encrypted via SSL/TLS. See ***Modbus Configuration*** on page 79 for more information about how to configure the Device Server for a Modbus environment.



Configure the following parameters:

Modbus/RTU Select this option if the Modbus Slave is configured using the Modbus/RTU protocol.

Modbus/ASCII Select this option if the Modbus Slave is configured using the Modbus/ASCII protocol.

Append CR/LF

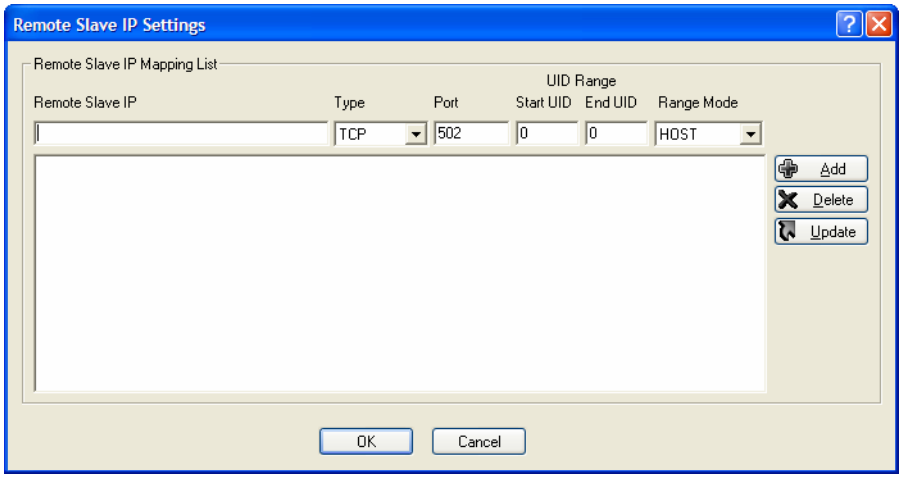
When **Modbus/ASCII** is selected, adds a CR/LF to the end of the transmission; most Modbus devices require this option. The default is **On**.

Remote IP Slave Mappings Button

Click this button to launch the Remote Slave IP Settings window, where you can configure the TCP/Ethernet Modbus Slaves that the Modbus Master on the Line will communicate with.

Remote IP Slave Mappings

This window allows you to configure all the Modbus Slaves, which reside on the Ethernet/TCP side of the Device Server, that will be receiving messages from the Modbus Master. See ***Modbus Configuration*** on page 79 for more information about how to configure the Device Server for a Modbus environment.



Configure the following parameters:

Remote Slave IP

The IP address of the TCP/Ethernet Modbus Slave.

Protocol

Specify the protocol that is used between the Modbus Master and Modbus Slave(s), either TCP or UDP.

Port

The destination port of the remote Modbus TCP Slave that the Device Server will connect to.

Start UID

When **Range Mode** is **Host** and you have sequential Modbus Slave IP addresses (for example, 10.10.10.1, 10.10.10.2, 10.10.10.3, etc.), you can specify a UID range and the Device Server will automatically increment the last digit of the configured IP address. Therefore, you can specify a UID range of 1-100, and the Device Server will route Master Modbus messages to all Modbus Slaves with IP addresses of 10.10.10.1 - 10.10.10.100.

End UID

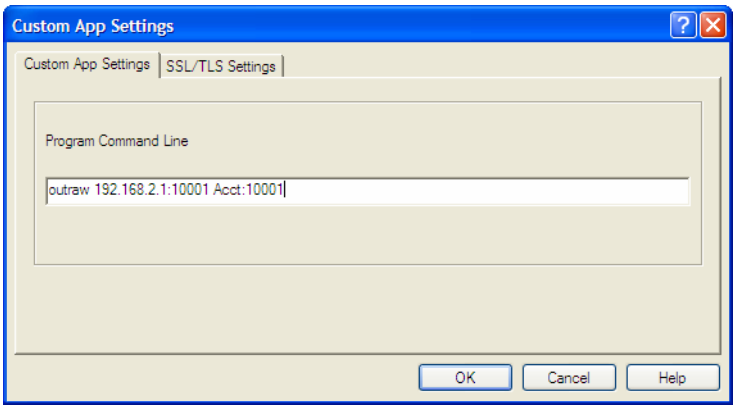
When **Range Mode** is **Host** and you have sequential Modbus Slave IP addresses (for example, 10.10.10.1, 10.10.10.2, 10.10.10.3, etc.), you can specify a UID range and the Device Server will automatically increment the last digit of the configured IP address. Therefore, you can specify a UID range of 1-100, and the Device Server will route Master Modbus messages to all Modbus Slaves with IP addresses of 10.10.10.1 - 10.10.10.100.

Range Mode

If you specify **Host**, the IP address is used for the first UID specified in the range. The last octect in the IPv4 address is then incremented for subsequent UID's in that range. The **Host** option is not applicable for IPv6 addresses. If you specify **Gateway**, the Modbus Master Gateway will use the same IP address when connecting to all the remote Modbus slaves in the specified UID range.

Custom App Settings

You can create a custom application that can run on a specific serial line in Device Server using the Perle SDK.



Configure the following parameter:

Program Command Line

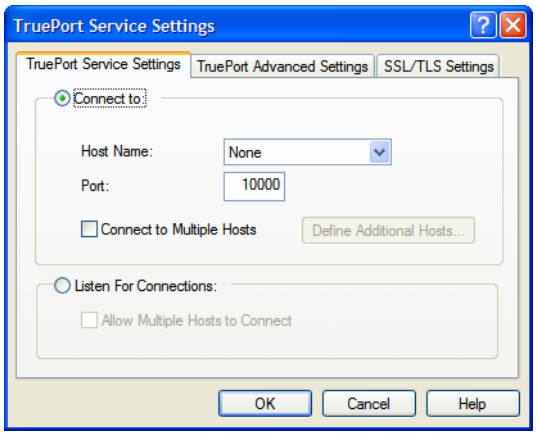
The name of the SDK program executable that has been already been downloaded to the Device Server, plus any parameters you want to pass to the program. Maximum of 80 characters. Use the **`shell`** CLI command as described in the *SDK Programmer's Guide* to manage the files that you have downloaded to the Device Server. For example, using sample outraw program, you would type:

`outraw 192.168.2.1:10001 Acct:10001`

if you were starting the application on a line.

TruePort Settings

When the **Line Service** is set to **TruePort**, data is sent through the connection in its original format. This raw TCP/IP connection can be initiated from the Device Server to the configured host or from the host to the Device Server, depending on the settings.

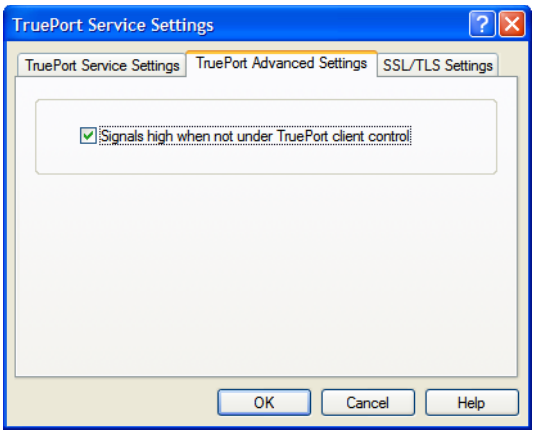


Configure the following parameters:

Connect to	Specify this option when the Device Server is initiating the connection to the TruePort host. This is the default.
Host Name	The name of the target host.
Port	The port number the target host is listening on for incoming connections.
Connect to Multiple Hosts	When enabled, allows a serial device connected to this serial port to communicate to multiple hosts running TruePort Lite.
Define Additional Hosts Button	Click this button to define the hosts that this serial port will connect to.
Listen For Connections	When enabled, allows the TruePort client to initiate communication to the Device Server.
Allow Multiple Hosts to Connect	When this option is enabled, multiple hosts can connect to a serial device that is connected to this serial port.

TruePort Advanced Tab

This setting affects the EIA-232 signals on the Device Server's TruePort configured port.



Configure the following parameter:

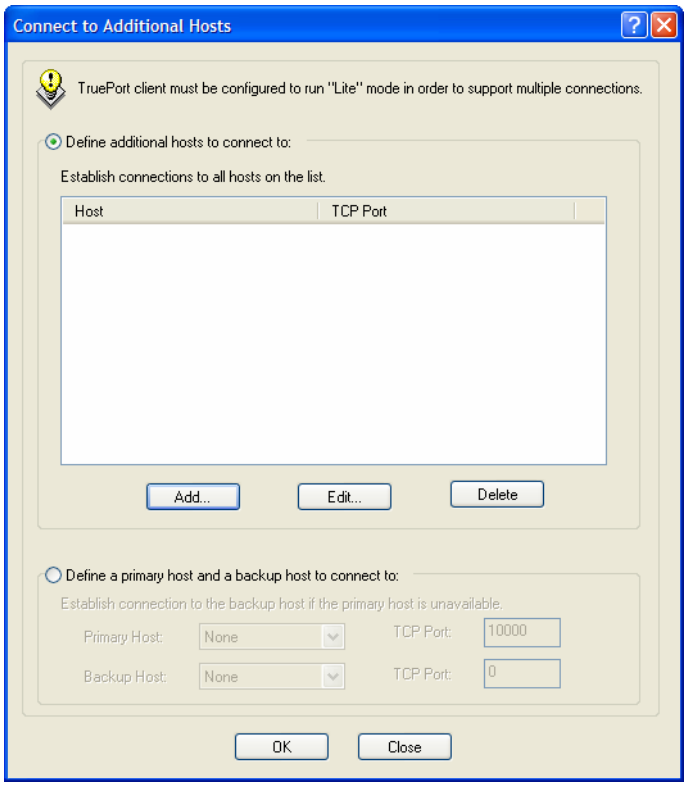
Signals high when... When a TruePort line becomes active, this option has the following impact:

- **TruePort Lite Mode**—When enabled, the EIA-232 signals remain high (active). When disabled, the EIA-232 signals remain low (inactive).
- **TruePort Full Mode**—Same as TruePort Lite Mode, except that when the TruePort client connects to the Device Server TruePort port, the TruePort client application can control the state of the EIA-232 signals.

Default: Enabled

TruePort Multihost

You can define a list of hosts that the serial device will communicate to through TruePort Lite or a primary/backup host.



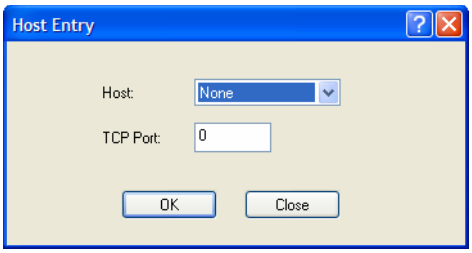
Configure the following parameters:

Define additional hosts to connect to	When this option is enabled, you can define up to 49 hosts that the serial device connected to this serial port will attempt communicate to.
Add Button	Click the **Add** button to add a host to the list of hosts that will be receiving communication from the serial device connected to the Device Server.
Edit Button	Highlight an existing host and click the **Edit** button to edit a host in the list of hosts that will be receiving communication from the serial device connected to the Device Server.
Delete Button	Click the **Delete** button to delete a host to the list of hosts that will be receiving communication from the serial device connected to the Device Server.
Define a primary host and backup...	When this option is enabled, you need to define a primary host that the serial device connected to this serial port will communicate to and a backup host, in the event that the Device Server looses communication to the primary host.
Primary Host	Specify a preconfigured host that the serial device will communicate to through the Device Server.
TCP Port	Specify the TCP port that the Device Server will use to communicate to the **Primary Host**.
Backup Host	Specify a preconfigured host that the serial device will communicate to through the Device Server if the Device Server cannot communicate with the **Primary Host.**

TCP Port Specify the TCP port that the Device Server will use to communicate to the **Backup Host**.

Adding/Editing a Multihost Entry

When you click the **Add** or **Edit** button, the Host Entry window appears. The hosts in the multihost list must already be defined (see ***Configuring Hosts*** on page 219) If you add a host that was defined with its fully qualified domain name (FQDN), it must be resolvable by your configured DNS server



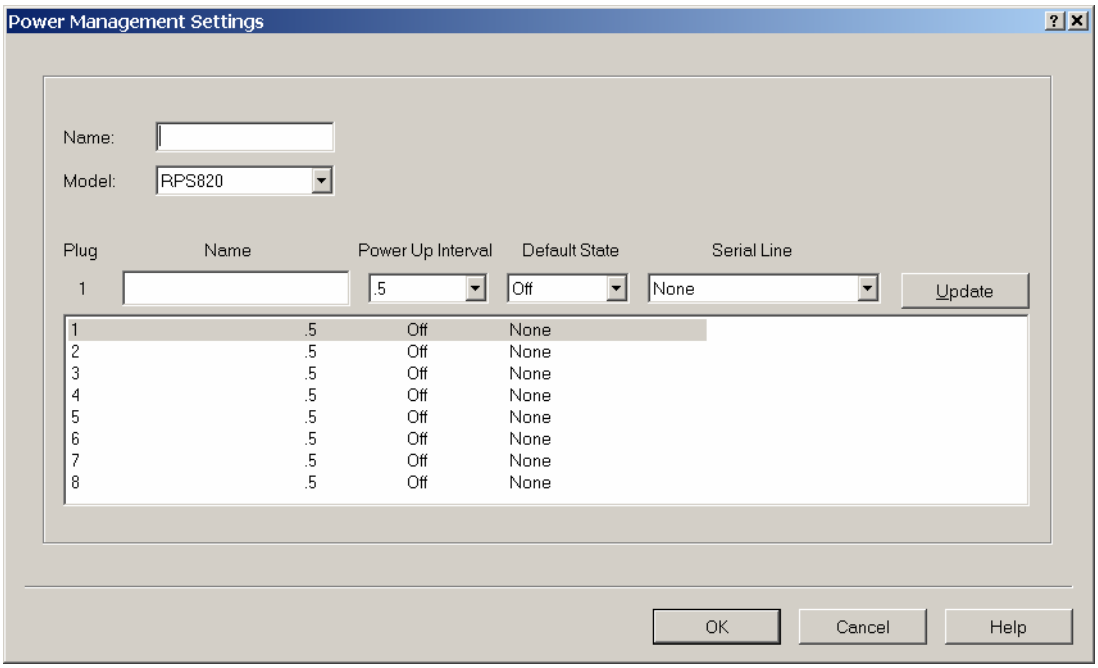
Configure the following parameters:

Host Name Enter the name of the target host.

TCP Port The port number the target host is listening on for incoming connections.

Power Management Settings

When the **Line Service** is set to **Power Management**, it indicates that the line has a serial connection to a Perle Remote Power Switch (RPS).



Configure the following parameters:

Name Specify a name for the RPS.

Model Specify the power bar model, either RPS820, RPS830, RPS1620, RPS1630.

Plug Displays the plug number you are configuring.

Name Specify a name for the plug to make it easier to recognize and manage.

Power Up Interval Specify the amount of time, in seconds, that the power bar will wait before powering up a plug. This can be useful if you have peripherials that need to be started in a specific order. The default is .5 seconds.

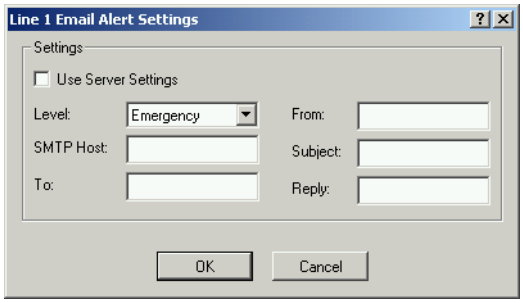
Default State Sets the default state of the plug, either **on** or **off**. The default is **off**.

Serial Line Associate a serial line(s) connected to a serial device that is plugged into the power bar on that plug.

Update Button Updates the plug's settings.

Configuring Line Email Alerts

Line email alerts are specific to events that occur on the line. An email is sent to the specified recipient(s) when an event occurs that meets the **Level** criteria.



Configure the following parameters:

User Server Settings Determines whether you want the **Line** to inherit the **Email Alert** settings from the **Server Email Alert**. If this is enabled, **Server** and **Line** notification events will have the same **Email Alert** setting.

Level Choose the event level that triggers an email notification:

- **Emergency**
- **Alert**
- **Critical**
- **Error**
- **Warning**
- **Notice**
- **Info**
- **Debug**

You are selecting the lowest notification level; therefore, when you select **Debug**, you will get an email notification for all events that trigger a message.

SMTP Host The SMTP host (email server) that will process the email notification request. This can be either a host name defined in the Device Server host table or the SMTP host IP address.

To An email address or list of email addresses that will receive the email notification.

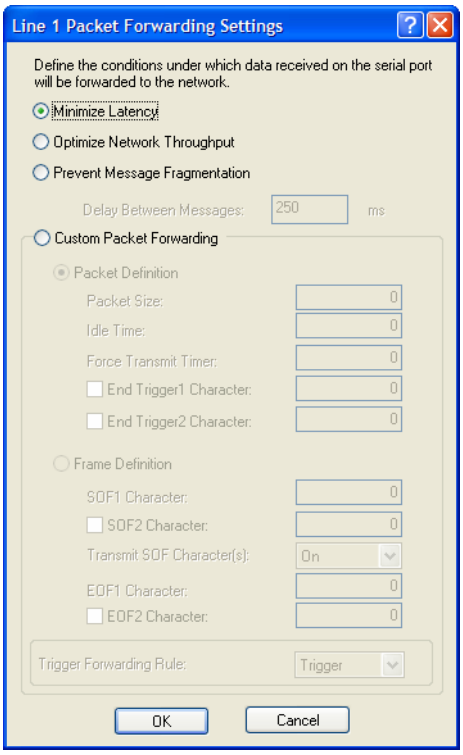
From This field can contain an email address that might identify the Device Server name or some other value.

Subject A text string, which can contain spaces, that will display in the **Subject** field of the email notification.

Reply To The email address to whom all replies to the email notification should go.

Packet Forwarding

The Packet Forwarding feature allows you to control how the data coming from a serial device is packetized before forwarding the packet onto the LAN network.



Configure the following parameters:

Minimize Latency This option ensures that all application data is immediately forwarded to the serial device. Select this option for timing-sensitive applications.

Optimize Network Throughput This option provides optimal network usage while ensuring that the application performance is not compromised. Select this option when you want to minimize overall packet count, such as when the connection is over a WAN.

Prevent Message Fragmentation This option detects the message, packet, or data blocking characteristics of the serial data and preserves it throughout the communication. Select this option for message-based applications or serial devices that are sensitive to inter-character delays within these messages.

Delay Between Messages The minimum time, in milliseconds, between messages that must pass before the data is forwarded by the Device Server. The range is 0-65535. The default is 250 ms.

Custom Packet Forwarding This option allows you to define the packet forwarding rules based on the packet definition or the frame definition.

Packet Definition

This section allows you to set a variety of packet definition options. The first criteria that is met causes the packet to be transmitted. For example, if you set a **Force Transmit Timer** of **1000** ms and a **Packet Size** of **100** bytes, whichever criteria is met first is what will cause the packet to be transmitted.

Packet Size

The number of byte that must be received from the serial port before the packet is transmitted to the network. A value of zero (0) ignores this parameter. Valid values are 0-1024 bytes. The default is 0.

Idle Time

The amount of time, in milliseconds, that must elapse between characters before the packet is transmitted to the network. A value of zero (0) ignores this parameter. Valid values are 0-65535 ms. The default is 0.

Force Transmit Timer

When the specified amount of time, in milliseconds, elapses after the first character is received from the serial port sender, the packet is transmitted. A value of zero (0) ignores this parameter. Valid values are 0-65535 ms. The default is 0.

End Trigger1 Character

When enabled, specifies the character that when received will define when the packet is ready for transmission. The actual transmission of the packet is based on the Trigger Forwarding Rule. Valid values are in hex 0-FF. The default is 0.

End Trigger2 Character

When enabled, creates a sequence of characters that must be received to specify when the packet is ready for transmission (if the End Trigger1 character is not immediately followed by the End Trigger2 character, the Device Server waits for another End Trigger1 character to start the End Trigger1/End Trigger2 character sequence). The actual transmission of the packet is based on the Trigger Forwarding Rule. Valid values are in hex 0-FF. The default is 0.

Frame Definition

This section allows you to control the frame that is transmitted by defining the start and end of frame character(s). If the internal buffer (1024 bytes) is full before the EOF character(s) are received, the packet will be transmitted and the EOF character(s) search will continue. The default frame definition is SOF=00 and EOF=00.

SOF1 Character

When enabled, the Start of Frame character defines the first character of the frame, any character(s) received before the Start of Frame character is ignored. Valid values are in hex 0-FF. The default is 0.

SOF2 Character

When enabled, creates a sequence of characters that must be received to create the start of the frame (if the SOF1 character is not immediately followed by the SOF2 character, the Device Server waits for another SOF1 character to start the SOF1/SOF2 character sequence). Valid values are in hex 0-FF. The default is 0.

Transmit SOF Character(s)

When enabled, the SOF1 or SOF1/SOF2 characters will be transmitted with the frame. If not enabled, the SOF1 or SOF1/SOF2 characters will be stripped from the transmission.

EOF1 Character

Specifies the End of Frame character, which defines when the frame is ready to be transmitted. The actual transmission of the frame is based on the Trigger Forwarding Rule. Valid values are in hex 0-FF. The default is 0.

EOF2 Character

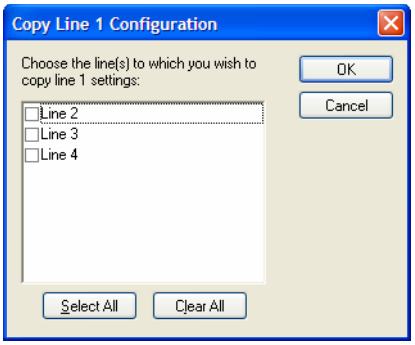
When enabled, creates a sequence of characters that must be received to define the end of the frame (if the EOF1 character is not immediately followed by the EOF2 character, the Device Server waits for another EOF1 character to start the EOF1/EOF2 character sequence), which defines when the frame is ready to be transmitted. The actual transmission of the frame is based on the Trigger Forwarding Rule. Valid values are in hex 0-FF. The default is 0.

Trigger Forwarding Rule Determines what is included in the Frame (based on the EOF1 or EOF1/EOF2) or Packet (based on Trigger1 or Trigger1/Trigger2). Choose one of the following options:

- **Strip-Trigger**—Strips out the EOF1, EOF1/EOF2, Trigger1, or Trigger1/Trigger2, depending on your settings.
- **Trigger**—Includes the EOF1, EOF1/EOF2, Trigger1, or Trigger1/Trigger2, depending on your settings.
- **Trigger+1**—Includes the EOF1, EOF1/EOF2, Trigger1, or Trigger1/Trigger2, depending on your settings, plus the first byte that follows the trigger.
- **Trigger+2**—Includes the EOF1, EOF1/EOF2, Trigger1, or Trigger1/Trigger2, depending on your settings, plus the next two bytes received after the trigger.

Copying Line Settings to Another Line(s)

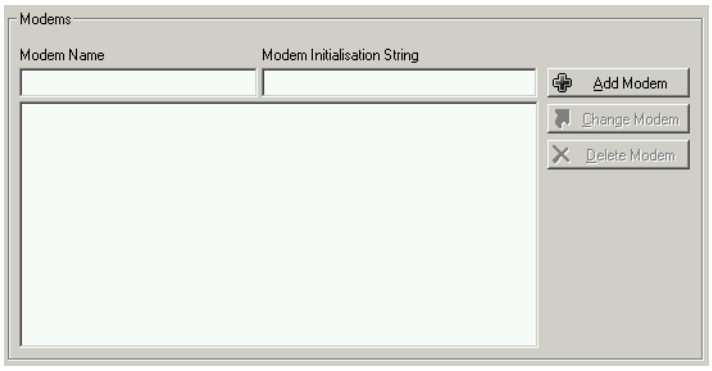
On 4-port+ models, you can selectively copy the current line's setting to another line(s). Click the **Save & Copy Line Configuration** button to display the Copy Line window.



Check the boxes of the lines you want to copy the current line's setting to or click **Select All** to select all the lines; you can also clear all the selections by clicking **Clear All**. When you are done, click **OK** to copy the settings to the selected line(s) or **Cancel** to exit the window without copying any line settings.

Configuring Modems

You need to configure a modem if there is a modem connected to the Device Server. If your Device Server model contains an internal modem or a PCI slot for a modem card, a permanent modem string exists in your configuration.



Configure the following parameters:

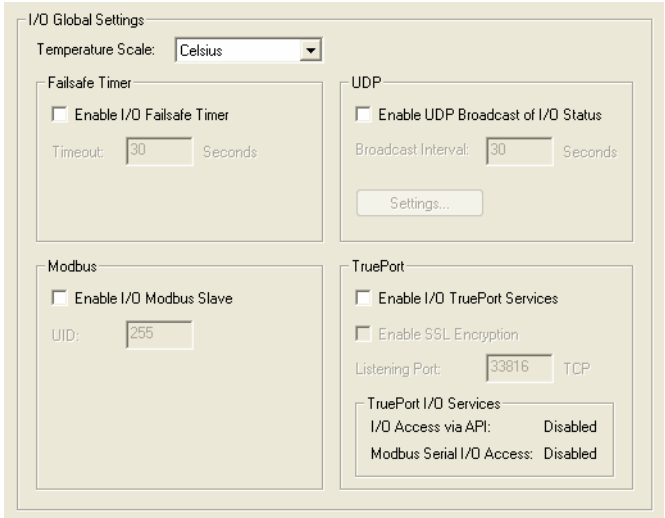
Modem Name	The name of the modem. Do not use spaces.
Modem Initialisation String	The initialisation string of the modem; see your modem's documentation.

Configuring I/O

This configuration entry will appear in the DeviceManager only when connected to a Device Server that supports I/O or when an I/O model configuration is selected.

Global Settings

The I/O Global Settings enable/disable options that support all I/O channels on your Device Server.



Temperature Settings

Temperature Scale Select the temperature scale that will be used to display temperature data, either Fahrenheit or Celsius. The default is Celsius.

This option is for Temperature channels only.

Failsafe Timer Settings

The **Failsafe Timer** is activated when no I/O operation occurs within the specified amount of time; I/O operations include everything that can be done manually in the **I/O Status/Control** or programatically.

Enable I/O Failsafe Timer	Enables/disables the **Failsafe Timer**. This is the global setting that must be enabled to set the **Failsafe Action** on the channel for digital outputs and relays. When this timer expires because of no I/O activity within the specified time interval, the **Failsafe Action** set for the channel determines the action on the output.
Timeout	The number of seconds that must elapse with no I/O activity before the channel **Failsafe Action** is triggered. Valid values are 1-9999. The default is 30 seconds.

Modbus Settings

Enabling the Modbus option makes the Device Server act as a Modbus Slave, allowing Modbus Masters to communicate with the Device Server to control and/or retrieve I/O data.

Enable I/O Modbus Slave	Enables/disables Modbus as the communication protocol for all the I/O channels.
UID	This is the UID you are assigning to the Device Server, which is acting as a Modbus slave.

TruePort Settings

These TruePort settings pertain specifically to using TruePort to allow serial Modbus Masters to access Device Server I/O data over the network or allowing a serial application to access the Device Server I/O data over the network using the Perle API (see ***Accessing I/O Data Via TruePort*** on page 378 for more information).

Enable I/O TruePort Service	Enables/disables serial Modbus application access to the I/O over the network using the TruePort COM redirector feature.
Enable SSL Encryption	Enables/disables SSL encryption for the I/O data between the Device Server and the TruePort host.
Listening Port	The TCP port that the Device Server will listen to for I/O channel data requests from TruePort.
I/O Access via API	Displays the access status of being able to access the I/O data via a custom application using the Perle API.
Modbus Serial I/O Access	Displays the status of a serial Modbus Master being able to access the Device Server over the network to access I/O data.

UDP Settings

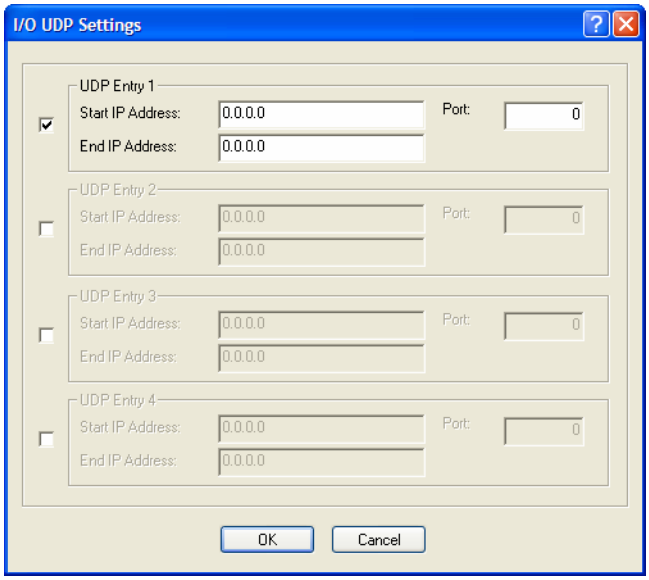
The I/O UDP broadcast feature periodically broadcasts the state of the I/O status in a UDP message.

Enable UDP Broadcast of I/O Status	Enables/disables UDP broadcast of I/O channel status (data).
Broadcast Interval	Enter the interval, in seconds, for UDP broadcasts of I/O channel status (data). Valid values are 1-9999. Default value is 30 seconds.

Setting Button

Click the **Settings** button to access the I/O UDP Broadcast window, where you can define the IP addresses of the receivers of the UDP broadcast.

The I/O UDP settings window allows you to configure the UDP broadcast recipients.



Configure the following parameters:

UDP Entry

When enabled, broadcasts I/O status (data) to the specified range of IP addresses.

Start IP Address

The first host IP address in the range of IP addresses (for IPV4 or IPV6) that the Device Server will listen for messages from and/or send messages to.

End IP Address

The last host IP address in the range of IP addresses (for IPV4, not required for IPV6) that the Device Server will listen for messages from and/or send messages to.

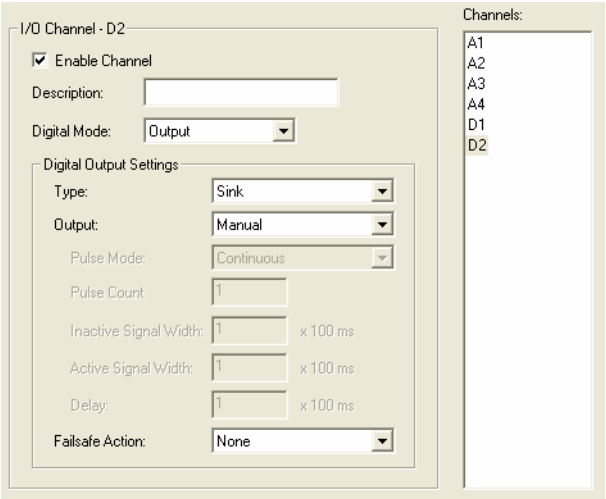
Port

The UDP port that the Device Server will use to relay messages to servers/hosts.

Channels

Digital Output

When the channel is set for digital output, either voltage is applied to the channel or the channel is grounded. Note that the internal jumpers must match the software setting and must be set to Output (by default, they are set to Input); see ***Digital I/O Module*** on page 48 to find out how to set the internal jumpers.



Configure the following parameters:

Enable Channel

Enables the channel, allowing the settings to become active.

Description

Provide a description of the channel, making it easier to identify. The channel description can be up to 20 characters.

Digital Mode

Specify whether the channel will drive the line (output) or will be reading the status of the line (input). The default is **Input**. The internal jumpers must match the software configuration, so if you change this setting to **Output**, you will have to also change the internal hardware jumpers.

Type

Specify the type of digital output:

- **Sink**—Specifies that the channel will be grounded when active.
- **Source**—Specifies that the channel will provide voltage when active.
- **Sink and Source**—Specifies that channel will be grounded when it is inactive and will provide voltage when it is active.

The default is **Sink**.

Output

Specify how the channel output will be handled:

- **Manual**—You must manually manipulate the channel output.
- **Pulse**—Activates and deactivates the channel output activity in intervals after it is manually activated.
- **Inactive-to-Active Delay**—The channel output will remain inactive for the specified time interval after it is manually started.
- **Active-to-Inactive Delay**—The channel output will go inactive after the specified time interval after it is manually started.

The default is **Manual**.

Pulse Mode

When the **Output** is **Pulse**, you can have it pulse in a **Continuous** manner or specify a pulse **Count** (each count consists of an active/inactive sequence). The default is **Continuous**.

Pulse Count

The channel output will pulse for the specified number of times; each count consists of an active/inactive sequence. The default is 1.

Inactive Signal Width

How long the channel will remain inactive during pulse mode. Valid values are 1-9999 x 100 ms. The default is 100 ms.

Active Signal Width

How long the channel will be active during the pulse mode. Valid values are 1-9999 x 100 ms. The default is 100 ms.

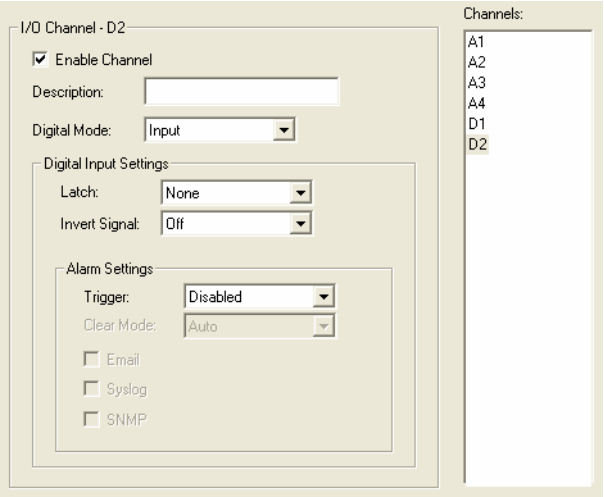
Failsafe Action

When there has been no I/O activity within the specified time (set in the Global Settings) and the **Failsafe Timer** is triggered, you can set the **Failsafe Action** to:

- **None**—The state of the Digital/Relay output remains the same, no change.
- **Activate Output**—Activates the channel.
- **Deactivate Output**—Deactivates the channel.

Digital Input

When the channel is set for digital input, it monitors voltage or current. Note that the internal jumpers must match the software setting and must be set to Input, which is the default; see ***Digital I/O Module*** on page 48 to find out how to set the internal jumpers.



Configure the following parameters:

Enable Channel

Enables the channel, allowing the settings to become active.

Description

Provide a description of the channel, making it easier to identify. The channel description can be up to 20 characters.

Digital Mode

Specify whether the channel will drive the line (output) or will be reading the status of the line (input). The default is **Input**. The internal jumpers must match the software configuration, so if you change this setting to **Output**, you will have to also change the internal hardware jumpers.

Latch

Latches (remembers) the activity transition (active to inactive or inactive to active). The default is None.

Invert Signal Inverts the actual condition of the I/O signal in the status; therefore, an inactive status will be displayed as active.

Trigger When the trigger condition is met, triggers the specified alarm action. Triggers can be:

- **Disabled**—No alarm settings. This is the default.
- **Inactive**—When the expected Digital input is active, going inactive will trigger an alarm.
- **Active**—When the expected Digital input is inactive, going active will trigger an alarm.

Clear Mode Specify **Manual** to manually clear an alarm. Specify **Auto** to automatically clear the alarm when the trigger condition changes; for example, if the **Trigger** is **Inactive** and the alarm is triggered, once the input becomes active again, the alarm will be cleared when **Auto** is set. The default is **Auto**.

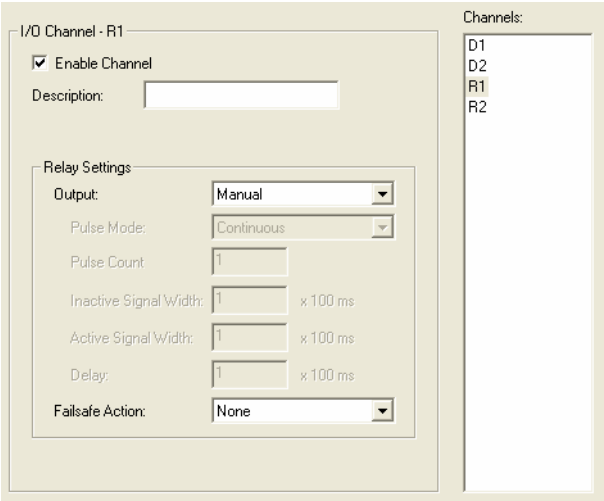
Email Sends an email alert to an email account(s) set up in the Server settings (the Line Email Alert settings are not used with this feature) when an alarm is triggered or cleared. The email alert data includes the severity level and the value that caused the alarm to trigger or clear. The **Email Alert** is associated with **Level Critical**.

Syslog Sends a message to syslog when an alarm is triggered or cleared. The syslog entry includes the severity level and the value that caused the alarm to trigger or clear. The syslog message is associated with **Level Critical**.

SNMP Sends an SNMP trap when an alarm is triggered or cleared. The trap consists of the severity level and whether the alarm was triggered or cleared.

Relays

Relay channels can open or close a contact for a higher voltage circuit using a lower level control voltage.



Configure the following parameters:

Enable Channel Enables the channel, allowing the settings to become active.

Description Provide a description of the channel, making it easier to identify. The channel description can be up to 20 characters.

Output

Specify how the channel output will be handled:

- **Manual**—You must manually manipulate the channel output.
- **Pulse**—Activates and deactivates the channel output activity in intervals after it is manually activated.
- **Inactive-to-Active Delay**—The channel output will remain inactive for the specified time interval after it is manually started.
- **Active-to-Inactive Delay**—The channel output will go inactive after the specified time interval after it is manually started.

The default is **Manual**.

Pulse Mode

When the **Output** is **Pulse**, you can have it pulse in a **Continuous** manner or specify a pulse **Count** (each count consists of an active/inactive sequence). The default is **Continuous**.

Pulse Count

The channel output will pulse for the specified number of times; each count consists of an active/inactive sequence. The default is 1.

Inactive Signal Width

How long the channel will remain inactive during pulse mode. Valid values are 1-9999 x 100 ms. The default is 100 ms.

Active Signal Width

How long the channel will be active during the pulse mode. Valid values are 1-9999 x 100 ms. The default is 100 ms.

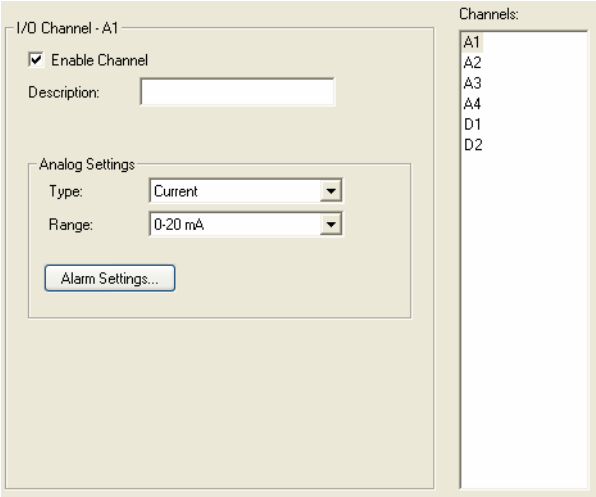
Failsafe Action

When there has been no I/O activity within the specified time (set in the Global Settings) and the **Failsafe Timer** is triggered, you can set the **Failsafe Action** to:

- **None**—The state of the Digital/Relay output remains the same, no change.
- **Activate Output**—Activates the channel.
- **Deactivate Output**—Deactivates the channel.

Analog

Analog channels monitor current/voltage input. Note that the internal jumpers must match the software setting (by default, they are set to Current); see ***Analog Input Module*** on page 49 to find out how to set the internal jumpers.

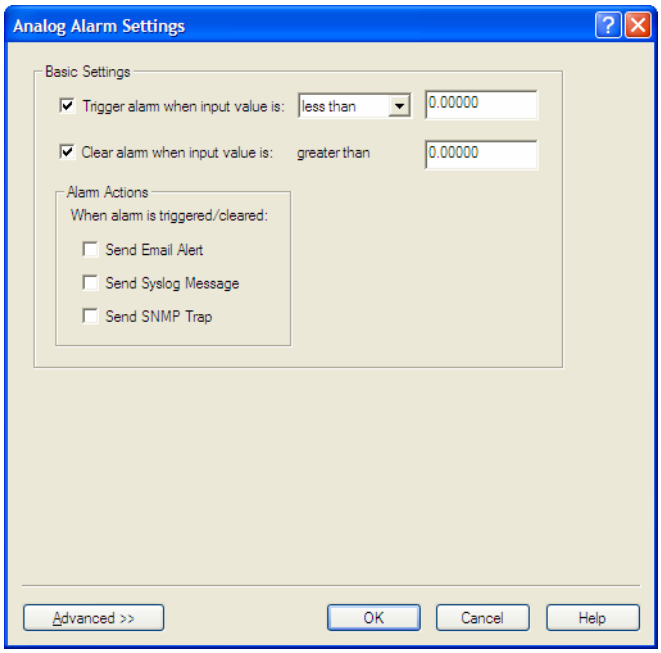


Configure the following parameters:

Enable Channel	Enables the channel, allowing the settings to become active.
Description	Provide a description of the channel, making it easier to identify. The channel description can be up to 20 characters.
Type	Select the type of input being measured, either **Current** or **Voltage**. The default is **Current**.
Range	Select the range for the measurement type. The default for **Current** is **`0-20mA`**. The default for **Voltage** is **`+/-10V`**.

Basic Alarm Settings

The basic Analog Alarm Settings window allows you to configure one severity alarm, whereas the advanced window allows you to configure up to five severity alarm levels.



Configure the following parameters:

Trigger alarm when input value is Specify the value that will trigger an alarm, the measurement is based on the **Type** and **Range** that you specify. This value must not fall within the scope of the value used to clear an alarm.

Clear alarm when input value is Specify that value that will clear an alarm, the measurement is based on the **Type** and **Range** that you specify. This value must not fall within the scope of the value used to trigger an alarm.

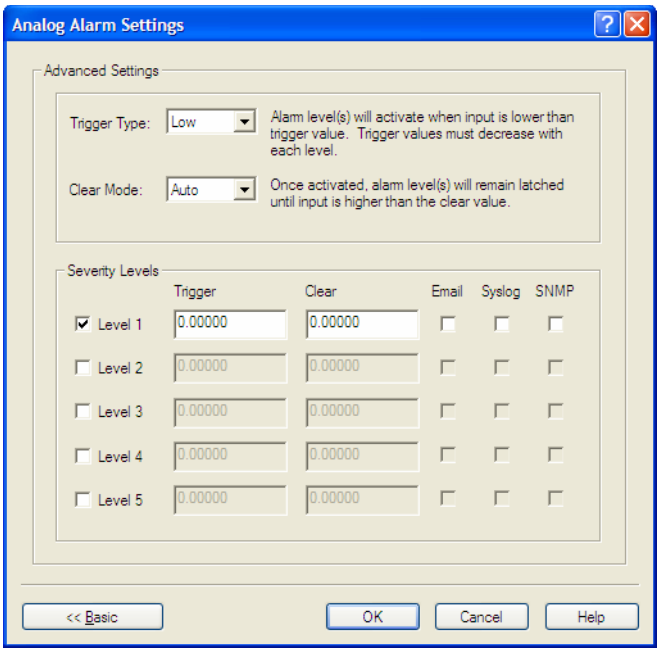
Send Email Alert Sends an email alert to an email account(s) set up in the Server settings (the **Line Email Alert** settings are not used with this feature) when an alarm is triggered or cleared. The email alert data includes the severity level and the value that caused the alarm to trigger or clear. The Email Alert is associated with **Level Critical**.

Send Syslog Alert Sends a message to syslog when an alarm is triggered or cleared. The syslog entry includes the severity level and the value that caused the alarm to trigger or clear. The syslog message is associated with **Level Critical**.

Send SNMP Alert Sends an SNMP trap when an alarm is triggered or cleared. The trap consists of the severity level and whether the alarm was triggered or cleared.

Advanced Alarm Settings

The advanced Analog Alarm Settings window expands the basic alarm settings options to up to five severity levels.



Configure the following parameters:

Trigger Type

If the **Trigger Type** is **Low**, an alarm is triggered when the input drops below the specified **Trigger** value; other severity level trigger values must decrease in value with each subsequent level. If the **Trigger Type** is **High**, an alarm is triggered when the input is higher than the specified **Trigger** value; other severity level trigger values must increase in value with each subsequent level.

Clear Mode

To clear an alarm, the input must drop below the specified value when **Trigger Type** is **High** or go above the specified value when **Trigger Type** is **Low**.

Level 1-5

Defines the Level severity settings for up to five levels. If the **Trigger Type** is **Low**, an alarm is triggered when the input drops below the specified **Trigger** value; other severity level trigger values must decrease in value with each subsequent level. If the **Trigger Type** is **High**, an alarm is triggered when the input is higher than the specified **Trigger** value; other severity level trigger values must increase in value with each subsequent level.

Trigger

If the **Trigger Type** is **Low**, an alarm is triggered when the input drops below the specified **Trigger** value; other severity level trigger values must decrease in value with each subsequent level. If the **Trigger Type** is **High**, an alarm is triggered when the input is higher than the specified **Trigger** value; other severity level trigger values must increase in value with each subsequent level.

Clear

To clear an alarm, the input must drop below the specified value when **Trigger Type** is **High** or go above the specified value when **Trigger Type** is **Low**.

Email

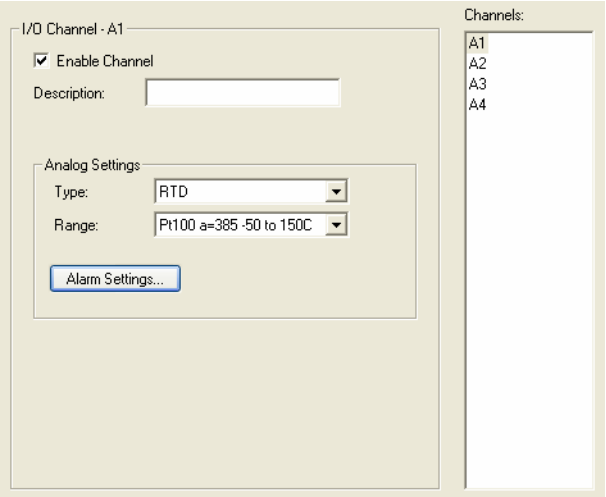
Sends an email alert to an email account(s) set up in the Server settings (the **Line Email Alert** settings are not used with this feature) when an alarm is triggered or cleared. The email alert data includes the severity level and the value that caused the alarm to trigger or clear. The Email Alert is associated with **Level Critical**.

Syslog Sends a message to syslog when an alarm is triggered or cleared. The syslog entry includes the severity level and the value that caused the alarm to trigger or clear. The syslog message is associated with **Level Critical**.

SNMP Sends an SNMP trap when an alarm is triggered or cleared. The trap consists of the severity level and whether the alarm was triggered or cleared.

Temperature

Temperature channels monitor either RTD or thermocouple inputs for the most common ranges.



Configure the following parameters:

Enable Channel Enables the channel, allowing the settings to become active.

Description Provide a description of the channel, making it easier to identify. The channel description can be up to 20 characters.

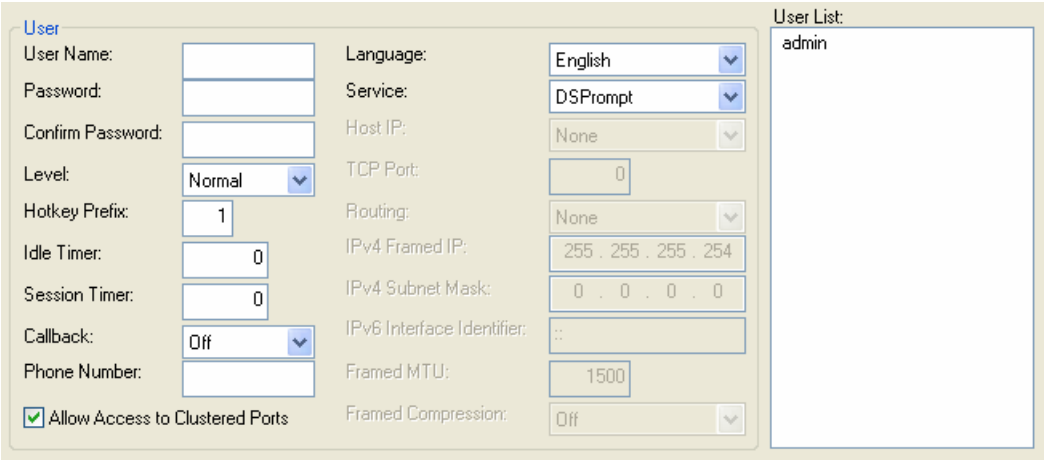
Type Specify the type of sensor you are using to measure temperature, either RTD or thermocouple. The default is RTD.

Range Specify the temperature range that you want to measure. For RTD, the default is Pt100 a=385 -50 to 150C. For thermocouple, the default is J 0 to 760C.

See ***Basic Alarm Settings*** on page 211 and ***Advanced Alarm Settings*** on page 212 for the Alarm Settings information.

Configuring Users

You can configure up to four users in the Device Server's local user database for all DS, SDS, and SCS 1-port to 4-port desktop models, in addition to the Admin user. You can configure up to 48 users in the Device Server's local user database for all STS models and 8-port+ SCS and SDS models, in addition to the Admin user.



Configure the following parameters:

User Name — The name of the user. Do not use spaces. This case-sensitive field accepts a maximum of 16 characters.

Password — The password the user will need to enter to login to the Device Server. This case-sensitive field accepts a maximum of 16 characters.

Confirm Password — Enter the user's password again to verify it is entered correctly.

Level — The access that a user is allowed:

- **Admin**—The admin level user has total access to the Device Server. You can create more than one admin user account but we recommend that you only have one. They can monitor and configure the Device Server.
- **Normal**—The Normal level user has limited access to the Device Server. Limited CLI commands and Menu access are available with the ability to configure the user's own configuration settings.
- **Restricted**—The Restricted level user can only access predefined sessions or access the Easy Port Access menu.
- **Menu**—The menu level user will only be able to access predefined session or access the Easy Port Access menu. The Easy Port Access allows the user to connect to the accessible line without disconnecting their initial connection to the Device Server. Does not have any access to CLI commands.

Hotkey Prefix

The prefix that a user types to control the current session. The default value is **hex 01**, which corresponds to **Ctrl-a** (**^a**) (hex value 02 would be Ctrl-b (^b), etc.):

- **^a *number***—To switch from one session to another, press **^a** and then the required session number. For example, **^a 2** would switch you to session 2. Pressing **^a 0** will return you to the Device Server Menu.
- **^a n**—Display the next session. The current session will remain active. The lowest numbered active session will be displayed.
- **^a p**—Display the previous session. The current session will remain active. The highest numbered active session will be displayed.
- **^a m**—To exit a session and return to the Device Server. You will be returned to where you left off. The session will be left running.
- **^a l**—(Lowercase L) Locks the line until the user unlocks it. The user is prompted for a password (any password, excluding spaces) and locks the line. Next, the user must retype the password to unlock the line.
- **^r**—When you switch from a session back to the Menu, the screen may not be redrawn correctly. If this happens, use this command to redraw it properly. This is always **Ctrl R**, regardless of the **Hotkey Prefix**.

The **User Hotkey Prefix** value overrides the **Line Hotkey Prefix** value. You can use the **Hotkey Prefix** keys to lock a line only when the line **Lock** parameter is **On**.

Idle Timer

The amount of time, in seconds, that the **Idle Timer** will run. Use this timer to close a connection because of inactivity. When the **Idle Timer** expires, because there has been no exchange of data within the specified time, the Device Server will close the connection. The default value is **0** (zero), meaning that the **Idle Timer** will not expire (the connection is open permanently). The maximum value is 4294967 seconds. The **User Idle Timer** will override the **Line Idle Timer**, with the exception of reverse SSH or reverse Telnet sessions.

Session Timer

The amount of time, in seconds, that the **Session Timer** will run. Use this timer to forcibly close a user's session (connection). When the **Session Timer** expires, the Device Server will end the connection. The default value is **0** (zero), meaning that the session timer will not expire (the session is open permanently, or until the user logs out). The maximum value is 4294967 seconds. The **User Session Timer** will override the **Line Session Timer**, with the exception of reverse SSH or reverse Telnet sessions.

Callback

When **On**, enter a phone number for the Device Server to call the user back (the **Callback** parameter is unrelated to the **Line Dial** parameter).

Note: the Device Server will allow callback only when a user is authenticated. If the protocol over the link does not provide authentication, there will be no callback. Therefore, when the **Line Service** is set to **PPP**, you must use either **PAP** or **CHAP**, because these protocols provide authentication. The default is **Off**.

The Device Server supports another type of callback, **Roaming Callback**, which is configurable when the **Line Service** is set to **PPP**.

Phone Number

The phone number the Device Server will dial to callback the user (you must have set **Callback** to **On**). Enter the number without spaces.

Allow Access to Clustered Ports

When enabled, allows the user access to Device Servers that have been configured in the clustering group. The default is on.

Language

You can specify whether a user will use **English** or **Customlang** as the language that appears in the Menu, CLI, or WebManager. The Device Server supports one custom language that must be downloaded to the Device Server; otherwise, **Customlang** defaults to English.

Service

The type of service that the user will use.

Host IP

When the **User Service** is set to **Telnet**, **Rlogin**, **SSH**, or **TCP_clear**, the target host IP address or preconfigured host name. If no IP address is specified, the **Host IP** value in the **Default User** configuration will be used. The default is **0.0.0.0**. or None.

TCP Port

When the **User Service** is **Telnet**, **TCP_clear**, or **SSH**, this is the target port number. The default value will change based on the type of **Service** selected; the most common known port numbers are used as the default values.

Routing

Determines the routing mode used for RIP packets on the PPP and SLIP interfaces. Values are:

- **None**—RIP packets are neither received nor sent by the Device Server.
- **Send**—RIP packets can only be sent by the Device Server.
- **Listen**—RIP packets can only be received by the Device Server.
- **Send and Listen**—RIP packets are sent and received by the Device Server.

IPv4 Framed IP

Used for **User Service PPP** or **SLIP**, sets the IP address of the remote user. Enter the address in dot decimal notation as follows:

- **255.255.255.254** (default)—The Device Server will use the **Remote IP Address** set in the **PPP** settings for the line.
- **255.255.255.255**—When the **User Service** is **PPP**, the Device Server will allow the remote machine to specify its IP address (overriding the IP address negotiation value configured in the **PPP** settings).
- **255.255.255.255**—When the **User Service** is **SLIP**, the Device Server will use the **Remote IP Address** set for the line (no negotiation).
- **n.n.n.n**—(where **n** is a number) Enter the IP address of your choice. This IP address will then be used in preference to the **Remote IP Address** set for a line.

IPv4 Subnet Mask

If the remote user is on a subnet, enter the network's subnet mask. For example, a subnet mask of 255.255.0.0.

IPv6 Interface Identifier

Used for **User Service PPP**, sets the IP address of the remote user. Enter the address in IPv6 format. The first 64 bits of the Interface Identifier must be zero, therefore, ::abcd:abcd:abcd:abcd is the expected format.

Framed MTU

Used for **User Service PPP** or **SLIP**, specifies the maximum size of packets, in bytes, being transferred across the link. On noisy links it might be preferable to fragment large packets being transferred over the link, since there will be quicker recovery from errors. Depending on whether you have selected a **User Service** of **SLIP** or **PPP**, details are as follows:

- **PPP**—**Framed MTU** will be the maximum size of packets that the Device Server port will accept. This value is negotiated between the two ends of the link. The valid range is 64-1500. The default value is **1500** bytes.
- **SLIP**—**Framed MTU** will be the maximum size of packets being sent by the Device Server. The Device Server will send SLIP packets in the range 256-1006 bytes. The default value is **256** bytes.

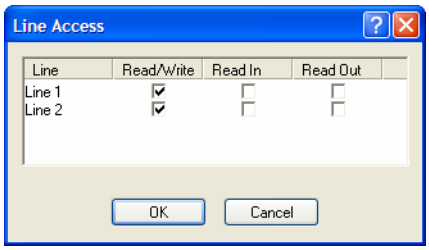
The **Framed MTU** value will be used in preference to the **MTU**/**MRU** values set for a **Line**.

Framed Compression

Used for **User Service PPP** or **SLIP**, determines whether Van Jacobsen Compression is used on the link. VJ compression is a means of reducing the standard TCP/IP header from 40 octets to approximately 5 octets. This gives a significant performance improvement, particularly when interactive applications are being used. For example, when the user is typing, a single character can be passed over the link with a 40 octet header attached. VJ Compression has little effect on other types of links, such as ftp, where the packets are much larger. The **Framed Compression** value will be used in preference to the **VJ Compression** value set for a **Line**. The default is **Off**.

Configuring Line Access

Line Access defines the read/write privileges that a user has while accessing a line.



Configure the following options:

Line Access

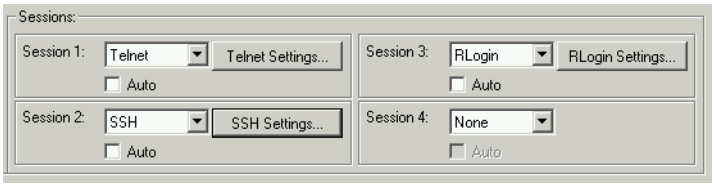
Specifies the user access rights to each Device Server device line. Options are:

- **Read/Write**—Users are given read and write access to the line.
- **Read In**—Users are given access to read only outbound data, data that is going from the Device Server to the device.
- **Read Out**—Users are given access to read only inbound data, data that is going from the device to the Device Server.

Users can read data going in both directions by selecting both the **Read In** and **Read Out** options.

Configuring Sessions

You can configure user **Sessions** to limit the access the user has to the network and the way the user connects to a host. Users who are **Level Normal** or **Admin** can define **Free Sessions**, in addition to using defined sessions. Users who are **Level Restricted** or **Menu** can only access predefined sessions.



Configure the following parameters:

Session — You can create up to four predefined sessions for each user. You can specify the connection service and its settings for each session.

Auto — Specify whether or not the session(s) will start automatically when the user logs into the Device Server.

The following **Session** connections are available:

- **None**—No connection is configured for this session.
- **Telnet**—For information on the Telnet configuration window, see ***Telnet Settings*** on page 172.
- **Rlogin**—For information on the Rlogin configuration window, see ***Rlogin Settings*** on page 173.
- **SSH**—For information on the SSH configuration window, see ***SSH Client Settings*** on page 182.

Configuring the Default User

The **Default User**'s parameters are the parameters that all users who log into the Device Server will inherit unless they have a local user profile or are authenticated by RADIUS or TACACS+. For example, when a user logs into the Device Server and is externally authenticated, then that user will inherit the **Default User** configuration, unless the external authentication method is RADIUS or TACACS+ and the user's parameters are passed to the Device Server from RADIUS/TACACS+ or the User has a local user profile.

When you add new users to the Device Server, they will initially inherit any parameters set in the **Default User** (the parameters can be changed on a per user basis).

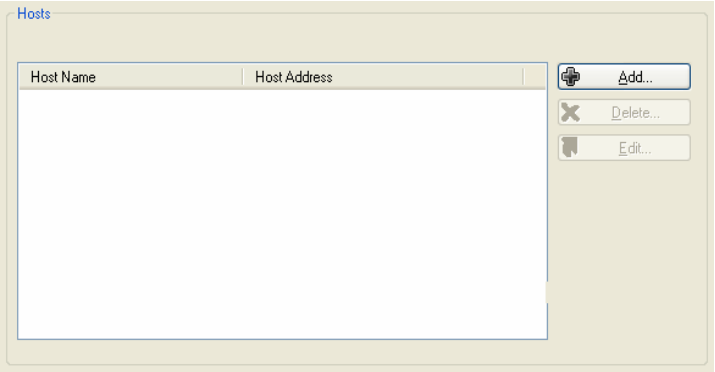
For information on the **Default User** configuration parameters, see ***Configuring Users*** on page 214.

Configuring the Network

The network configuration parameters define the network that the Device Server will be operating within.

Configuring Hosts

One of the first things you will probably want to configure is the hosts that the Device Server or Users will be interacting with, since most configuration windows require that the host already be configured. You can configure up to 20 hosts on all desktop models and up to 40 hosts on rack mount models.

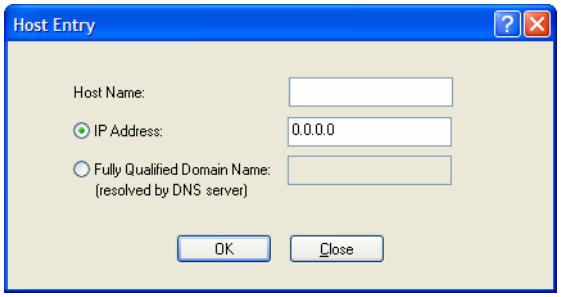


Configure the following parameters:

Add Button	Displays the Host Entry window, which is where the host entry is defined.
Delete Button	Deletes a host from the host table.
Edit Button	Edits a host that already exists in the host table.

Adding/Editing Hosts

When you add a host, you can either specify its IP address or its fully qualified domain name (FQDN). FQDN's must be resolvable by your configured DNS server.

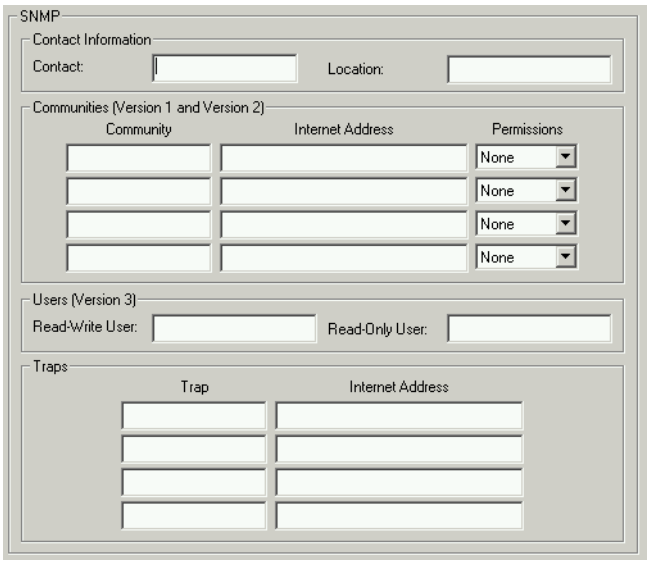


Configure the following parameters:

Host Name	Enter the name of the target host.
IP Address	Specify the IPv4 IP address of the host.
Fully Qualified Domain Name	When you have DNS defined in the Device Server, you can enter a DNS resolvable fully qualified domain name (note: FQDN's are excluded as accessible hosts when **IP Filtering** is enabled).

Configuring SNMP

If you are using the Device Server SNMP MIB-based configuration/management option, you can use the DeviceManager to easily set up SNMP users, traps, and communities. The Device Server supports the SNMP traps for restart and SNMP community authentication error. For more information on SNMP, see ***SNMP*** on page 70.



Configure the appropriate parameters:

Contact — The name and contract information of the person who manages this SMNP node.

Location — The physical location of the SNMP node.

Community — The name of the group that devices and management stations running SNMP belong to.

Internet Address — The IP address of the SNMP manager that will send requests to the Device Server. If the address is `0.0.0.0`, any SNMP manager with the **Community Name** can access the Device Server.

Permissions — Permits the Device Server to respond to SNMP requests by:

- **None**—There is no response to requests from SNMP.
- **Readonly**—Responds only to Read requests from SNMP.
- **Readwrite**—Responds to both Read and Write requests from SNMP.

Read-Write User — Specified user can view and edit SNMP variables.

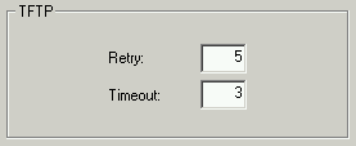
Read-Only User — Specified user can only view SNMP variables.

Trap — The trap receiver is the network management system (NMS) that should receive the SNMP traps. This NMS must have the same SNMP community string as the trap sender.

Internet Address — Defines the hosts (by IP address) that will receive trap messages generated by the Device Server. Up to four trap hosts can be defined.

Configuring TFTP

These parameters configure the TFTP settings for the Device Server's connections to hosts (as opposed to the TFTP settings under **Tools**, **Options**, which configure the TFTP settings for the DeviceManager's connection to a Device Server).



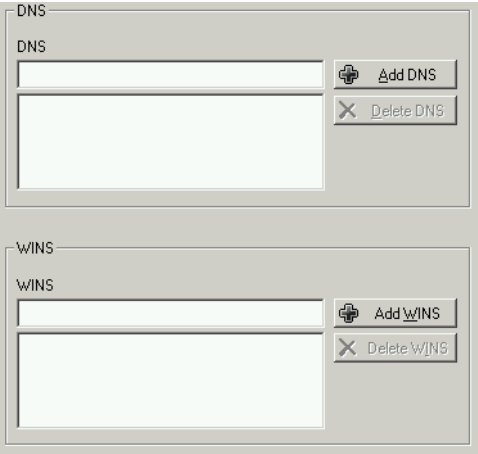
Configure the following parameters:

Retry — The number of times the Device Server will retry to transmit a TPFT packet to/from a host when no response is received. Enter a value between 0 and 5. The default is **5**. A value of **0** (zero) means that the Device Server will not attempt a retry should TFTP fail.

Timeout — The time, in seconds, that the Device Server will wait for a successful transmit or receipt of TFTP packets before retrying a TFTP transfer. Enter a value between 3 and 10. The default is **3** seconds.

Configuring DNS/WINS

You can configure WINS servers for PPP-client name resolution and DNS servers for PPP-client name resolution and Device Server host name resolution (for example, when specifying **Bootup** file).



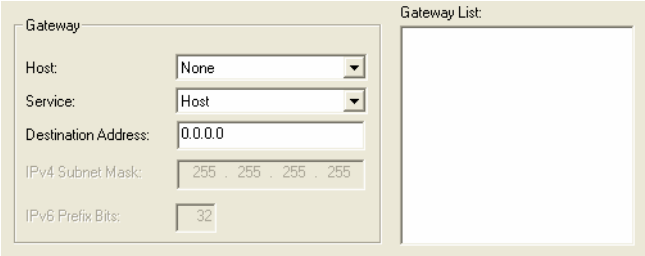
Configure the following parameters:

DNS — You can specify the IP addresses for up to four DNS (Domain Name Servers) hosts in your network.

WINS — You can specify the IP addresses for up to four WINS (Windows Internet Naming Service) hosts in your network.

Configuring Gateways

You can configure gateways to allow the Device Server access to hosts that are not within the local network segment.



Configure the following parameters:

Host

You can specify up to 20 hosts on desktop models and 49 hosts on rack mount models to act as gateways in your network. Each gateway host must be defined in the Device Server's host table.

Service

Specify the type of gateway:

- **Default**—A gateway which provides general access beyond your local network.
- **Host**—A gateway reserved for accessing a specific host external to your local network.
- **Network**—A gateway reserved for accessing a specific network external to your local network.

Destination Address

When the gateway is a **Host** or **Network** gateway, you must specify the IP address of the target host machine/network.

IPv4 Subnet Mask

When the gateway is a **Network** gateway, you must specify the network's subnet mask.

IPv6 Prefix Bits

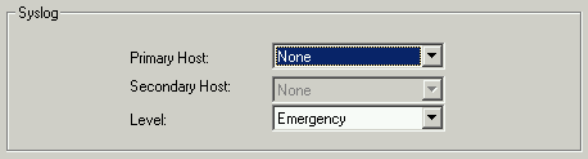
If the IP address is IPv6, then the Prefix Bits range is 0-128.

Gateway List

The list of defined gateways.

Configuring Syslog

You can configure where the system log messages are going to be sent and specify the lowest level message that the Device Server will send syslog messages for.



Configure the following options:

Primary Host

The first preconfigured host that the Device Server will attempt to send system log messages to; messages will be displayed on the host's monitor.

Secondary Host

If the Device Server cannot communicate with the primary host, then the Device Server will attempt to send system log messages to this preconfigured host; messages will be displayed on the host's monitor.

Level

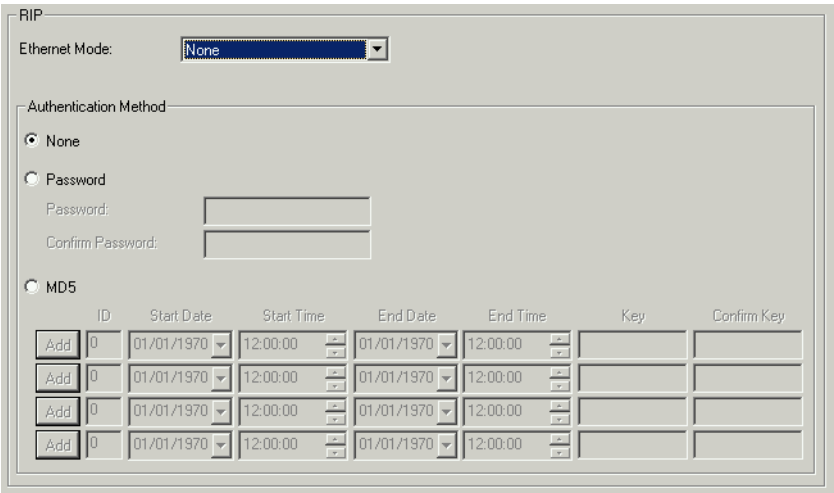
Choose the event level that triggers a syslog entry:

- **Emergency**
- **Alert**
- **Critical**
- **Error**
- **Warning**
- **Notice**
- **Info**
- **Debug**

When you select a **Level**, all the levels that appear above it in the list also trigger a syslog entry. For example, if you select **Error**, all **Error**, **Critical**, **Alert**, and **Emergency** events will be logged.

Configuring RIP

You can configure the Routing Information Protocol (RIP) that will define the communication between the Device Server and the local network (this has no impact on the **Routing** parameters that can be set for PPP in both the **PPP** and **User** configuration windows).



Configure the following parameters:

Ethernet Mode

Enable/disable RIP (Routing Information Protocol) mode for the Ethernet interface with one of the following options:

- **None**—Disables RIP over the Ethernet interface.
- **Send**—Sends RIP over the Ethernet interface.
- **Listen**—Listens for RIP over the Ethernet interface.
- **Send and Listen**—Sends RIP and listens for RIP over the Ethernet interface.

Authentication Method

Specify the type of RIP authentication:

- **None**—No authentication for RIP.
- **Password**—Simple RIP password authentication.
- **MD5**—Use MD5 RIP authentication.

Password

Specify the password that allows the router tables to be updated.

Confirm Password

Retype in the password to verify that you typed in it correctly.

ID

The **MD5** identification key.

Start Date

The start date that the MD5 key becomes valid. The date format is dependent on your system's settings.

Start Time

The time that the MD5 key becomes valid. The time format is dependent on your system's settings.

End Date

The last day that the MD5 key is valid. The date format is dependent on your system's settings.

End Time

The time that the MD5 key becomes invalid. The time format is dependent on your system's settings.

Key

The MD5 key that is being used by your routers.

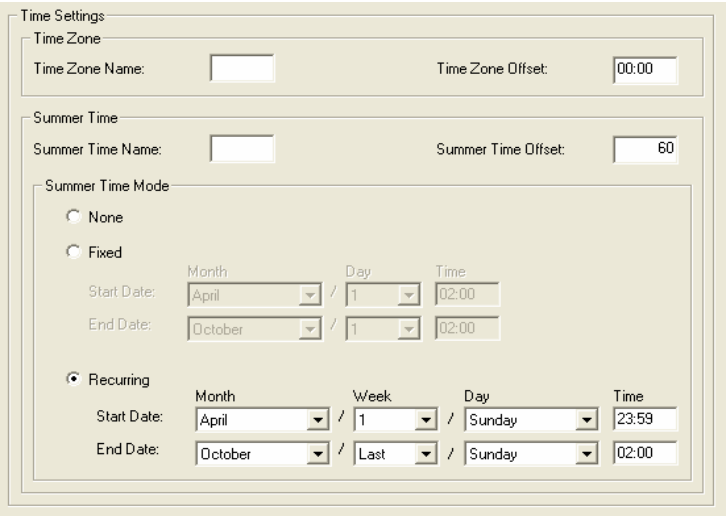
Confirm Key Retype the MD5 key that is being used by your routers to verify that it was typed correctly.

Configuring Time

You can configure an SNTP server to automate the time in the Device Server and configure an automatic summertime (daylight savings time) time change.

Configuring Time Settings

You can configure when the Device Server automatically changes to summer time settings (daylight savings time).



Configure the following parameters:

Time Zone Name The name of the time zone to be displayed during standard time. Maximum 4 characters and minimum 3 characters (do not use angled brackets < >).

Time Zone Offset The offset from UTC for your local time zone. Specify in the format of hours *hh* (valid -12 to +14) and minutes *mm* (valid 0 to 59 minutes) for the offset from UTC.

Summer Time Name The name of the configured summer time zone; this will be displayed during the summer time setting. Maximum 4 characters and minimum 3 characters (do not use angled brackets < >). If this parameter is not set, then the summertime feature will not work.

Summer Time Offset The offset from standard time in minutes. Valid values are 0 to 180.

Summer Time Mode You can configure the summer time to take effect:

- **None**—No summer time change.
- **Fixed**—The summer time change goes into effect at the specified time every year. For example, April 15 at 1:00 pm.
- **Recurring**—The summer time changes goes into effect every year at same relative time. For example, on the third week in April on a Tuesday at 1:00 pm.

Fixed Start Date Sets the exact date and time in which the Device Server's clock will change to summer time (daylight saving time) hours.

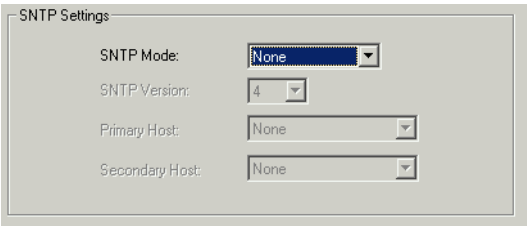
Fixed End Date Sets the exact date and time in which the Device Server's clock will end summer time hours and change to standard time.

Recurring Start Date Sets the relative date and time in which the Device Server's clock will change to summer time (daylight saving time) hours. Sunday is considered the first day of the week.

Recurring End Date Sets the relative date and time in which the Device Server's clock will end summer time hours and change to standard time. Sunday is considered the first day of the week.

Configuring SNTP Settings

You can configure an SNTP server that will synchronize the Device Server's internal clock with the SNTP time.



Configure the following options:

SNTP Mode The SNTP mode. Valid modes are:

- **None**—SNTP is turned off.
- **Unicast**—Sends a request packet periodically to the Primary host. If communication with the Primary host fails, the request will be sent to the Secondary host.
- **Multicast**—Listen for any broadcasts from an SNTP server and then synchronizes its internal clock to the message.
- **Anycast**—Sends a request packet as a broadcast on the LAN to get a response from any SNTP server. The first response that is received is used to synchronize its internal clock and then operates in **Unicast** mode with that SNTP server.

SNTP Version Version of SNTP. Valid values are 1 to 4. Default value is **4**.

Primary Host The name of the primary SNTP server from the Device Server host table. Valid with **Unicast** and **Multicast** modes, although in **Multicast** mode, the Device Server will only accept broadcasts from the specified host SNTP server.

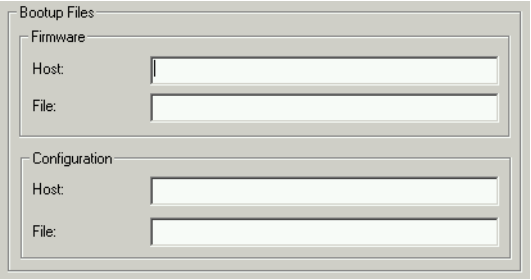
Secondary Host The name of the secondary SNTP server from the Device Server host table. Valid with **Unicast** and **Multicast** modes, although in **Multicast** mode, the Device Server will only accept broadcasts from the specified host SNTP server.

Configuring Administration Tasks

You can specify new configuration and firmware files that will go into effect the next time the Device Server is rebooted and a message of the day (MOTD) file, whose contents will be displayed when User's log into the Device Server.

Configuring Bootup Files

When you specify a configuration and/or firmware file(s), the files will be downloaded via TFTP to the Device Server the next time it is rebooted.



Configure the following parameters:

Firmware Host The host name or IP address of the server that contains the configuration or firmware file. If you use a host name, it must exist in the Device Server's host table or be resolved by DNS.

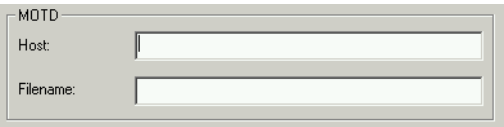
Firmware File The path and file name, relative to the default path of your TFTP server software, of the update software for the Device Server that will be loaded when the Device Server is rebooted.

Configuration Host The host name or IP address of the server that contains the configuration or firmware file. If you use a host name, it must exist in the Device Server's host table or be resolved by DNS.

Configuration File The path and file name, relative to the default path of your TFTP server software, of the configuration software for the Device Server that will be loaded when the Device Server is rebooted.

Configuring the MOTD File

You can specify a file whose content will be displayed to users after they connect to the Device Server, but before they log in. The Device Server will retrieve the file content every time a user connects to the Device Server, so you can change the content of the file without reconfiguring it within the Device Server.



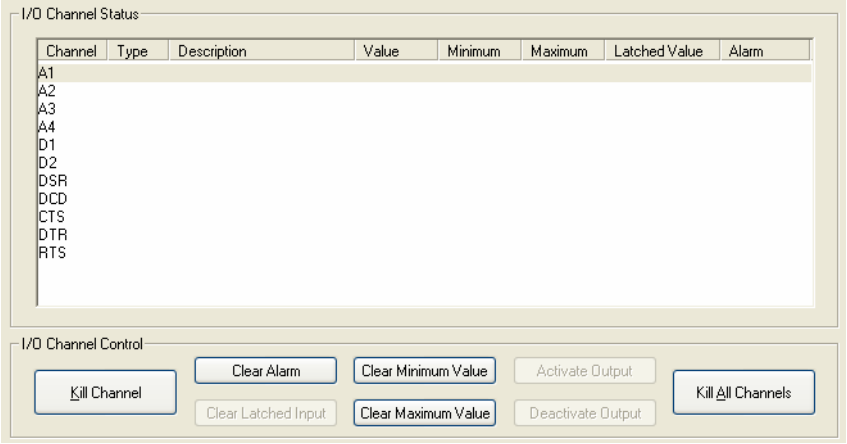
Configure the following parameters:

Host The host that the Device Server will be getting the Message of the Day file from.

Filename The path and file name, relative to the default path of your TFTP server software, of the file that contains a string that is displayed when a user connects to the Device Server.

I/O Status/Control

The I/O Status/Control window allow you to view I/O status and manually control I/O data.



A brief description of the control buttons follows:

Kill Channel	Resets the highlighted channel (click on a channel to highlight it).
Clear Alarm	Clears the alarm. Note that if the condition that tripped the alarm still exists, the alarm will not look like it's cleared, but will reflect the appropriate alarm level severity. Alarm Level 0 means that the alarm has not been triggered.
Clear Latched Input	Clears the latch value.
Clear Minimum Value	Clears the minimum value.
Clear Maximum Value	Clears the maximum value.
Activate Output	Manually activates the channel output.
Deactivate Output	Manually deactivates the channel output.
Kill All Channels	Resets all the channels.

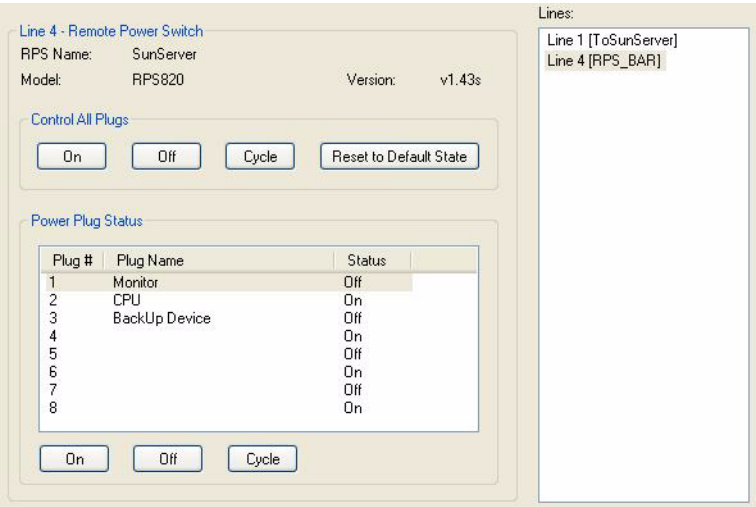
Power Management

When you are connected to a Device Server that has a serial connection to the Perle Remote Power Switch (RPS) and has the **Line Service** set to **Power Management** for that line or has plugs associated with a line(s), the Power Management control becomes available.

You can use the Power Management control to actively manage all power plugs associated with a line or individual plugs when the line is connected to the RPS.

Managing the RPS

When you select a line that is connected to the RPS unit, you can individually managing the RPS plugs.



The following information is displayed:

RPS Name	Displays the name of the RPS unit.
Model	Displays the RPS model type.
Version	Displays the software version on the RPS unit.

Control All Plugs

When you click on any of the buttons in this group, all the of RPS plugs will change to the specified state:

On	Turns all the RPS plugs on.
Off	Turns all the RPS plugs off.
Cycle	Turns all the RPS plugs off and then on.
Reset to Default	Resets all the RPS plugs to the default state as defined in the Power Management line settings.

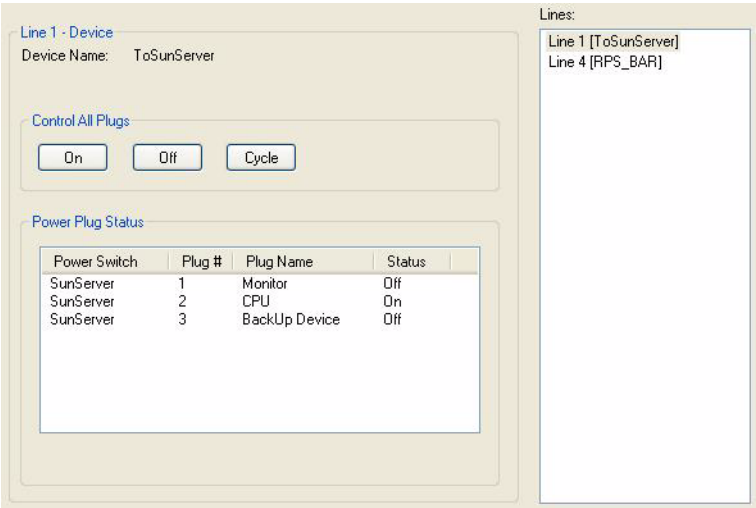
Control Individual Plugs

When you click on a Power Plug and then click on a button in this group, the individual Power Plug will change to the specified state:

On	Turns the selected plug on.
Off	Turns the selected plug off.
Cycle	Turns the selected plug off and then on.

Managing Plugs Associated with a Line

When you select a line that has power plugs associated with it, you can manage all the plugs as a group for that line.



The following information is displayed:

Device Name	Displays the name for the serial line.
Power Plug Status	Displays the plug status for every plug associated with the line.

When you click on a button in the Control All Device Plugs group, all the plugs displayed in the Power Plug Status list will change to the specified state:

On	Turns all the plugs in the Power Plug Status list on.
Off	Turns all the plugs in the Power Plug Status list off.
Cycle	Turns all the plugs in the Power Plug Status list off and then on.

Statistics

After you are connected to a Device Server, you can view statistics about the Device Server and its network environment. This can help you to troubleshoot problems or can provide valuable information about the Device Server's environment.

Tools

Saving a Configuration To File

When you connect to a Device Server, the Device Server's configuration is automatically uploaded to the DeviceManager. We suggest that you save the configuration to a file at this point, in case you need to revert to a working configuration in the future, by selecting **Tools**, **Save Configuration to File**. You can choose to save the configuration to the Device Server's native binary format or to a text file, which can be edited with a text editor. Either format can be reloaded into the DeviceManager at any time.

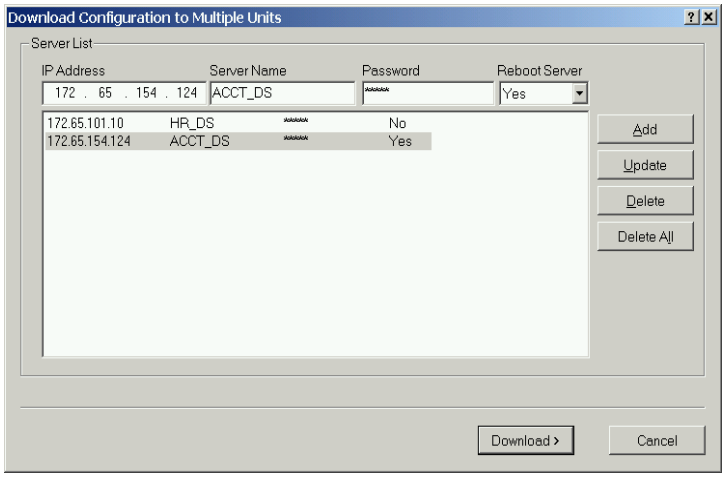
Getting a Configuration File

The DeviceManager can get a local configuration file (either binary or text) when you select **Tools**, **Get Configuration**, **Import from File**. The DeviceManager can also get the configuration from the Device Server it's connected to when you select **Tools**, **Get Configuration**, **Upload from Unit**; this can be useful if you've made changes to the Device Server's configuration that you would like to discard, you can simply reload the Device Server's current configuration into the DeviceManager.

Configuring Multiple Device Servers

You can configure multiple Device Servers at one time with the active configuration file. Any value in the configuration file's **Server Name** and **Internet Address** parameters will be overwritten by the values specified in the **Server Name** and **IP Address** fields (these fields cannot be left blank)

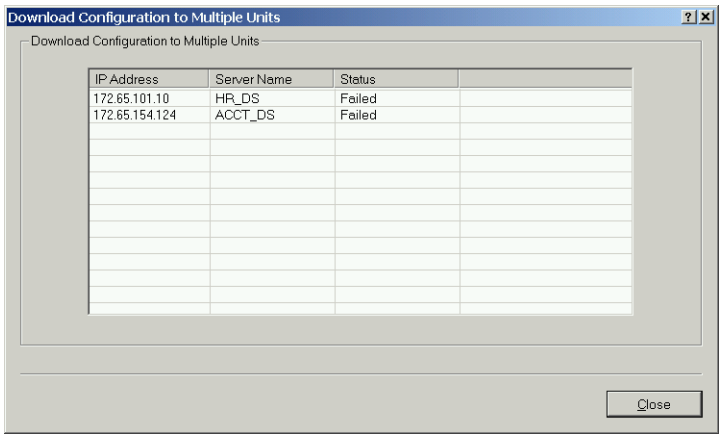
1. Select **Tools**, **Download Configuration to Multiple Units**. The Download Configuration to Multiple Units window is displayed.



2. Enter the following information for each Device Server that you want to configure with the same configuration file:

IP Address	Enter the IP address of the Device Server that you want to download the configuration to.
Server Name	The name of the Device Server. The Device Server name that you put in this field is passed into the configuration before it is downloaded to the Device Server and cannot be left blank.
Password	Enter the Admin user password for the Device Server.
Reboot Server	Determines whether or not the Device Server is rebooted after it has received the new configuration. The new configuration definitions will not go into effect until the Device Server is rebooted.

3. Click **Add** to add the Device Server to the download list. You can also click on a Device Server and edit any information and then click **Update** to make the edits permanent.
4. Click the **Download>** button to start the download process. A status window will display with the configuration download status.

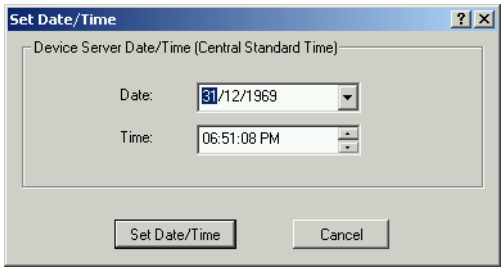


Downloading Device Server Firmware

To upgrade the Device Server firmware (software), select **Tools**, **Download Firmware to Unit**. Once the firmware download is complete, you will be prompted to reboot the Device Server. You can choose to reboot the Device Server at another time by selecting **Tools**, **Reboot Server**. Upgrading the firmware does not affect the Device Server's configuration file or downloaded custom files.

Setting the Device Server's Date and Time

To set the Device Server's system clock, select **Tools**, **Set Unit Time/Date**. The Set Date/Time window is displayed.



Configure the following parameters:

Date — The Device Server's date. The format of the Device Server's date is dependent on the Windows operating system and regional settings.

Time — The Device Server's internal clock time, based on your PC's time zone. For example, if your PC's time zone is set to Pacific Standard Time (GMT -8:00) and the Device Server's time zone is set to Eastern Standard Time (GMT -5:00), the Device Server's time is three hours ahead of your PC's time. If you set the Device Server's time to 2:30 pm, the Device Server's actual internal clock time is 5:30 pm.

Rebooting the Device Server

When you download any file (configuration, keys, certificates, firmware, etc.) to the Device Server, you must reboot the Device Server for it to take effect by selecting **Tools**, **Reboot Server**.

Resetting the Device Server to Factory Defaults

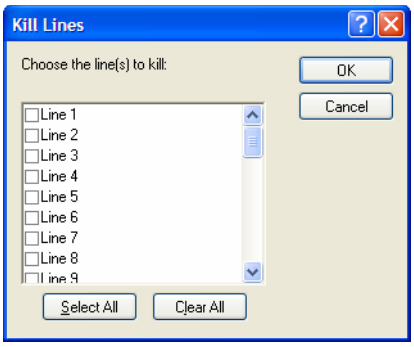
You can reset the Device Server to its factory default configuration by selecting **Tools**, **Reset to Factory Default**. The Device Server will automatically reboot itself with the factory default configuration.

Resetting the SecurID Node Secret

If you are using SecurID external authentication, you can select **Tools**, **Reset SecurID Node Secret** to reset the node secret. You do not need to reboot the Device Server for this to take effect, it works instantly.

Resetting/Killing a Line

After you make changes to the **Line** configuration parameters and click the **Apply** button, you can reset/kill the line to test the changes by selecting **Tools**, **Kill Line**. If you are connecting to a 1-port Device Server, you might be prompted to confirm to kill the line; if you are on a 2-port+ Device Server, the following window is displayed (shown for a SCS32):



Select the lines you want to reset/kill and then click **OK**. You can reset all the lines by clicking the **Select All** button and then clicking **OK**.

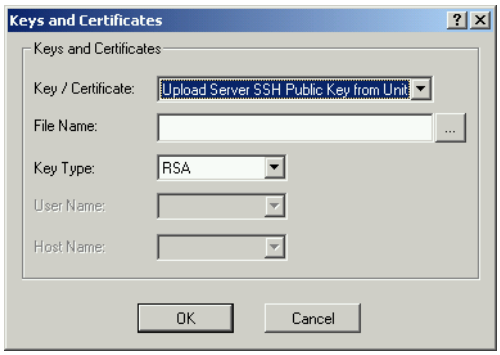
If you are happy with the configuration changes, you can download the configuration by selecting **Tools**, **Download Configuration to Unit**. Of course, your new configuration will not take effect until you reboot the Device Server by selecting **Tools**, **Reboot Server**.

Keys and Certificates

You will need to download/upload keys and/or certificates if you are using:

- **SSH**
- **LDAP** with **TLS**
- Secure HTTP (**HTTPS**)
- **SSL/TLS**

See ***Keys and Certificates*** on page 98 for information on when you need to download/upload keys and/or certificates.



Configure the following parameters:

Key / Certificate

Select the key or certificate that you want to download to the Device Server or upload the Device Server SSH Public Key.

- **Upload Server SSH Public Key from Unit**, used for reverse SSH connections
- **Download SSH User Public Key to Unit**, used for reverse SSH connections
- **Download SSH User Private Key to Unit**, used for Device Server Users with silent/direct SSH connections
- **Download SSH Host Public Key to Unit**, used for Device Server Users with silent/direct SSH connections
- **Download SSL/TLS Private Key to Unit**, required if using HTTPS and/or SSL/TLS
- **Download SSL/TLS Certificate to Unit**, required if using HTTPS and/or SSL/TLS
- **Download SSL/TLS CA to Unit**, required if using LDAP with TLS and/or SSL/TLS

File Name

The file that you are going to download/upload to/from the Device Server via TFTP.

Key Type

Specify the type of authentication that will be used for the SSH session. The following list details the keys that support each key type.

- ***RSA**—Server SSH Public Key, SSH User Public Key, SSH User Private Key, SSH Host Public Key
- **DSA**—Server SSH Public Key, SSH User Public Key, SSH User Private Key, SSH Host Public Key
- ****RSA1**—SSH User Private Key, SSH Host Public Key

***RSA** is used with SSH-2

****RSA1** is used with SSH-1

User Name	The name of the user for whom you are downloading the **SSH User Public** or **Private Key** to the Device Server.
Host Name	The name of the host for which you are downloading the **SSH Host Public** or **Private Key** to the Device Server.

Custom Files

Saving Crashes to a Dump File

If the Device Server should crash, you can save the crash information (dump) to a file that can be sent to Technical Support for interpretation. This should probably be done only under the guidance of Technical Support.

Downloading Terminal Definitions

You can create up to three custom terminal definitions and download them to the Device Server (if you need a terminal definition that is not currently defined within the Device Server). It is important that you remember which Device Server Terminal Definition you download your custom terminal definition under.

For example, if you download a custom terminal definition as **Terminal Definition 2**, you must select **Terminal Type Term2** in the **Line** window to use that terminal definition.

See ***Creating Terminal Definition Files*** on page 102 for information on creating custom terminal definitions.

Downloading a Language File

You can download one custom language file that can be specified in the **User** configuration window. See ***Language support*** on page 100 for information on creating custom language files.

Downloading a Custom App File

You can download each custom application file, created with the Perle SDK, to the Device Server using this option.

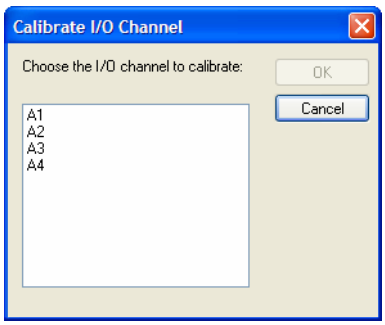
Downloading a Wireless WAN Driver

You can download a custom wireless WAN driver to the Device Server using this option.

I/O Channels

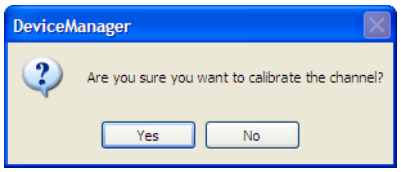
Calibrating Analog Channels

Analog Input can be calibrated for Analog and Temperature Device Server models. To calibrate either of these models, select **Tools**, **I/O Channels**, **Calibrate**.

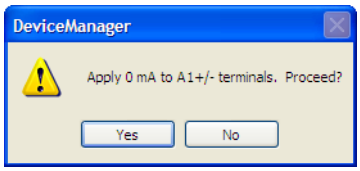


Select the channel you want to calibrate. This example uses an A4 model that has channel A1 set to Current with a Range of 0 to 20mA.

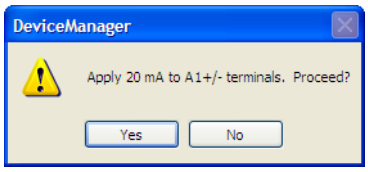
If you have not disabled confirmation messages (**Tools**, **Options**), you will get prompted to verify channel calibration.



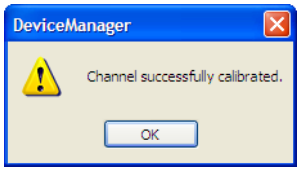
Click **Yes** to proceed with calibration. You are now prompted to apply 0 mA to the positive (+) and negative (-) terminals. Once that is done, click **Yes** to proceed.



You are now prompted to apply 20 mA to the positive (+) and negative (-) terminals. Once that is done, click **Yes** to proceed.



Once calibration is successfully completed, click **OK** to finish the process.



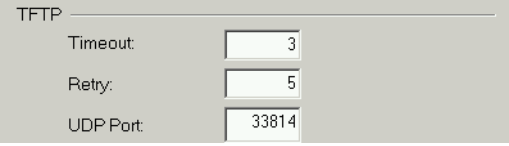
Resetting Calibration Data

When you select **Tools**, **I/O Channels**, **Reset Calibration Data**, you are resetting the calibration values to the factory default values.

Setting DeviceManager Options

When you select **Tools**, **Options**, you can set the following:

- **Confirmation Messages**—Specify whether you want to receive confirmation messages for all of the following selections:
  - **Tools**, **Download Configuration to Unit**
  - **Tools**, **Reboot Server**
  - **Tools**, **Reset to Factory Defaults**
  - **Tools**, **Reset SecurID Node Secret**
  - **Tools**, **Kill Line(s)**
  - Anytime you click a **Delete** button
- **TFTP**—Sets the TFTP options for communication between the DeviceManager and a Device Server.



Configure the following parameters:

Retry — The number of times the Device Server will retry to transmit a TPFT packet to/from a host when no response is received. Enter a value between 0 and 5. The default is **5**. A value of **0** (zero) means that the Device Server will not attempt a retry should TFTP fail.

Timeout — The time, in seconds, that the DeviceManager will wait for a successful transmit or receipt of TFTP packets before retrying a TFTP transfer. Enter a value between 3 and 10. The default is **3** seconds.

UDP Port — The port that the DeviceManager will use to TFTP to the Device Server. The default port is 33814 (ports 33812 and 33813 are also in use by the DeviceManager).

- **Statistics**—Specify whether or not you want to have the statistics automatically refresh and the refresh rate.

6 WebManager and EasyPort Web

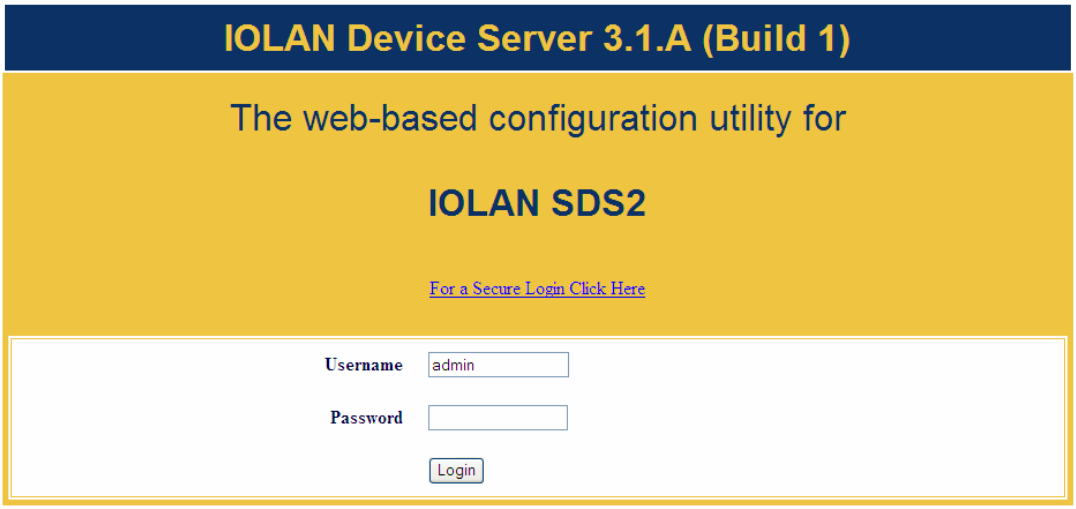
Introduction

This chapter provides general information about using the WebManager, a web browser configuration and management utility, and EasyPort Web, which provides web browser access to Device Servers in a cluster group, power management (when using the Perle Remote Power Switch), and Reverse Telnet/SSH sessions.

Using WebManager

Logging into WebManager

To log into WebManager, simply type in the Device Server's IP address into the Address field of your web browser; for example: **`123.123.123.123`**. The WebManager Login screen will be displayed:

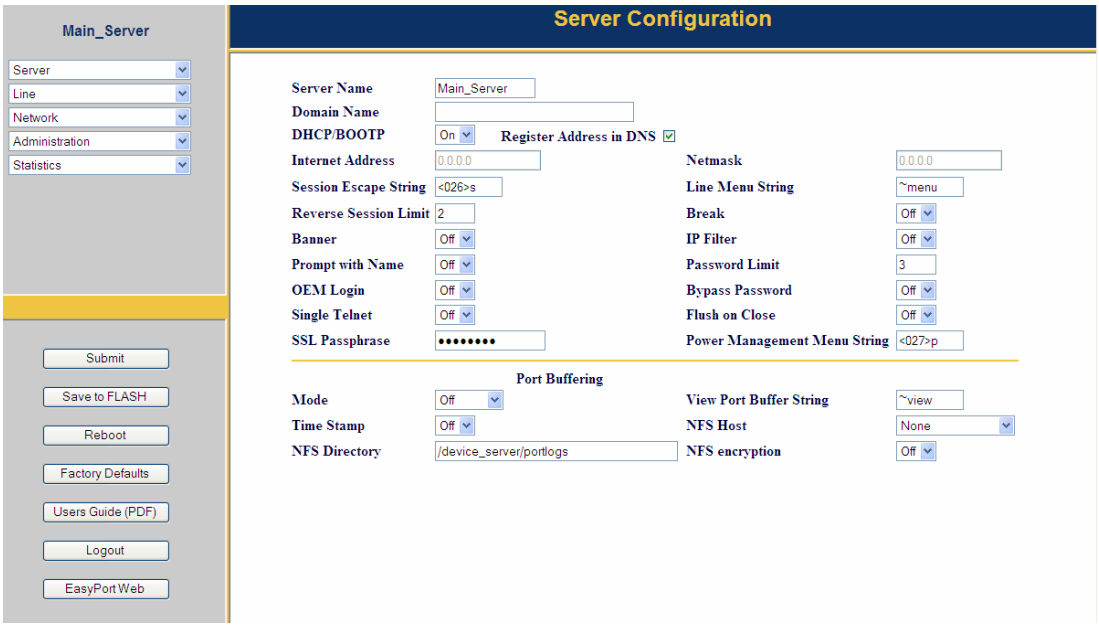


The Login screen displays the Device Server's firmware version and model. Type in your **Username** and **Password**. If you are accessing the Device Server as the admin user or as a user who has **Admin** level access rights, you will get the WebManager configuration/management screen. If you are a user with **Normal**, **Menu**, or **Restricted** level access rights, you will automatically see EasyPort Web; see ***EasyPort Web*** on page 241 for more information about this utility.

Configuring the Device Server Using WebManager

If you are the admin user or a user with **Admin** level rights, you can configure the Device Server through WebManager. Unlike using the DeviceManager configuration/management application, there is no way to save the configuration locally, as you are connected live to the Device Server when you are using WebManager.

The Server Configuration window is launched when you log on, displaying the configuration running in the Device Server.



You navigate through the different configuration windows by selecting the configuration window from the drop-down options in the upper-left hand corner of the browser; for example, from the **Line** drop-down button you can configure the line in **Port Settings**.

When you have completed all the changes to a configuration window, click the **Submit** button. After you make all your configuration changes, click the **Save to FLASH** button. If you want your changes to take effect immediately, click the **Reboot** button. You can make changes to a line, **Submit** them, and then click the **Kill Line** button to test the changes immediately; however, if you do not click the **Save to FLASH** button, your changes will be lost the next time the Device Server reboots. After you click the **Reboot** button, you will need to reconnect and login to the Device Server.

Note: Use the WebManager's drop-down menus to navigate through the WebManager. Do not use the browser's Back button.

EasyPort Web

EasyPort Web is available to all users (unless the HTTPD or HTTPSD service is disabled for the Device Server). Users with **Admin** level access rights can launch EasyPort Web by clicking the **EasyPort Web** button in WebManager. All other users will automatically see EasyPort Web when they access a Device Server by entering the Device Server's IP address in a web browser.

EasyPort Web Configuration Requirements

The following configuration requirements must be met when using EasyPort Web:

- The Device Server that is being accessed through a **Reverse SSH** port via EasyPort Web must have the **SSH Server** parameter, **Allow SSH-1 Protocol**, enabled.
- The computer running EasyPort Web must have the Java runtime environment installed (Java 1.42 or later) to access **Reverse Telnet** and/or **Reverse SSH** ports. You can download the latest Java runtime environment from www.java.com.

Reverse Session Users

When a user is configured, the user is assigned an access **Level** to the Device Server. Users with **Normal**, **Menu**, or **Restricted Level** access rights can launch EasyPort Web by opening a web browser and typing the IP address of the Device Server in the web browser's **Address** field. The user will need to type in a **Username** and **Password**. The user will then be able to access any of the lines that have been configured with **Reverse Telnet** or **Reverse SSH Service** line settings. In the example below, only line 2 has been configured for **Rev SSH**, so that is all the user can access.

The user can click the **SSH** button and a java applet is launched to make an SSH connection to the server.

Power Management

When a user connects to the Device Server through EasyPort Web and there is a Perle Remote Power Switch (RPS) either connected to the Device Server or there is a line(s) associated with an RPS plug(s), the user can manage the power to the plugs. See ***Power Management*** on page 128 for a more complete explanation and example.

Clustered Device Servers

When a user connects to a Master Device Server through EasyPort Web, the user will be able to access all the Slave Device Servers that have **Reverse Telnet**/**SSH** defined lines. Users can then either connect directly to a specific Slave Device Server's line or to a Slave Device Server unit. See ***Clustering*** on page 122 for a more complete explanation and example.

7 Command Line Interface

Introduction

This chapter provides the command line interface (CLI) options available for the Device Server. The commands are grouped by function.

CLI Conventions

This section explains how to interpret the CLI syntax. If you are an existing IOLAN+ customer and would like to configure the Device Server in the native IOLAN+ interface, you can type the command **`iolan+`** to display and use the native IOLAN+ interface (you must have **User Level Admin**). See your ***IOLAN+ User Guide*** for information on using the IOLAN+ interface.

Command Syntax

Each command is broken down into several categories:

- **Description**—Provides a brief explanation of how the command is used.
- **User Level**—Shows which user level(s) (Restricted, Normal, and/or Admin) can issue the command. Some commands have options that are available for one user level and not for another level; this usually occurs when a command is valid for both Normal and Admin user levels, where the Admin user level command will have extended options.
- **Syntax**—Shows the actual command line options. The options can be typed in any order on the command line. The syntax explanation will use the following command to break down the command syntax:

  ```
  set service [dhcp/bootp on|off] [telnetd on|off] [httpd on|off]
  [snmpd on|off] [spcd on|off] [syslog on|off] [dmgrd on|off]
  ```

  - Square brackets ([]) show the options that are available for the command. You can type a command with each option individually, or string options together in any order you want. For example,

    `set service dhcp/bootp on telnetd off`

  - Angle brackets (<>) show that the text inside the brackets is a description for a variable value that you must fill in according to your requirements. In the **`set server`** command, you must determine the values for **`domain`**, **`internet`**, **`name`**, **`password-limit`**, and **`subnet-bit-length`**, if you wish to specify them and not use their defaults (default values provided in the **Options** description). The angle brackets can also contain a range that can be used.
  - The pipe (|) shows an 'or' condition. For example, valid values for **`telnetd`** are either **`on`** or **`off`**.
- **Options**—Provides an explanation of each of the options for a command and the default value if there is one. Some commands do not have any options, so this category is absent.

Command Shortcuts

When you type a command, you can specify the shortest unique version of that command or you can press the **ESC** or **TAB** key to complete the command. For example, the following command:

```
set telnet-client map-to-crlf off
```

can be typed as:

```
set tel map off
```

or, you can use the **ESC** key to complete the lines as you go along:

```
set tel<ESC>net-client ma<ESC>p-to-crlf off
```

where the **ESC** key was pressed to complete the option as it was typed.

Command Options

When you are typing commands on the command line (while connected to the Device Server), you can view the options by typing a question mark (**?**), **ESC**, or **TAB** key after any part of the command to see what options are available/valid. For example:

```
DS$ set vmodem ?
failure-string
host
port
style
success-string
suppress
DS$ set vmodem failure-string ?
<text>                  30 characters maximum
DS$ set vmodem failure-string "Vmodem failed" ?
failure-string
host
port
style
success-string
suppress
Or press Enter to confirm command
DS$ set vmodem failure-string "Vmodem failed"
DS$ show vmodem
Host
Host Port
Success String
Failure String          "Vmodem failed"
Suppress                Off
Style                   Numeric
DS$
```

Server Commands

This section defines all the CLI commands associated with configuring the Device Server's server parameters.

Server Commands

Set Console

Description Sets the flow control and baud rate on Device Server models that have a dedicated console port.

User Level Admin

Syntax

```
set console [flow none|soft|hard]
 [speed 9600|19200|38400|57600|115200]
```

Options

flow

For Device Server models that have a dedicated console port, defines whether the data flow is handled by using software (**Soft**), hardware (**Hard**), or no (**None**) flow control.

speed

For Device Server models that have a dedicated console port, specifies the baud rate of the line connected to the console port.

Set Custom-App

Description You can create a custom application that can run on the Device Server using the Perle SDK.

User Level Admin

Syntax

```
set custom-app server program-command-line <command>
```

Options

program-command-line

The name of the SDK program executable that has been already been downloaded to the Device Server, plus any parameters you want to pass to the program. Maximum of 80 characters. Use the **`shell`** CLI command as described in the *SDK Programmer's Guide* to manage the files that you have downloaded to the Device Server. For example, using sample **`outraw`** program, you would type:

```
outraw -s 0 192.168.2.1:10001 Acct:10001
```

if you were starting the application on the Server (notice the **`-s 0`** parameter specifies Line 1).

Set Port-Buffering

Description Configures port buffering.

User Level Admin

Syntax

```
set port-buffering [mode off|local|remote|both]
 [nfs-directory <text>] [nfs-encryption on|off]
 [nfs-host <config_host>] [time-stamp on|off]
 [view-port-buffer-string <text>]
```

Options

mode

Specifies where the port buffer log is kept, either **Off**, **Local**, **Remote**, or **Both**. If **Remote** or **Both** is selected, you must specify an NFS server location for the port buffer log.

nfs-directory

The directory and/or subdirectories where the **Remote Port Buffering** files will be created. This field is used when Port Buffering **Mode** is set to **Remote** or **Both**. For multiple Device Servers using the same NFS host, it is recommended that each Device Server have its own unique directory to house the remote port log files. The default is **`/device_server/portlogs`**.

nfs-encryption

Determines if the data sent to the NFS host is sent encrypted or in the clear across the LAN. The default is set of **Off**.

NOTE: When NFS encryption is enabled, the Decoder utility software is required to be installed on the NFS host for decrypting the data to a readable format. The Decoder utility software can be found on the installation CD-ROM and on the **`www.perle.com`** website.

nfs-host

The NFS host that the Device Server will use for its **Remote Port Buffering** feature. The Device Server will open a file on the NFS host for each reverse SSH or reverse Telnet line, and send any port data to be written to those files. The default is **None**. This field is required when **Mode** is set to **Remote** or **Both**.

time-stamp

Enable/disable time stamping of the port buffer data.

view-port-buffer-string

The string (up to 8 characters) used by a a session connected to a serial port to display the port buffer for that particular serial port. You can specify control (unprintable) codes by putting the decimal value in angle brackets < > (for example, **Escape b** is **<027>b**). The default is **~view**.

Set Server

Description Sets server parameters.

User Level Admin

Syntax

```
set server [auto-obtain-dns on|off] [auto-obtain-gw on|off]
 [auto-obtain-wins on|off] [banner on|off] [break on|off]
 [bypass-password on|off] [dhcp-update-dns on|off]
 [dhcp/bootp on|off] [domain <string>] [internet <IPV4_address>]
 [flush-on-close on|off] [line-menu-string <string>]
 [monitor-connection-every <1-32767>] [name <string>]
 [netmask <IPV4_address>][oem-login on|off]
 [password-limit <0-10>] [prompt-with-name on|off]
 [ip-filter on|off] [session-escape-string <string>]
 [single-telnet on|off] [monitor-connection-every <seconds>]
 [active-standby on|off] [miimon <milliseconds>]
 [updelay <milliseconds>] [power-management-menu-string <string>]

set server internet [eth1|eth2] <IPV4_address> [netmask]

set server internet [eth1|eth2] dhcp/bootp on dhcp-update-dns on
 domain-prefix <text>

set server internet [eth1|eth2] dhcp/bootp on dhcp-update-dns off

set server internet [eth1|eth2] dhcp/bootp off <IPV4_address>
 [<netmask>]

set server tftp [retry <integer>] [timeout <integer>]

set server ssl-passphrase
```

Options

auto-obtain-dns

When DHCP/BOOTP is enabled, you can enable this option to have the Device Server receive the DNS IP address from the DHCP/BOOTP server.

auto-obtain-gw

When DHCP/BOOTP is enabled, you can enable this option to have the Device Server receive the Default Gateway IP address from the DHCP/BOOTP server.

auto-obtain-wins

When DHCP/BOOTP is enabled, you can enable this option to have the Device Server receive the WINS IP address from the DHCP/BOOTP server.

banner

This parameter concerns the banner information (product name/software version). This banner information is presented to a user with a login prompt. For security reasons, you can turn off the display of this information. The default is **Off**.

break

Enables/disables proprietary inband SSH break signal processing as well as the existing Reverse Telnet break signal. This parameter can also enable/disable the out-of-band break signals for TruePort. The default value is **Off**.

bypass-password

When set, authorised users who do not have a password set, with the exception of the Admin user, WILL NOT be prompted for a password at login with **Local Authentication**.

dhcp-update-dns

The DHCP server will update the DNS server when the Device Server requests a DHCP IP address (the communication between the DNS server and the DHCP server must already be set up in your network).

dhcp/bootp

Enables the DHCP/BOOTP client process in the Device Server. By default, this is disabled/off. If this is enabled, the server IP address parameter is disabled.

domain

Unique name for your domain, your location in the global network. Like Hostname, it is a symbolic, rather than a numerical, identifier.

domain-prefix

(SCS models only) A domain prefix to uniquely identify the Ethernet interface to the DNS when the Device Server has two Ethernet interfaces. The format of the Ethernet interface will take the form of <*Server Name*>.<*Domain Prefix*>.<*Domain Name*> or <*Server Name*>.<*Domain Prefix*>, depending on what is configured.

flush-on-close

When enabled, deletes any pending data when a port is closed; as opposed to maintaining the port to send pending data. The default value is **Off**.

internet

The Device Server's unique IPv4 network IP address. If you are using the Device Server in an IPv6 network, this field can be left blank.

internet [eth1|eth2]

Dual Ethernet SCS models require that you specify which Ethernet connection you are setting, either **eth1** or **eth2**.

name

You must supply a name for the Device Server.

netmask

The network subnet mask. For example, 255.255.0.0.

line-menu-string

The string used to access to the Easy Port Access menu without disconnecting the initial reverse SSH or reverse Telnet session.The default string is **~menu**.

monitor-connection-every

Specify how often, in seconds, the Device Server will send a TCP Keepalive to services that support TCP Keepalive. The default is 30 seconds.

oem-login

When set, and a custom language file is in use, the login prompt will use the string defined in the language file as the login prompt instead of the default prompt, **`login:`**.

password-limit

The number of authentication attempts a user is allowed for a serial port connection (this applies to **Line Service DSLogin** and Console mode connections). If this limit is exceeded, the port is disabled for 5 minutes. A user with Admin level rights can restart the port, bypassing the timeout, by issuing a kill on the disabled port. The default value is **3**.

prompt-with-name

Displays the **Server Name** field value instead of default product name. When enabled, the **Server Name** is displayed in the Device Server login prompt, CLI prompt, WebManager login screen, and the heading of the Menu. The default value is **Off**.

ip-filter

A security feature that when enabled, the Device Server will only accept data from hosts configured in the Device Server's **Host Table** with an IP address (hosts configured with a Fully Qualified Domain Name, FQDN, will not be able to access the Device Server when this option is enabled). The default value is **Off**.

single-telnet

Sets all reverse connections (raw, SSH, and telnet) to a one connection at a time mode. Server-side applications will get a (socket) connection refused until:

- All data from previous connections on that serial port has drained
- There are no other connections
- Up to a 1 second interconnection poll timer has expired

This also enables a per-connection keepalive TCP keepalive feature. After approximately 3 minutes of network connection idle time, the connection will send a gratuitous ACK to the network peer, thus either ensuring the connection stays active OR causing a dropped connection condition to be recognised by the reverse service (all connections).

Applications using Single Telnet need to be aware that there can be some considerable delay between a network disconnection and the port being available for the next connection attempt; this is to allow any data sent on prior connections to be transmitted out of the serial port. Application network retry logic needs to accommodate this feature. The default value is **Off**.

active-standby

(SCS only) Enables/disables the feature of automatically assigning the Ethernet 1 IP address to Ethernet 2 if Ethernet 1 should fail to communicate to the network.

miimon

(SCS only) The interval in which the active interface is checked to see if it is still communicating. The default is 100 ms.

updelay

(SCS only) The time that the Device Server will wait to make the secondary interface (Ethernet 2) active after it has been detected as up.

power-management-menu-string

Users accessing the Device Server through reverse sessions can enter the string to bring up a power bar management menu. This is a decimal value. The default value is **<016>** or **Ctrl-p** on the keyboard.

session-escape-string

A configurable string that allows access to a port to view the multisession screen options, allowing the various options while accessing the particular port on the Device Server. You can specify control (unprintable) codes by putting the decimal value in angle brackets < > (for example, **ESC-b** is **<027>b**). The default value is **Ctrl-z s** (**<026>s** in decimal).

retry

The number of times the Device Server will retry to transmit a TPFT packet to/from a host when no response is received. Enter a value between 0 and 5. The default is **5**. A value of **0** (zero) means that the Device Server will not attempt a retry should TFTP fail.

timeout

The time, in seconds, that the Device Server will wait for a successful transmit or receipt of TFTP packets before retrying a TFTP transfer. Enter a value between 3 and 10. The default is **3** seconds.

ssl-passphrase

This is the SSL/TLS passphrase used to generate an encrypted RSA/DSA private key. This private key and passphrase are required for both HTTPS and SSL/TLS connections, unless an unencrpyted private key was generated, then the SSL passphrase is not required. Make sure that you download the SSL private key and certificate if you are using the secure HTTP option (HTTPS) or SSL/TLS. If both RSA and DSA private keys are downloaded to the Device Server, they need to be generated using the same SSL passphrase for both to work.

Set SSL Server

Description Sets the default SSL/TLS parameters for the server.

User Level Admin

Syntax

```
set ssl server [version any|tslv1|sslv3] [type client|server]
 [verify-peer on|off]
 [validation-criteria
   country <code>|state-province <text>|locality <text>
   |organisation <text>|organisation-unit <text>
   |common-name <text>|email <email_addr>]
```

Options

version

Specify whether you want to use:

- **Any**—The Device Server will try a TLSv1 connection first. If that fails, it will try an SSLv3 connection. If that fails, it will try an SSLv2 connection.
- **TLSv1**—The connection will use only TLSv1.
- **SSLv3**—The connection will use only SSLv3.

The default is **Any**.

type

Specify whether the Device Server will act as an SSL/TLS client or server. The default is **Client**.

verify-peer

Enable this option when you want the Validation Criteria to match the Peer Certificate for authentication to pass. If you enable this option, you need to download an SSL/TLS certificate authority (CA) list file to the Device Server.

validation-criteria

Any values that are entered in the validation criteria must match the peer certificate for an SSL connection; any fields left blank will not be validated against the peer certificate.

country

A two character country code; for example, US. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

state-province

Up to a 128 character entry for the state/province; for example, IL. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

locality

Up to a 128 character entry for the location; for example, a city. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

organisation

Up to a 64 character entry for the organisation; for example, Accounting. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

organisation-unit

Up to a 64 character entry for the unit in the organisation; for example, Payroll. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

common-name

Up to a 64 character entry for common name; for example, the host name or fully qualified domain name. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

email

Up to a 64 character entry for an email address; for example, acct@anycompany.com. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Set Service

Description Sets server service parameters.

User Level Admin

Syntax

```
set service [routed on|off] [telnetd on|off] [sshd on|off]
 [httpd on|off] [snmpd on|off] [spcd on|off] [sntp on|off]
 [httpsd on|off] [syslog on|off] [dmgrd on|off] [modbusd on|off]
```

Options

routed

Route daemon process in the Device Server on port 520.

telnetd

Telnet daemon process in the Device Server on port 23.

sshd

SSH daemon process in the Device Server on port 22.

httpd

HTTP daemon process in the Device Server on port 80.

snmpd

SNMP daemon process in the Device Server on port 161.

spcd

SPC (TruePort) daemon process in the Device Server that supports TruePort Full Mode on UDP port 668. You can still communicate with the Device Server in Light Mode when this service is disabled.

sntp

SNTP client process in the Device Server.

httpsd

HTTPS daemon process in the Device Server on port 443.

syslog

Syslog client process in the Device Server.

dmgrd

DeviceManager daemon process in the Device Server. If you disable this service, you will not be able to connect to the Device Server with the DeviceManager application. DeviceManagerD listens on port 33812 and sends on port 33813.

modbusd

Modbus daemon process in the Device Server on port 502.

Show Console

Description For Device Server models that have a dedicated console port, shows the set parameter values.
User Level Admin
Syntax `show console`

Show Custom-App

Description Shows the custom application server settings.
User Level Admin
Syntax `show custom-app server`

Show Server

Description Shows the parameters set for the server.
User Level Admin, Normal
Syntax `show server`

Show Port-Buffering

Description Shows the port buffering settings.
User Level Normal, Admin
Syntax `show port-buffering`

Show Modbus

Description Shows the Modbus settings for the gateway.
User Level Normal, Admin
Syntax `show modbus gateway`

Hardware Commands

Set Ethernet

Description Sets the serial line speed and duplex.
User Level Admin
Syntax

```
set ethernet [eth1|eth2] speed-and-duplex
 auto|10-half|10-full|100-half|100-full|1000-half|1000-full
```

Options **eth1|eth2**

You must specify the Ethernet interface if you have an SCS model with dual Ethernet.

auto|10-half|10-full|100-half|100-full|1000-half|1000-full

Define the Ethernet connection speed at one of the following (desktop models don't support 1000 Mbps):

- **auto**—automatically detects the Ethernet interface speed and duplex
- **10 Mbps Half Duplex**
- **10 Mbps Full Duplex**
- **100 Mbps Half Duplex**
- **100 Mbps Full Duplex**
- **1000 Mbps Half Duplex**
- **1000 Mbps Full Duplex**

Show Hardware

Description Shows the hardware settings/information.
User Level Normal, Admin
Syntax

```
show hardware
```

SSH Server Commands

Set SSH-Server

See ***Keys and Certificates*** on page 98 for information about the keys and certificates that need to be uploaded or downloaded with the Device Server's SSH server.

Description Configures the Device Server's SSH server.
User Level Admin
Syntax

```
set ssh-server [authentication rsa on|off]
 [authentication dsa on|off] [authentication password on|off]
 [authentication keyboard-interactive on|off]
 [break-string <text>] [compression on|off] [ssh1 on|off]
 [verbose on|off]

set ssh-server cipher [3des on|off] [blowfish on|off]
 [cast on|off] [aes on|off] [arcfour on|off]
```

Options **authentication rsa**

An authentication method used by SSH version 1 and 2. Use RSA authentication for the SSH session.

authentication dsa

An authentication method used by SSH version 2. Use DSA authentication for the SSH session.

authentication password

The user types in a password for authentication.

authentication keyboard-interactive

The user types in a password for authentication.

compression

Requests compression of all data. Compression is desirable on modem lines and other slow connections, but will only slow down things on fast networks.

verbose

Displays debug messages on the terminal.

break-string

The break string used for inband SSH break signal processing. A break signal is generated on a specific serial port only when the server's break option is enabled and the user currently connected using reverse SSH has typed the break string exactly. The default is set to **~break**, where **~** is tilde; the break string can be up to eight characters.

ssh1

Allows the user's client to negotiate an SSH-1 connection, in addition to SSH-2.

cipher

Specify that ciphers that the Device Server's SSH server can use to negotiate data encryption with an SSH client session.

Show SSH-Server

Description Shows the SSH server settings.
User Level Admin
Syntax `show ssh-server`

SSL/TLS Commands

Set SSL Server

Description Sets the default SSL/TLS parameters for the server.
User Level Admin
Syntax

```
set ssl server [version any|tslv1|sslv3] [type client|server]
 [verify-peer on|off]
 [validation-criteria
   country <code>|state-province <text>|locality <text>
   |organisation <text>|organisation-unit <text>
   |common-name <text>|email <email_addr>]
```

Options **version**

Specify whether you want to use:

- **Any**—The Device Server will try a TLSv1 connection first. If that fails, it will try an SSLv3 connection. If that fails, it will try an SSLv2 connection.
- **TLSv1**—The connection will use only TLSv1.
- **SSLv3**—The connection will use only SSLv3.

The default is **Any**.

type

Specify whether the Device Server will act as an SSL/TLS client or server. The default is **Client**.

verify-peer

Enable this option when you want the Validation Criteria to match the Peer Certificate for authentication to pass. If you enable this option, you need to download an SSL/TLS certificate authority (CA) list file to the Device Server.

validation-criteria

Any values that are entered in the validation criteria must match the peer certificate for an SSL connection; any fields left blank will not be validated against the peer certificate.

country

A two character country code; for example, US. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

state-province

Up to a 128 character entry for the state/province; for example, IL. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

locality

Up to a 128 character entry for the location; for example, a city. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

organisation

Up to a 64 character entry for the organisation; for example, Accounting. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

organisation-unit

Up to a 64 character entry for the unit in the organisation; for example, Payroll. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

common-name

Up to a 64 character entry for common name; for example, the host name or fully qualified domain name. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

email

Up to a 64 character entry for an email address; for example, acct@anycompany.com. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Set SSL Server Cipher-suite

Description Sets the default SSL/TLS cipher suite parameters.

User Level Admin

Syntax

```
set ssl server cipher-suite
 option1|option2|option3|option4|option5
 encryption any|aes|3des|des|arcfour|arctwo|none
 min-key-size 40|56|64|128|168|256
 max-key-size 40|56|64|128|168|256
 key-exchange any|rsa|edh-rsa|edh-dss|adh
 hmac any|sha1|md5
```

Options **option1|option2|option3|option4|option5**

Sets the priority of the cipher suite, with **`option1`** being highest priority and **`option5`** lowest priority.

encryption

Select the type of encryption that will be used for the SSL connection:

- Any—Will use the first encryption format that can be negotiated.
- AES
- 3DES
- DES
- ARCFOUR
- ARCTWO
- None—Removes any values defined for the cipher option.

The default value is **Any**.

min-key-size

The minimum key size value that will be used for the specified encryption type. The default is **40**.

max-key-size

The maximum key size value that will be used for the specified encryption type. The default is **256**.

key-exchange

The type of key to exchange for the encryption format:

- **Any**—Any key exchange that is valid is used (this does not, however, include ADH keys).
- **RSA**—This is an RSA key exchange using an RSA key and certificate.
- **EDH-RSA**—This is an EDH key exchange using an RSA key and certificate.
- **EDH-DSS**—This is an EDH key exchange using a DSA key and certificate.
- **ADH**—This is an anonymous key exchange which does not require a private key or certificate. Choose this key if you do not want to authenticate the peer device, but you want the data encrypted on the SSL/TLS connection.

The default is **Any**.

hmac

Select the key-hashing for message authentication method for your encryption type:

- Any
- MD5
- SHA1

The default is **Any**.

Show SSL

Description Shows the SSL/TLS settings/information.
User Level Normal, Admin
Syntax `show ssl`

Modbus Commands

Set Modbus Gateway

Description Sets the authentication method for the Device Server.
User Level Admin
Syntax

```
set modbus gateway [addr-mode embedded|re-mapped]
  [broadcast on|off] [char-timeout <number>]
  [req-next-delay <number>] [exceptions on|off]
  [idle-timer <number>] [mess-timeout <number>]
  [port <TCP/UDP_port>] [req-queuing on|off]
  [remapped-id <1-247>] [ssl on|off]
```

Options

addr-mode

Determines if the original UID address will be embedded in the transmission header or if a specified (remapped) UID will be embedded in the transmission header.

broadcast

When enabled, a UID of 0 (zero) indicates that the message will be broadcast to all Modbus Slaves. The default is **Off**.

char-timeout

Used in conjunction with the Modbus RTU protocol, specifies how long to wait, in milliseconds, after a character to determine the end of frame. The default is **30** ms.

req-next-delay

A delay, in milliseconds, to allow serial slave(s) to re-enable receivers before issuing next Modbus Master request. The default is **50** ms.

exceptions

When enabled, an exception message is generated and sent to the initiating Modbus device when any of the following conditions are encountered: there is an invalid UID, the UID is not configured in the Gateway, there is no free network connection, there is an invalid message, or the target device is not answering the connection attempt. The default is **On**.

idle-timer

Specifies the number of seconds that must elapse without any network or serial traffic before a connection is dropped. If this parameter is set to 0 (zero), a connection will not be dropped (with the following exceptions: the TCP KeepAlive causes the connection to be dropped or the Modbus device drops the connection). The default is **10** seconds.

mess-timeout

Time to wait, in milliseconds, for a response message from a Modbus TCP or serial slave (depending if the Modbus Gateway is a Master Gateway or Slave Gateway, respectively) before sending a Modbus exception. The default is **1000** ms.

port

The network port number that the Slave Gateway will listen on for both TCP and UDP messages. The default is **502**.

req-queuing

When enabled, allows multiple, simultaneous messages to be queued and processed in order of reception. The default is **On**.

remapped-id

Specify the UID that will be inserted into the message header for the Slave Modbus serial device. Valid values are 1-247.

ssl

When enabled, Modbus Slave Gateway messages to remote TCP Modbus Masters are encrypted via SSL/TLS.

Show Modbus

Description Sets the authentication method for the Device Server.
User Level Admin
Syntax

```
show modbus gateway

show modbus slave|master <line_number>
```

Authentication Commands

Set Authentication

Description Sets the authentication method for the Device Server.
User Level Admin
Syntax

```
set authentication type primary|secondary
 none|local|radius|kerberos|ldap|tacacs+|securid|nis
 [secondary-as-backup on|off]
```

Options

primary

The first authentication method that the Device Server attempts. Some type of authentication must be done by the Device Server, therefore, **None** is not a valid option for the **Primary Authentication Method**.

secondary

If the **Primary Authentication Method** fails, the next authentication method that the Device Server attempts. You can choose to use authentication methods in combination. For example, you can specify the **Primary Authentication Method** as **Local** and the **Secondary Authentication Method** as **RADIUS**. Therefore, some users can be defined in the Device Server (**Local**) others in **RADIUS**.

none|local|radius|kerberos|ldap|tacacs+|securid|nis

Specify the authentication method that the Device Server will use to authenticate users (this must already be set up in your network).
secondary-as-backup

When enabled, the Secondary Authentication method server will be tried only when the Device Server cannot communicate with the Primary Authentication method server.

Set Authentication Local

Description Configures local authentication settings. When you configure the Device Server to authenticate user's locally, you can require that the user's be configured in the User table or you can allow Guest users, who can log into the Device Server using any ID, but must know the configured password.

User Level Admin

Syntax

```
set authentication local [guest-mode on|off] [password <text>]
```

Options

guest-mode

Allow users who are not defined in the **User** database to log into the Device Server with any user ID and the specified password. **Guest** users inherit their settings from the **Default User**'s configuration.

password

The password that **Guest** users must use to log into the Device Server.

Set Authentication Kerberos

Description Configures Kerberos authentication settings.

User Level Admin

Syntax

```
set authentication kerberos [kdc-domain <string>]
 [port <TCP_port>] [realm <string>]
```

Options

kdc-domain

The name of a host running the KDC (Key Distribution Center) for the specified realm. The host name that you specify must either be defined in the Device Server's **Host Table** before the last reboot or be resolved by DNS.

port

The port that the Kerberos server listens to for authentication requests. If no port is specified, the default port 88 is used.

realm

The Kerberos realm is the Kerberos host domain name, in upper-case letters.

Set Authentication LDAP

Description Configures LDAP authentication settings.

User Level Admin

Syntax

```
set authentication ldap [base <string>]
 [host <hostname/IP_addr>] [port <TCP_port>] [tls on|off]
 [tls-port <TCP_port>]
```

Options

base

The domain component (dc) that is the starting point for the search for user authentication.

host

The name or IP address of the LDAP host. If you use a host name, that host must either have been defined in the Device Server's **Host Table** before the last reboot or be resolved by DNS. If you are using **TLS**, you must enter the same string you used to create the LDAP certificate that resides on your LDAP server.

port

The port that the LDAP host listens to for authentication requests. The default port is 389.

tls

Enables/disables the Transport Layer Security (TLS) with the LDAP host.

tls-port

Specify the port number that LDAP will use for **TLS**. The default is port 636.

Set Authentication NIS

Description Sets NIS authentication parameters.
User Level Admin
Syntax
```
set authentication nis [domain <string>] [primary <config_host>]
 [secondary <config_host>]
```

Options

domain

The NIS domain name.

primary

The primary NIS host that is used for authentication.

secondary

The secondary NIS host that is used for authentication, should the primary NIS host fail to respond.

Add RADIUS

Description Adds an accounting or authentication RADIUS host.
User Level Admin
Syntax
```
add radius accounting-host <config_host> secret
add radius auth-host <config_host> secret
```

Options

accounting-host

Name of the primary RADIUS accounting host.

Name of the secondary RADIUS accounting host.

auth-host

Name of the primary RADIUS authentication host.

Name of the secondary RADIUS authentication host.

secret

The secret (password) shared between the Device Server and the RADIUS authentication host.

After typing the word **secret** and pressing **Enter**, you will be prompted to enter the secret and then re-enter the secret.

Delete RADIUS

Description Deletes an accounting or authentication RADIUS host.
User Level Admin
Syntax
```
delete radius accounting <accounting_host>
delete radius authentication <authentication_host>
```

Options

accounting

Deletes the specified accounting host from the RADIUS authentication settings.

authentication

Deletes the specified authentication host from the RADIUS authentication settings.

Set Authentication RADIUS

Description Sets RADIUS parameters.
User Level Admin
Syntax

```
set authentication radius [accounting on|off]
 [acct-authenticator on|off] [acct-port <UDP_port>]
 [auth-port <UDP_port>] [retry <integer>] [timeout <integer>]
```

Options

accounting

Enables/disables RADIUS accounting.

acct-authenticator

Enables/disables whether or not the Device Server validates the RADIUS accounting response.

acct-port

The port that the RADIUS host listens to for accounting requests. The default port is 1813.

auth-port

The port that the RADIUS host listens to for authentication requests. The default port is 1812.

retry

The number of times the Device Server tries to connect to the RADIUS server before erroring out. Valid values are 0-255. The default is **5**.

timeout

The time, in seconds, that the Device Server waits to receive a reply after sending out a request to a RADIUS accounting or authentication host. If no reply is received before the timeout period expires, the Device Server will retry the same host up to and including the number of retry attempts. Valid values are 1-255. The default is **3** seconds.

Set Authentication TACACS+

Description Configures TACACS+ authentication settings.
User Level Admin
Syntax

```
set authentication tacacs+ [port <TCP_port>]
 [primary <config_host>] [secondary <config_host>]
 [secret <string>]
```

Options

port

The port number that TACACS+ listens to for authentication requests. The default port number is 49.

primary

The primary TACACS+ host that is used for authentication.

secondary

The secondary TACACS+ host that is used for authentication, should the primary TACACS+ host fail to respond.

secret

The TACACS+ shared secret is used to encrypt/decrypt TACACS+ packets in communications between two devices. The shared secret may be any alphanumeric string. Each shared secret must be configured on both client and server sides.

Set Authentication SecurID

Description Configures SecurID authentication settings.

User Level Admin

Syntax

```
set authentication securid primary [host <config_host>]
 [port <TCP_port>] [encryption des|sdi] [legacy on|off]

set authentication securid replica [host <config_host>]
 [port <TCP_port>] [encryption des|sdi] [legacy on|off]

set authentication securid reset secret
```

Options

primary host

The first SecurID server that is tried for user authentication.

replica host

If the first SecurID server does not respond to an authentication request, this is the next SecurID server that is tried for user authentication.

port

The port number that SecurID listens to for authentication requests. The default port number is 5500.

encryption

You can specify either **SDI** or **DES** encryption for SecurID server communication. The default is **SDI** encryption.

legacy

If you are running SecurID 3.x or 4.x, you need to run in **Legacy Mode**. If you are running SecurID 5.x or above, do not select **Legacy Mode**.

reset secret

Resets the SecurID secret (password) in the Device Server.

Show Authentication

Description Shows the authentication settings. If you type just the **`show authentication`** command, the configured primary and secondary authentication methods are displayed.

User Level Admin

Syntax

```
show authentication radius|ldap|tacacs+|nis|kerberos|securid
```

Option

`radius`|**`ldap`**|**`tacacs+`**|**`nis`**|**`kerberos`**|**`securid`**

Displays the authentication settings for the specified authentication method.

TruePort Baud Commands

Set TruePort Remap-Baud

Description Sets the TruePort baud remapping values.
User Level Admin
Syntax

```
set trueport remap-baud
 50|75|110|134|150|200|300|600|1200|1800|2400|4800|9600|19200|
   38400
 50|75|110|134|150|200|300|600|1200|1800|2400|4800|9600|19200|
   38400|57600|115200|230400|28800|[custom <baud_rate]
```

Options **50|75|110|134|150|200|300|600|1200|1800|2400|4800|9600|19200|38400**

The configured baud rate of the TruePort client.

50|75|110|134|150|200|300|600|1200|1800|2400|4800|9600|19200|38400|57600|115200|230400|28800|[custom <*baud_rate*>**]**

The actual baud rate that runs between the Device Server and the connected serial device. You can also specify a custom baud rate; valid values are 50-230400.

Show TruePort

Description Shows the Device Server TruePort remapping table.
User Level Normal, Admin
Syntax

```
show trueport
```

Email Commands

Set Email-Alert Server

Description Configures email alert settings for the server.
User Level Admin
Syntax

```
set email-alert server [from <email_addr>]
 [level emergency|alert|critical|error|warning|notice|info|debug]
 [mode on|off] [to <email_addr>] [reply-to <email_addr>]
 [smtp-host <string>] [subject <string>]
```

Options **from**

This field can contain an email address that might identify the Device Server name or some other value.

level

Choose the event level that triggers an email notification:

- **Emergency**
- **Alert**
- **Critical**
- **Error**
- **Warning**
- **Notice**
- **Info**
- **Debug**

You are selecting the lowest notification level; therefore, when you select **Debug**, you will get an email notification for all events that trigger a message.

mode

Determines whether or not email notification is turned on. Default is **Off**.

to

An email address or list of email addresses that will receive the email notification.

reply-to

The email address to whom all replies to the email notification should go.

smtp-host

The SMTP host (email server) that will process the email notification request. This can be either a host name defined in the Device Server host table or the SMTP host IP address.

subject

A text string, which can contain spaces, that will display in the **Subject** field of the email notification.

If the text string contains spaces, enclose the string in quotes.

Show Email-Alert Server

Description Shows how the server email alert is configured.
User Level Admin
Syntax

```
show email-alert server
```

Clustering Commands

Add Clustering Slave-IP

Description Adds a slave Device Server to the clustering group.
User Level Admin
Syntax

```
add clustering slave-ip <IPv4_address>
 number-of-ports 1|2|4|8|16|24|32|48 [protocol telnet|ssh]
 [starting-master-tcp-port <10001-65535>]
 [starting-slave-ds-port <10001-65535>]
```

Options ***<IPv4_address>***

Specify the IP address of the Slave Device Server in the clustering group. The IP address must be in a valid IPv4 format.

number-of-ports

Specify the number of ports in the Slave Device Server that you are adding to the clustering group.

protocol

Specify the protocol that will be used to access the Slave Device Server port, SSH or Telnet.

starting-master-tcp-port

Specify the TCP port number you want to map the first Slave Device Server DS Port number to. This number should not be a port number that is already in use by the Master Device Server.

starting-slave-ds-port

Specify the first DS Port number (as specified in the Slave Device Server's Line configuration) on the slave host. By default, this is 10001 and incriments by one for each line/port.

Delete Clustering Slave-IP

Description Deletes a Slave Device Server from the clustering group. Type **`delete clustering slave-ip ?`** to get a list of Slave Device Server IP addresses.

User Level Admin

Syntax **`delete clustering slave-ip <`***`IPv4_address`***`>`**

Option <***IPv4_address***>

Specify the IP address of the Slave Device Server in the clustering group. The IP address must be in a valid IPv4 format.

Set Clustering Slave-IP

Description Adds a slave Device Server to the clustering group.

User Level Admin

Syntax

```
set clustering slave-ip <IPv4_address> port <number>
 [master-tcp-port <10001-65535>] [name <port_name>]
 [protocol telnet|ssh|not-used] [slave-ds-port <10001-65535>]
```

Options <***IPv4_address***>

Specify the IP address of the Slave Device Server in the clustering group. The IP address must be in a valid IPv4 format.

port

Specify the number of ports in the Slave Device Server that you are adding to the clustering group.

master-tcp-port

Specify the TCP port number you want to map the first Slave Device Server DS Port number to. This number should not be a port number that is already in use by the Master Device Server.

name

Specify the TCP port number you want to map the first Slave Device Server DS Port number to. This number should not be a port number that is already in use by the Master Device Server.

protocol

Specify the protocol that will be used to access the port, SSH, Telnet, or Not Used.

slave-ds-port

Specify the first DS Port number (as specified in the Slave Device Server's Line configuration) on the slave host. By default, this is 10001 and incriments by one for each line/port.

Show Clustering Slave-IP

Description Show a Slave Device Server's settings in the clustering group. Type **`show clustering slave-ip ?`** to get a list of Slave Device Server IP addresses.

User Level Admin

Syntax

```
show clustering slave-ip <IPv4_address> [get-port-names]
 [get-port-names-and-save]
```

Options ***<IPv4_address>***

Specify the IP address of the Slave Device Server in the clustering group. The IP address must be in a valid IPv4 format.

get-port-names

Retrieves the port/line names from the specified Slave Device Server.

get-port-names-and-save

Retrieves the port/line names from the specified Slave Device Server and saves them in the Slave Device Server clustering configuration.s

Dynamic DNS Commands

Set Dynamic-DNS

Description Configures the dynamic DNS parameters.

User Level Admin

Syntax

```
set dynamic-dns [on|off]
 [connection-method http|http-port-8245|https]
 [hostname <hostname>] [username <username>]
 [password <password>] [system-type dynamic|static|custom]
 [wildcard enable|disable|nochange]
```

Options **connection-method**

Specify how the Device Server is going to connect to the DynDNS.org server, via HTTP, HTTP through Port 8245, or HTTPS.

hostname

Specify the registered hostname with DynDNS.org that will be updated with the Device Server's IP address should it change. Put in the full name; for example, mydeviceserver.dyndns.org.

username

Specify the user name used to access the DynDNS.org server.

password

Specify the password used to access the DynDNS.org server.

system-type

Specify how your account was set up with DynDNS.org, using a Dynamic, Static, or Custom IP address schema.

wildcard

Adds an alias to **`*.yourhost.ourdomain.ext`** pointing to the same IP address as entered for **`yourhost.ourdomain.ext`**.

Set Dynamic-DNS SSL

Description Sets the SSL/TLS parameters for the connection between the Device Server and the DNS server.

User Level Admin

Syntax

```
set dynamic-dns ssl [verify-peer on|off]
 [validation-criteria
   country <code>|state-province <text>|locality <text>
   |organisation <text>|organisation-unit <text>
   |common-name <text>|email <email_addr>]
```

Options

verify-peer

Enable this option when you want the Validation Criteria to match the Peer Certificate for authentication to pass. If you enable this option, you need to download an SSL/TLS certificate authority (CA) list file to the Device Server.

validation-criteria

Any values that are entered in the validation criteria must match the peer certificate for an SSL connection; any fields left blank will not be validated against the peer certificate.

country

A two character country code; for example, US. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

state-province

Up to a 128 character entry for the state/province; for example, IL. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

locality

Up to a 128 character entry for the location; for example, a city. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

organisation

Up to a 64 character entry for the organisation; for example, Accounting. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

organisation-unit

Up to a 64 character entry for the unit in the organisation; for example, Payroll. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

common-name

Up to a 64 character entry for common name; for example, the host name or fully qualified domain name. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

email

Up to a 64 character entry for an email address; for example, acct@anycompany.com. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Set Dynamic-DNS SSL Cipher-Suite

Description Sets the SSL/TLS cipher suite parameters for the connection between the Device Server and the DNS server.

User Level Admin

Syntax

```
set dynamic-dns ssl cipher-suite
 option1|option2|option3|option4|option5
 encryption any|aes|3des|des|arcfour|arctwo|none
 min-key-size 40|56|64|128|168|256
 max-key-size 40|56|64|128|168|256
 key-exchange any|rsa|edh-rsa|edh-dss|adh
 hmac any|sha1|md5
```

Options **option1|option2|option3|option4|option5**

Sets the priority of the cipher suite, with **`option1`** being highest priority and **`option5`** lowest priority.

encryption

Select the type of encryption that will be used for the SSL connection:

- Any—Will use the first encryption format that can be negotiated.
- AES
- 3DES
- DES
- ARCFOUR
- ARCTWO
- None—Removes any values defined for the cipher option.

The default value is **Any**.

min-key-size

The minimum key size value that will be used for the specified encryption type. The default is **40**.

max-key-size

The maximum key size value that will be used for the specified encryption type. The default is **256**.

key-exchange

The type of key to exchange for the encryption format:

- **Any**—Any key exchange that is valid is used (this does not, however, include ADH keys).
- **RSA**—This is an RSA key exchange using an RSA key and certificate.
- **EDH-RSA**—This is an EDH key exchange using an RSA key and certificate.
- **EDH-DSS**—This is an EDH key exchange using a DSA key and certificate.
- **ADH**—This is an anonymous key exchange which does not require a private key or certificate. Choose this key if you do not want to authenticate the peer device, but you want the data encrypted on the SSL/TLS connection.

The default is **Any**.

hmac

Select the key-hashing for message authentication method for your encryption type:

- Any
- MD5
- SHA1

The default is **Any**.

Show Dynamic-DNS

Description Shows the dynamic DNS settings.
User Level Admin
Syntax `show dynamic-dns`

PCI Commands

Set PCI Card

Description Sets the type of card in the PCI slot.
User Level Admin
Syntax `set pci card none|modem|wireless-wan`

Option **card**

(SCS models only) If you are using an internal PCI modem card or a wireless WAN card in the PCI slot, specify **PCI Modem** or **Wireless WAN**. If you are not using an internal PCI card, keep this parameter as **None**, the default value.

Show PCI

Description Displays the PCI line settings.
User Level Admin
Syntax `show pci`

Set PCI Wireless-WAN

Description Configures the wireless WAN parameters
User Level Admin
Syntax

```
set pci wireless-wan [access-point-name <name>]
 [init-string <modem_init_string>]
 [model sierra|sony-ericsson|standard|custom]
 [password <password>][phone-number <phone_number>]
 [user <username>]
```

Options **access-point-name**

Specify the APN required by your internet provider to access their network. See the internet provider documentation for more information.

init-string

Specify the initialisation string required by your internet service provider for your wireless WAN card.

model

Specify the wireless WAN card you are using. If the wireless WAN card you are using is not listed, try the standard driver. If that does not work, look at the Perle website for a custom driver.

password

Specify the password required by your internet provider to access their network.

phone-number

Specify the phone number provided by your service provider to access their wireless network. The phone number will probably take a format similar to `*99***1#`.

user

Specify the name required by your internet provider to access their network.

Show Wireless-WAN

Description Displays the wireless WAN settings.
User Level Admin
Syntax `show wireless-wan`

User Commands

Logged Into the Device Server Commands

Admin

Description Changes a Normal-level user to the Admin user. When you press **Enter** after you type this command, you will be prompted for the Admin password.
User Level Normal
Syntax `admin`

Help

Description Displays help on using the command line interface (CLI).
User Level Restricted, Normal, Admin
Syntax `help`

Kill Line

Description Restarts a line. On 2+ port Device Servers, you can specify a port number and then a range of ports; for example, `kill line 4, 10-13, 15`. This also resets the Perle PCI modem card on SDS M models. 1-port models use simply `kill line`.
User Level Normal, Admin
Syntax `kill line *|<number>|<number range>`

Kill Session

Description Kills an active session.
User Level Restricted, Normal, Admin
Syntax `kill session 1|2|3|4`

Options **1**|**2**|**3**|**4**

The number of the session you want to kill.

Logout

Description Logs the user out from the Device Server.
User Level Restricted, Normal, Admin
Syntax `logout`

Menu

Description Switches from the CLI mode to the Menu.
User Level Restricted, Normal, Admin
Syntax `menu`

Ping

Description Pings the specified host/IP address.
User Level Normal, Admin
Syntax `ping <hostname/IP_address> [<packet_size>] [<#_of_packets>]`

Options *<hostname/IP_address>*

The name (host name or DNS name) or IP address of the machine you are trying to ping (verify the connection with).

<packet_size>

Enter the number of data bytes to be sent. The maximum size is 2000 bytes.

<#_of_packets>

Enter the number of the packets you want to send.

Resume

Description Resumes a started session.
User Level Restricted, Normal, Admin
Syntax `resume 1|2|3|4`

Options **1|2|3|4**

The number of the session you want to resume.

Rlogin

Description Starts an rlogin session to the specified host/IP address.
User Level Normal, Admin
Syntax `rlogin <hostname/IP_address> [termtype <terminal_name>] [user <string>]`

Options *<hostname/IP_address>*

The name of the target host.

termtype

Type of terminal attached to this line; for example, ansi or wyse60.

user

The name of the user logging into the rlogin session.

Screen

Description Switches from the CLI mode to the Menu.
User Level Restricted, Normal, Admin
Syntax `screen`

Set Termtype

Description Sets the type of terminal being used for the current session.
User Level Normal, Admin
Syntax

```
set termtype
 wyse60|vt100|ansi|dumb|tvi925|ibm3151te|vt320|hp700|term1|term2|
 term3
```

Option **wyse60|vt100|ansi|dumb|tvi925|ibm3151te|vt320|hp700|term1|term2|term3**

Specifies the type of terminal connected to the line:

- **Dumb**
- **WYSE60**
- **VT100**
- **ANSI**
- **TVI925**
- **IBM3151TE**
- **VT320** (specifically supporting VT320-7)
- **HP700** (specifically supporting HP700/44)
- **Term1, Term2, Term3** (user-defined terminals)

Set User

Description Sets the current user's settings.
User Level Normal, Admin
Syntax

```
set user . [hotkey-prefix <00-7f>] [language english|customlang]
 [routing none|send|listen|send-and-listen] [password]
```

Options **hotkey-prefix**

The prefix that a user types to control the current session. The default value is **hex 01**, which corresponds to **Ctrl-a** (**^a**) (hex value 02 would be Ctrl-b (^b), etc.):

- **^a *number***—To switch from one session to another, press **^a** and then the required session number. For example, **^a 2** would switch you to session 2. Pressing **^a 0** will return you to the Device Server Menu.
- **^a n**—Display the next session. The current session will remain active. The lowest numbered active session will be displayed.
- **^a p**—Display the previous session. The current session will remain active. The highest numbered active session will be displayed.
- **^a m**—To exit a session and return to the Device Server. You will be returned to where you left off. The session will be left running.
- **^a l**—(Lowercase L) Locks the line until the user unlocks it. The user is prompted for a password (any password, excluding spaces) and locks the line. Next, the user must retype the password to unlock the line.
- **^r**—When you switch from a session back to the Menu, the screen may not be redrawn correctly. If this happens, use this command to redraw it properly. This is always **Ctrl R**, regardless of the **Hotkey Prefix**.

The **User Hotkey Prefix** value overrides the **Line Hotkey Prefix** value. You can use the **Hotkey Prefix** keys to lock a line only when the line **Lock** parameter is **On**.

language

You can specify whether a user will use **English** or **Customlang** as the language that appears in the Menu, CLI, or WebManager. The Device Server supports one custom language that must be downloaded to the Device Server; otherwise, **Customlang** defaults to English.

routing

Determines the routing mode used for RIP packets on the PPP and SLIP interfaces. Values are:

- **None**—RIP packets are neither received nor sent by the Device Server.
- **Send**—RIP packets can only be sent by the Device Server.
- **Listen**—RIP packets can only be received by the Device Server.
- **Send and Listen**—RIP packets are sent and received by the Device Server.

password

The password the user will need to enter to login to the Device Server. This case-sensitive field accepts a maximum of 16 characters.

Set User Session

Description Sets the current user's session settings.

User Level Normal, Admin

Syntax

```
set user . session 1|2|3|4|* [auto on|off]
 [type off|telnet|rlogin|ssh]

set user . session 1|2|3|4|* telnet-options [host <config_host>]
 [port <TCP_port>] [termtype <terminal_name>] [line-mode on|off]
 [map-cr-crlf on|off] [local-echo on|off] [echo <00-7f>]
 [eof <00-7f>] [erase <00-7f>] [intr <00-7f>] [quit <00-7f>]

set user . session 1|2|3|4|* rlogin-options [host <config_host>]
 [termtype <terminal_name>]

set user . session 1|2|3|4|* ssh-options [host <config_host>]
 [port <TCP_port>] [termtype <terminal_name>]
 [protocol ssh-1|ssh-2|ssh-2/1] [compression on|off]
 [verbose on|off] [auto-login on|off] [name <string>]
 [password <string>] [ssh-1-cipher 3des|des|blowfish]
 [authentication rsa on|off] [authentication dsa on|off]
 [authentication keyboard-interactive on|off]

set user . session 1|2|3|4|* ssh-options
 ssh-2-cipher-list <3des blowfish cast aes arcfour>
```

Options

session

Specifies the session number (or all, *) that you are configuring.

auto

Specify whether or not the session(s) will start automatically when the user logs into the Device Server.

telnet-options

See ***Set Telnet-Client*** on page 295.

rlogin-options

See ***Set Rlogin-Client*** on page 294.

ssh-options

See ***Set SSH-Client*** on page 296.

Show Line Users

Description Shows the users who are on the line.

User Level Admin

Syntax

```
show line users
```

SSH

Description Starts an SSH session to the specified host/IP address.

User Level Normal, Admin

Syntax

```
ssh <hostname/IP_address> [<TCP_port>]
 [termtype <terminal_name>] [authentication rsa on|off]
 [authentication dsa on|off]
 [authentication keyboard-interactive on|off]
 [compression on|off] [protocol ssh-1|ssh-2|ssh-2,1]
 [ssh-1-cipher 3des|des|blowfish]
 [ssh-2-cipher-list <3des blowfish cast aes arcfour> end-list]
 [user <name>] [verbose on|off]
```

Options

<hostname/IP_address>

The name of the target host.

<TCP_port>

The port number the target host is listening on for incoming connections.

termtype

Type of terminal attached to this line; for example, ANSI or WYSE60.

authentication rsa

An authentication method used by SSH version 1 and 2. When enabled, an SSH client session will try to authenticate via RSA.

authentication dsa

An authentication method used by SSH version 2. When enabled, an SSH client session will try to authenticate via DSA.

authentication keyboard-interaction

The user types in a password for authentication.

compression

Requests compression of all data. Compression is desirable on modem lines and other slow connections, but will only slow down things on fast networks.

protocol

Specify whether you are using SSH-1, SSH-2, or a combination of the two protocols, SSH-2, SSH-1.

ssh-1-cipher

Select the encryption method (cipher) that you want to use for your SSH version 1 connection:

- **3DES**
- **Blowfish**

ssh-2-cipher-list

Select the order of negotiation for the encryption method (ciphers) that the Device Server will use for the SSH version 2 connection:

- **3DES**
- **Blowfish**
- **AES**
- **Arcfour**
- **CAST**

user

The name of the user logging into the SSH session.

verbose

Displays debug messages on the terminal.

Syslog Console

Description Starts/stops or displays the status of the syslog console.
User Level Admin
Syntax `syslog console start|stop`

`syslog console status`

Options **start|stop**

Start or stop console logging. When console logging is enabled, syslog messages will be echoed to the current console. These messages are filtered based on the level set in the (remote) syslog options.

status

Displays the current console logging status (enabled or disabled).

Show Sessions

Description Shows available sessions.
User Level Restricted, Normal, Admin
Syntax `show sessions`

Show Termtype

Description Shows the terminal type for the current session.
User Level Admin
Syntax `show termtype`

Start

Description Starts a predefined session. Only inactive sessions are displayed.
User Level Restricted, Normal, Admin
Syntax `start 1|2|3|4`

Options **1|2|3|4**

The number of the session that you want to start.

Telnet

Description Starts a telnet session to the specified host/IP address.

User Level Normal, Admin

Syntax

```
telnet <hostname/IP_address> [<TCP_port>]
 [termtype <terminal_name>] [line-mode on|off]
 [map-cr-crlf on|off] [local-echo on|off]
 [echo <00-7f>] [eof <00-7f>] [erase <00-7f>] [intr <00-7f>]
 [quit <00-7f>] [escape <00-7f>]
```

Options

<hostname/IP_address>

The name of the target host.

<TCP_port>

The port number the target host is listening on for incoming connections.

termtype

Type of terminal attached to this line; for example, ANSI or WYSE60.

line-mode

When **On**, keyboard input is not sent to the remote host until **Enter** is pressed, otherwise input is sent every time a key is pressed. Default is **Off**.

map-cr-crlf

Maps carriage returns (CR) to carriage return line feed (CRLF). The default value is **Off**.

local-echo

Toggles between local echo of entered characters and suppressing local echo. Local echo is used for normal processing, while suppressing the echo is convenient for entering text that should not be displayed on the screen, such as passwords. This parameter can only be used when **Line Mode** is **On**. Default is **Off**.

echo

Defines the echo character. When **Line Mode** is **On**, typing the echo character echoes the text locally and sends only completed lines to the host. This value is in hexadecimal with a default value of **5** (ASCII value **^E**).

eof

Defines the end-of-file character. When **Line Mode** is **On**, entering the EOF character as the first character on a line sends the character to the remote host. This value is in hexadecimal with a default value of **4** (ASCII value **^D**).

erase

Defines the erase character. When **Line Mode** is **Off**, typing the erase character erases one character. This value is in hexadecimal with a default value of **8** (ASCII value **^H**).

intr

Defines the interrupt character. Typing the interrupt character interrupts the current process. This value is in hexadecimal with a default value of **3** (ASCII value **^C**).

quit

Defines the quit character. Typing the quit character closes and exits the current telnet session. This value is in hexadecimal with a default value of **1c** (ASCII value **FS**).

escape

Defines the escape character. Returns you to the command line mode. This value is in hexadecimal with a default value of **1d** (ASCII value **GS**).

Version

Description Displays firmware version and build.
User Level Normal, Admin
Syntax `version`

Configuring Users

Add User

Description Adds a user. You can add and configure up to four users in the Device Server.
User Level Admin
Syntax `add user <username>`

Option <***username***>

The name of the user, without spaces. When you finish the command and press Enter, you will be prompted to enter and re-enter a password for the user.

Delete User

Description Deletes a user.
User Level Admin
Syntax `delete user <config_user>`

Option <***config_user***>

You can see a list of users that can be deleted by typing `delete user ?`.

Set Default User

Description Configures the Default User.
User Level Admin
Syntax

```
set default user [callback on|off] [framed-compression on|off]
 [framed-ip <IPv4_address>]
 [framed-interface-id <IPv6_interface_id>]
 [framed-mtu <64-1500>] [hotkey-prefix <00-7f>]
 [idle-timer <0-4294967>]
 [host-ip None|<IP_address>|<config_host>]
 [language english|customlang]
 [level admin|normal|restricted|menu]
 [line-access readin|readout|readwrite on|off]
 [netmask <IPv4_address>] [phone-number <phone_number>]
 [routing none|send|listen|send-and-listen]
 [service dsprompt|telnet|tcp-clear|rlogin|ppp|slip|ssh]
 [sess-timer <0-4294967>] [port tcp-clear|telnet|ssh <TCP_port>]
 [access-clustered-ports on|off]
```

Options **callback**

When **On**, enter a phone number for the Device Server to call the user back (the **Callback** parameter is unrelated to the **Line Dial** parameter).

Note: the Device Server will allow callback only when a user is authenticated. If the protocol over the link does not provide authentication, there will be no callback. Therefore, when the **Line Service** is set to **PPP**, you must use either **PAP** or **CHAP**, because these protocols provide authentication. The default is **Off**.

The Device Server supports another type of callback, **Roaming Callback**, which is configurable when the **Line Service** is set to **PPP**.

framed-compression

Used for **User Service PPP** or **SLIP**, determines whether Van Jacobsen Compression is used on the link. VJ compression is a means of reducing the standard TCP/IP header from 40 octets to approximately 5 octets. This gives a significant performance improvement, particularly when interactive applications are being used. For example, when the user is typing, a single character can be passed over the link with a 40 octet header attached. VJ Compression has little effect on other types of links, such as ftp, where the packets are much larger. The **Framed Compression** value will be used in preference to the **VJ Compression** value set for a **Line**. The default is **Off**.

framed-ip

Used for **User Service PPP** or **SLIP**, sets the IP address of the remote user. Enter the address in dot decimal notation as follows:

- **255.255.255.254** (default)—The Device Server will use the **Remote IP Address** set in the **PPP** settings for the line.
- **255.255.255.255**—When the **User Service** is **PPP**, the Device Server will allow the remote machine to specify its IP address (overriding the IP address negotiation value configured in the **PPP** settings).
- **255.255.255.255**—When the **User Service** is **SLIP**, the Device Server will use the **Remote IP Address** set for the line (no negotiation).
- **n.n.n.n**—(where **n** is a number) Enter the IP address of your choice. This IP address will then be used in preference to the **Remote IP Address** set for a line.

framed-interface-id

Used for **User Service PPP**, sets the IP address of the remote user. Enter the address in IPv6 format. The first 64 bits of the Interface Identifier must be zero, therefore, ::abcd:abcd:abcd:abcd is the expected format.

framed-mtu

Used for **User Service PPP** or **SLIP**, specifies the maximum size of packets, in bytes, being transferred across the link. On noisy links it might be preferable to fragment large packets being transferred over the link, since there will be quicker recovery from errors. Depending on whether you have selected a **User Service** of **SLIP** or **PPP**, details are as follows:

- **PPP**—**Framed MTU** will be the maximum size of packets that the Device Server port will accept. This value is negotiated between the two ends of the link. The valid range is 64-1500. The default value is **1500** bytes.
- **SLIP**—**Framed MTU** will be the maximum size of packets being sent by the Device Server. The Device Server will send SLIP packets in the range 256-1006 bytes. The default value is **256** bytes.

The **Framed MTU** value will be used in preference to the **MTU**/**MRU** values set for a **Line**.

hotkey-prefix

The prefix that a user types to control the current session. The default value is **hex 01**, which corresponds to **Ctrl-a** (**^a**) (hex value 02 would be Ctrl-b (^b), etc.):

- **^a *number***—To switch from one session to another, press **^a** and then the required session number. For example, **^a 2** would switch you to session 2. Pressing **^a 0** will return you to the Device Server Menu.
- **^a n**—Display the next session. The current session will remain active. The lowest numbered active session will be displayed.
- **^a p**—Display the previous session. The current session will remain active. The highest numbered active session will be displayed.
- **^a m**—To exit a session and return to the Device Server. You will be returned to where you left off. The session will be left running.
- **^a l**—(Lowercase L) Locks the line until the user unlocks it. The user is prompted for a password (any password, excluding spaces) and locks the line. Next, the user must retype the password to unlock the line.
- **^r**—When you switch from a session back to the Menu, the screen may not be redrawn correctly. If this happens, use this command to redraw it properly. This is always **Ctrl R**, regardless of the **Hotkey Prefix**.

The **User Hotkey Prefix** value overrides the **Line Hotkey Prefix** value. You can use the **Hotkey Prefix** keys to lock a line only when the line **Lock** parameter is **On**.

idle-timer

The amount of time, in seconds, that the **Idle Timer** will run. Use this timer to close a connection because of inactivity. When the **Idle Timer** expires, because there has been no exchange of data within the specified time, the Device Server will close the connection. The default value is **0** (zero), meaning that the **Idle Timer** will not expire (the connection is open permanently). The maximum value is 4294967 seconds. The **User Idle Timer** will override the **Line Idle Timer**, with the exception of reverse SSH or reverse Telnet sessions.

host-ip

When the **User Service** is set to **Telnet**, **Rlogin**, **SSH**, or **TCP_clear**, the target host IP address or preconfigured host name. If no IP address is specified, the **Host IP** value in the **Default User** configuration will be used. The default is **0.0.0.0**. or None.

language

You can specify whether a user will use **English** or **Customlang** as the language that appears in the Menu, CLI, or WebManager. The Device Server supports one custom language that must be downloaded to the Device Server; otherwise, **Customlang** defaults to English.

level

The access that a user is allowed:

- **Admin**—The admin level user has total access to the Device Server. You can create more than one admin user account but we recommend that you only have one. They can monitor and configure the Device Server.
- **Normal**—The Normal level user has limited access to the Device Server. Limited CLI commands and Menu access are available with the ability to configure the user's own configuration settings.
- **Restricted**—The Restricted level user can only access predefined sessions or access the Easy Port Access menu.
- **Menu**—The menu level user will only be able to access predefined session or access the Easy Port Access menu. The Easy Port Access allows the user to connect to the accessible line without disconnecting their initial connection to the Device Server. Does not have any access to CLI commands.

netmask

(IPV4 only) If the remote user is on a subnet, enter the network's subnet mask. For example, a subnet mask of 255.255.0.0.

line-access

Specifies the user access rights to each Device Server device line. Options are:

- **Read/Write**—Users are given read and write access to the line.
- **Read In**—Users are given access to read only outbound data, data that is going from the Device Server to the device.
- **Read Out**—Users are given access to read only inbound data, data that is going from the device to the Device Server.

Users can read data going in both directions by selecting both the **Read In** and **Read Out** options.

phone-number

The phone number the Device Server will dial to callback the user (you must have set **Callback** to **On**). Enter the number without spaces. To change the phone number, overwrite the previous entry; to clear the phone number, set it to **""** (double quotes without a space).

routing

Determines the routing mode used for RIP packets on the PPP and SLIP interfaces. Values are:

- **None**—RIP packets are neither received nor sent by the Device Server.
- **Send**—RIP packets can only be sent by the Device Server.
- **Listen**—RIP packets can only be received by the Device Server.
- **Send and Listen**—RIP packets are sent and received by the Device Server.

service

The type of service that the user will use.

sess-timer

The amount of time, in seconds, that the **Session Timer** will run. Use this timer to forcibly close a user's session (connection). When the **Session Timer** expires, the Device Server will end the connection. The default value is **0** (zero), meaning that the session timer will not expire (the session is open permanently, or until the user logs out). The maximum value is 4294967 seconds. The **User Session Timer** will override the **Line Session Timer**, with the exception of reverse SSH or reverse Telnet sessions.

port

When the **User Service** is **Telnet**, **TCP_clear**, or **SSH**, this is the target port number. The default value will change based on the type of **Service** selected; the most common known port numbers are used as the default values.

access-clustered-ports

When enabled, allows the user access to Device Servers that have been configured in the clustering group. The default is on.

Set User

Description Sets user's settings. Normal-level users can configure only their own settings. Admin-level users can configure any user's settings, including their own (with the exception of their User Level, which must stay at Admin).

User Level Normal, Admin

Syntax

```
set user . [hotkey-prefix <00-7f>] [language english|customlang]
 [password] [routing none|send|listen|send-and-listen]
```

Admin User Only

```
set user .|<username>|* [callback on|off]
 [framed-compression on|off] [framed-ip <IPv4_address>]
 [framed-interface-id <IPv6_interface_id>]
 [framed-mtu <64-1500>] [hotkey-prefix <00-7f>]
 [idle-timer <0-4294967>]
 [host-ip None|<IP_address>|<config_host>]
 [language english|customlang]
 [level admin|normal|restricted|menu] [password]
 [line-access readin|readout|readwrite on|off]
 [netmask <IPv4_address>] [phone-number <phone_number>]
 [routing none|send|listen|send-and-listen]
 [service dsprompt|telnet|tcp-clear|rlogin|ppp|slip|ssh]
 [sess-timer <0-4294967>] [port tcp-clear|telnet|ssh <TCP_port>]
 [access-clustered-ports on|off]
```

Options

callback

When **On**, enter a phone number for the Device Server to call the user back (the **Callback** parameter is unrelated to the **Line Dial** parameter).

Note: the Device Server will allow callback only when a user is authenticated. If the protocol over the link does not provide authentication, there will be no callback. Therefore, when the **Line Service** is set to **PPP**, you must use either **PAP** or **CHAP**, because these protocols provide authentication. The default is **Off**.

The Device Server supports another type of callback, **Roaming Callback**, which is configurable when the **Line Service** is set to **PPP**.

framed-compression

Used for **User Service PPP** or **SLIP**, determines whether Van Jacobsen Compression is used on the link. VJ compression is a means of reducing the standard TCP/IP header from 40 octets to approximately 5 octets. This gives a significant performance improvement, particularly when interactive applications are being used. For example, when the user is typing, a single character can be passed over the link with a 40 octet header attached. VJ Compression has little effect on other types of links, such as ftp, where the packets are much larger. The **Framed Compression** value will be used in preference to the **VJ Compression** value set for a **Line**. The default is **Off**.

framed-ip

Used for **User Service PPP** or **SLIP**, sets the IP address of the remote user. Enter the address in dot decimal notation as follows:

- **255.255.255.254** (default)—The Device Server will use the **Remote IP Address** set in the **PPP** settings for the line.
- **255.255.255.255**—When the **User Service** is **PPP**, the Device Server will allow the remote machine to specify its IP address (overriding the IP address negotiation value configured in the **PPP** settings).
- **255.255.255.255**—When the **User Service** is **SLIP**, the Device Server will use the **Remote IP Address** set for the line (no negotiation).
- **n.n.n.n**—(where **n** is a number) Enter the IP address of your choice. This IP address will then be used in preference to the **Remote IP Address** set for a line.

framed-interface-id

Used for **User Service PPP**, sets the IP address of the remote user. Enter the address in IPv6 format. The first 64 bits of the Interface Identifier must be zero, therefore, ::abcd:abcd:abcd:abcd is the expected format.

framed-mtu

Used for **User Service PPP** or **SLIP**, specifies the maximum size of packets, in bytes, being transferred across the link. On noisy links it might be preferable to fragment large packets being transferred over the link, since there will be quicker recovery from errors. Depending on whether you have selected a **User Service** of **SLIP** or **PPP**, details are as follows:

- **PPP**—**Framed MTU** will be the maximum size of packets that the Device Server port will accept. This value is negotiated between the two ends of the link. The valid range is 64-1500. The default value is **1500** bytes.
- **SLIP**—**Framed MTU** will be the maximum size of packets being sent by the Device Server. The Device Server will send SLIP packets in the range 256-1006 bytes. The default value is **256** bytes.

The **Framed MTU** value will be used in preference to the **MTU**/**MRU** values set for a **Line**.

hotkey-prefix

The prefix that a user types to control the current session. The default value is **hex 01**, which corresponds to **Ctrl-a** (**^a**) (hex value 02 would be Ctrl-b (^b), etc.):

- **^a *number***—To switch from one session to another, press **^a** and then the required session number. For example, **^a 2** would switch you to session 2. Pressing **^a 0** will return you to the Device Server Menu.
- **^a n**—Display the next session. The current session will remain active. The lowest numbered active session will be displayed.
- **^a p**—Display the previous session. The current session will remain active. The highest numbered active session will be displayed.
- **^a m**—To exit a session and return to the Device Server. You will be returned to where you left off. The session will be left running.
- **^a l**—(Lowercase L) Locks the line until the user unlocks it. The user is prompted for a password (any password, excluding spaces) and locks the line. Next, the user must retype the password to unlock the line.
- **^r**—When you switch from a session back to the Menu, the screen may not be redrawn correctly. If this happens, use this command to redraw it properly. This is always **Ctrl R**, regardless of the **Hotkey Prefix**.

The **User Hotkey Prefix** value overrides the **Line Hotkey Prefix** value. You can use the **Hotkey Prefix** keys to lock a line only when the line **Lock** parameter is **On**.

idle-timer

The amount of time, in seconds, that the **Idle Timer** will run. Use this timer to close a connection because of inactivity. When the **Idle Timer** expires, because there has been no exchange of data within the specified time, the Device Server will close the connection. The default value is **0** (zero), meaning that the **Idle Timer** will not expire (the connection is open permanently). The maximum value is 4294967 seconds. The **User Idle Timer** will override the **Line Idle Timer**, with the exception of reverse SSH or reverse Telnet sessions.

host-ip

When the **User Service** is set to **Telnet**, **Rlogin**, **SSH**, or **TCP_clear**, the target host IP address or preconfigured host name. If no IP address is specified, the **Host IP** value in the **Default User** configuration will be used. The default is **0.0.0.0**. or None.

language

You can specify whether a user will use **English** or **Customlang** as the language that appears in the Menu, CLI, or WebManager. The Device Server supports one custom language that must be downloaded to the Device Server; otherwise, **Customlang** defaults to English.

level

The access that a user is allowed:

- **Admin**—The admin level user has total access to the Device Server. You can create more than one admin user account but we recommend that you only have one. They can monitor and configure the Device Server.
- **Normal**—The Normal level user has limited access to the Device Server. Limited CLI commands and Menu access are available with the ability to configure the user's own configuration settings.
- **Restricted**—The Restricted level user can only access predefined sessions or access the Easy Port Access menu.
- **Menu**—The menu level user will only be able to access predefined session or access the Easy Port Access menu. The Easy Port Access allows the user to connect to the accessible line without disconnecting their initial connection to the Device Server. Does not have any access to CLI commands.

line-access

Specifies the user access rights to each Device Server device line. Options are:

- **Read/Write**—Users are given read and write access to the line.
- **Read In**—Users are given access to read only outbound data, data that is going from the Device Server to the device.
- **Read Out**—Users are given access to read only inbound data, data that is going from the device to the Device Server.

Users can read data going in both directions by selecting both the **Read In** and **Read Out** options.

netmask

(IPV4 only) If the remote user is on a subnet, enter the network's subnet mask. For example, a subnet mask of 255.255.0.0.

password

The password the user will need to enter to login to the Device Server. This case-sensitive field accepts a maximum of 16 characters.

phone-number

The phone number the Device Server will dial to callback the user (you must have set **Callback** to **On**). Enter the number without spaces. To change the phone number, overwrite the previous entry; to clear the phone number, set it to **""** (double quotes without a space).

routing

Determines the routing mode used for RIP packets on the PPP and SLIP interfaces. Values are:

- **None**—RIP packets are neither received nor sent by the Device Server.
- **Send**—RIP packets can only be sent by the Device Server.
- **Listen**—RIP packets can only be received by the Device Server.
- **Send and Listen**—RIP packets are sent and received by the Device Server.

service

The type of service that the user will use.

sess-timer

The amount of time, in seconds, that the **Session Timer** will run. Use this timer to forcibly close a user's session (connection). When the **Session Timer** expires, the Device Server will end the connection. The default value is **0** (zero), meaning that the session timer will not expire (the session is open permanently, or until the user logs out). The maximum value is 4294967 seconds. The **User Session Timer** will override the **Line Session Timer**, with the exception of reverse SSH or reverse Telnet sessions.

port

When the **User Service** is **Telnet**, **TCP_clear**, or **SSH**, this is the target port number. The default value will change based on the type of **Service** selected; the most common known port numbers are used as the default values.

access-clustered-ports

When enabled, allows the user access to Device Servers that have been configured in the clustering group. The default is on.

Set User Session

Description Configures a user's session settings. See ***Set User Session*** on page 273 for the options descriptions.

User Level Admin

Syntax

```
set user .|<username>|* session 1|2|3|4|* [auto on|off]
 [type off|telnet|rlogin|ssh]

set user .|<username>|* session 1|2|3|4|* telnet-options
 [host <config_host>] [port <TCP_port>]
 [termtype <terminal_name>] [line-mode on|off]
 [map-cr-crlf on|off] [local-echo on|off]
 [echo <00-7f>] [eof <00-7f>] [erase <00-7f>] [intr <00-7f>]
 [quit <00-7f>]

set user .|<username>|* session 1|2|3|4|* rlogin-options
 [host <config_host>] [termtype <terminal_name>]

set user .|<username>|* session 1|2|3|4|*
 ssh-options [host <config_host>] [port <TCP_port>]
 [termtype <terminal_name>] [protocol ssh-1|ssh-2|ssh-2/1]
 [compression on|off] [verbose on|off] [auto-login on|off]
 [name <string>] [password <string>]
 [ssh-1-cipher 3des|des|blowfish] [authentication rsa on|off]
 [authentication password on|off]
 [authentication keyboard-interactive on|off]

set user .|<username>|* session 1|2|3|4|* ssh-options
 ssh-2-cipher-list <3des blowfish cast aes arcfour>
```

Show Default User

Description Shows the Default User's settings.

User Level Admin

Syntax

```
show default user
```

Show User

Description Shows user configuration settings.
User Level Admin
Syntax **`show user <`*`configured_user`*`>|.`**

Options <***configured_user***>

Show the settings for the specified user.

.

Show the settings for the current user.

Line Commands

1-Port vs. 2-Port+ Line Commands

If you are using a 1-port Device Server, the admin user does not have the option of using the number or all (*) options in the line commands, as there is only one line. In a 2-port+ Device Server, the admin user must specify **.** (current line), <***number***> (line number), or ***** (sets value for all lines) when configuring lines.

Line Commands

Set Line

Description Configures line parameters. The `set line` command does not work on modem ports/lines on models that have either an internal modem or a PCI modem card.
User Level Normal, Admin
Syntax

```
set line . [data-bits 5|6|7|8]
 [connection-method dial-in|dial-out|dial-in-out|direct-connect|
    ms-direct-host|ms-direct-guest]
 [idle-timer <0-4294967>] [line-name <name>]
 [modem-name <config_modem>] [pages 1|2|3|4|5|6|7]
 [parity none|even|odd|mark|space] [phone-number <phone_number>]
 [rev-sess-security on|off] [sess-timer <0-4294967>]
 [stop-bits 1|2|1.5] [termtype wyse60|vt100|ansi|dumb|tvi925|
   ibm3151te|vt320|hp700|term1|term2|term3]
```

Admin User Only

```
set line .|<number>|* ... [mode enabled|disabled] [break on|off]
 [map-cr-crlf on|off] [flowin on|off] [flowout on|off]
 [hotkey-prefix <00-7f>] [initial cli|menu] [keepalive on|off]
 [lock on|off] [motd on|off] [multisessions <integer>]
 [reset on|off] [dial-timeout <number>] [dial-retries <number>]
 [user <name>] [nouser] [line-termination on|off]
 [internet-address <IPv4_address>]
```

Options **mode**

Enables/disables a line (available only on 2-port+ models). The default is enabled.

data-bits

Specifies the number of bits in a byte. The default is **8**.

connection-method

Determines how a modem will work on the line. Select from the following options:

- **Direct Connect**—Indicates that there is not a modem on the line. This is the default.
- **Dial In**—Specify this option when a user is remote and will be dialing in via modem or ISDN TA.
- **Dial Out**—Specify this option when a modem is attached to the serial port and is being used to dial out.
- **Dial In/Out**—Specify this option when the Device Server is being used as a router (depending on which end of the link your Device Server is situated and how you want to initiate the communication).
- **MS Direct-Host**—Specify this option when the serial port is connected to a Microsoft Guest device. **Line Service** must be set to **PPP** for this option.
- **MS Direct-Guest**—Specify this option when the serial port is connected to a Microsoft Host device. **Line Service** must be set to **PPP** for this option.

idle-timer

Enter a time period, in seconds, for which the **Idle Timer** will run. Use this timer to close a connection because of inactivity. When the **Idle Timer** expires, the Device Server will end the connection. The maximum value is 4294967 seconds (about 49 days). The default value of **0** (zero) means the **Idle Timer** will not expire, so the connection is permanently open.

line-name

Provide a name for the line so it can be easily identified. The **Remote Port Buffering** logging feature uses the **Line Name** when creating a file on the remote NFS server.

modem-name

The name of the predefined modem that is used on this line.

pages

For **DSLogin** line service, this is the number of video pages the terminal supports. Valid values are 1-7. The default is **5** pages.

parity

Specifies if you are using **Even**, **Odd**, or **No parity** on the line. If you want to force a parity type, you can specify **Mark** for 1or **Space** for 0.

phone-number

The phone number to use when **Connection Method** is set to **Dial Out**.

rev-sess-security

Enables/disables login/password authentication, locally or externally, on reverse Telnet connections. The default is **Off**.

sess-time

Enter a time, in seconds, for which the **Session Timer** will run. Use this timer to forcibly close the session (connection). When the **Session Timer** expires, the Device Server will end the connection. The default value is **0** seconds so the port will never timeout. The maximum value is 4294967 seconds (about 49 days).

break

Specifies how a break is interpreted:

- **off**—The Device Server ignores the break key completely and it is not passed through to the host. This is the default setting.
- **local**—The Device Server deals with the break locally. If the user is in a session, the break key has the same effect as a hot key.
- **remote**—When the break key is pressed, the Device Server translates this into a telnet break signal which it sends to the host machine.
- **break-interrupt**—On some systems such as SunOS, XENIX, and AIX, a break received from the peripheral is not passed to the client properly. If the client wishes to make the break act like an interrupt key (for example, when the stty options **`-ignbrk`** and **`brkintr`** are set).

map-cr-crlf

When **Line Service Printer** is selected, defines the default end-of-line terminator as CR-LF (ASCII carriage-return line-feed) when enabled. Default is **Off**.

flowin

Determines if input flow control is to be used. Default is **On**. This is active only when **Line Flow Control** is set to **Soft**, **Hard**, or **Both**.

flowout

Determines if output flow control is to be used. Default is **On**. This is active only when **Line Flow Control** is set to **Soft**, **Hard**, or **Both**.

hotkey-prefix

The prefix that a user types to lock a line or redraw the Menu. The default value is **hex 01**, which corresponds to **Ctrl-a** (**^a**) (hex value 02 would be Ctrl-b (^b), etc.):

- **^a l**—(Lowercase L) Locks the line until the user unlocks it. The user is prompted for a password (any password, excluding spaces) and locks the line. Next, the user must retype the password to unlock the line.
- **^r**—When you switch from a session back to the Menu, the screen may not be redrawn correctly. If this happens, use this command to redraw it properly. This is always **Ctrl R**, regardless of the **Hotkey Prefix**.

You can use the **Hotkey Prefix** key to lock a line only when the **Line Lock** parameter is **On**.

initial

Specifies the initial interface a user navigates when logging into the line; either the **Menu** or a prompt for the **CLI**. The default is **CLI**.

keepalive

Enables a per-connection TCP keepalive feature; after approximately 3 minutes of network connection idle time, the connection will send a gratuitous ACK to the network peer, either ensuring the connection stays active OR causing a dropped connection condition to be recognised by the reverse raw service.

Applications using this feature need to be aware that there might be some considerable delay between a network disconnection and the port being available for the next connection attempt; this is to allow any data sent on prior connections to be transmitted out of the serial port buffer. Application network retry logic needs to accommodate this feature.

lock

When enabled, the user can lock his terminal with a password using the **Hotkey Prefix** (default Ctrl-a) **^a l** (lowercase L). The Device Server prompts the user for a password and a confirmation.

motd

Enables/disables the message of the day on the line.

multisessions

The number of extra reverse sessions available on a line (available only on 2 port+ models), in addition to the single session that is always available on the line. You can specify **0-7** multisessions per line. The default is **0** (zero). Total sessions available for the Device Server are 1-8 for the 2-/4-port models and 2x the number of ports for all other models.

user

For **DSLogin** line service, makes this a line that is dedicated to the specified user. Only this user will be able to log in on this line and they won't need to enter their login name - just their password. When the **Line Service** is set to **Direct** or **Silent Rlogin**, the **User** parameter is used as the Rlogin user name (since Rlogin will not prompt you for a user name).

nouser

Blanks out the User parameter, in case you want to change a dedicated user line to an undedicated line.

reset

Resets the terminal type connected to the line when a user logs out.

dial-timeout

The number of seconds the Device Server will wait to establish a connection to a remote modem. The default value is **45** seconds.

dial-retries

The number of times the Device Server will attempt to re-establish a connection with a remote modem. The default value is **2**.

stop-bits

Specifies the number of stop bits that follow a byte. The 1.5 option is only available on the 1-port and 2-port models, but not on the modem line (Line 2) of the SDS1M model.

term-type

Specifies the type of terminal connected to the line:

- **Dumb**
- **WYSE60**
- **VT100**
- **ANSI**
- **TVI925**
- **IBM3151TE**
- **VT320** (specifically supporting VT320-7)
- **HP700** (specifically supporting HP700/44)
- **Term1, Term2, Term3** (user-defined terminals)

line-termination

Used with **EIA-422** and **EIA-485** on SDS 8-port+ Device Server models, specifies whether or not the line is terminated; use this option when the line is connected to a device at the end of the EIA network.

internet-address

Used with reverse sessions, users can access serial devices connected to the Device Server by the specified Internet Address (or host name that can be resolved to the Internet Address in a DNS network). You must reboot the Device Server for the **Internet Address** to take affect (the kill line option does not apply to this parameter). This parameter must be in IPv4 format.

Set Line Interface

The SCS and STS Device Server models only support the EIA-232 interface and therefore does not require the **`interface`** parameter, instead you can just set the parameters for the EIA-232 interface.

Description Configures line interface (hardware) parameters.

User Level Admin

Syntax

```
set line .|<number>|* interface eia-232 [monitor-dcd on|off]
 [monitor-dsr on|off] [flow none|soft|hard|both]
 [speed 50|75|110|134|150|200|300|600|1200|1800|2400|4800|9600|
   19200|38400|57600|115200|230400|28800|custom <baud_rate>]

set line .|<number>|* interface eia-422
 [flow none|soft|hard|both]
 [speed 50|75|110|134|150|200|300|600|1200|1800|2400|4800|
   9600|19200|38400|57600|115200|230400|28800|
   custom <baud_rate>]]

set line .|<number>|* interface eia-485-half-duplex
 [tx-driver-control auto|rts] [flow none|soft]
 [echo-suppression on|off]]
 [speed 50|75|110|134|150|200|300|600|1200|1800|2400|4800|
   9600|19200|38400|57600|115200|230400|28800|custom <baud_rate>]

set line .|<number>|* interface eia-485-full-duplex
 [tx-driver-control auto|rts] [flow none|soft]
 [speed 50|75|110|134|150|200|300|600|1200|1800|2400|4800|
   9600|19200|38400|57600|115200|230400|28800|custom <baud_rate>]
```

Options

eia-232 | eia-422 | eia-485-half-duplex|eia-485-full-duplex

Specifies the type of serial line that is being used with the Device Server. Specify either EIA-232, EIA-422, EIA-485-half-duplex, or EIA-485-full-duplex. The SCS models support only EIA-232.

monitor-dcd

Specifies whether the RS-232 signal DCD (Data Carrier Detect) should be monitored. This is used with modems or any other device that sends a DCD signal. When it is monitored and the Device Server detects a DCD signal, the line service is started. Default is **Off**. If both **Monitor DCD** and **Monitor DSR** are enabled, both signals must be detected before the line service is started.

monitor-dsr

Specifies whether the RS-232 signal DSR (data set ready) should be monitored. This is used with modems or any device that sends a DSR signal. When it is monitored and the Device Server detects a DSR signal, the line service is started. Default is **Off**. If both **Monitor DCD** and **Monitor DSR** are enabled, both signals must be detected before the line service is started.

flow

Defines whether the data flow is handled by the software (**Soft**), hardware (**Hard**), **Both**, or **None**. If you are using **SLIP**, set to **Hard** only. If you are using **PPP**, set to either **Soft** or **Hard** (**Hard** is recommended). If you select **Soft** with **PPP**, you must set the **ACCM** parameter when you configure **PPP** for the **Line**.

tx-driver-control

Used with a **EIA-485** serial interface, if your application supports **RTS** (Request To Send), select this option. Otherwise, select **Auto**. Default is **Auto**.

duplex

Specify whether the line is **Full Duplex** (communication both ways at the same time) or **Half Duplex** (communication in one direction at a time).

echo-suppression

This parameter applies only to **EIA-485 Half Duplex** mode. All characters will be echoed to the user and transmitted across the serial ports. Some EIA-485 applications require local echo to be enabled in order to monitor the loopback data to determine that line contention has occurred. If your application cannot handle loopback data, echo suppression should be **On**. The default is echo suppression **Off**.

speed

Specifies the baud rate of the line; keep in mind that speed is affected by the length of the cable. You can also specify a custom baud rate; valid values are 50-230400.

Set Line Service

Description Sets the service for the line. For services that need further configuration, see ***Line Service Commands*** on page 294 to find the Line Service that you want to configure. SSL/TLS can be enabled for the following Line Services: DSLogin, Raw, Bidir, VModem, Server Tunnel, Client Tunnel, Modbus Master, Custom App, and TruePort.

User Level Admin

Syntax

```
set line .|<number>|* service bidir <config_host> <server_port>
 <host_port>

set line .|<number>|* service direct|silent rlogin <config_host>

set line .|<number>|* service direct raw <config_host>
 <host_port>

set line .|<number>|* service silent raw <config_host>
 <host_port>
 [multihost all|backup <config_backup_host> <host_port>|none]

set line .|<number>|* service direct|silent telnet|ssh
 <config_host> [<host_port>]

set line .|<number>|* service reverse raw [multihost on|off]|
 ssh|telnet <server_port>

set line .|<number>|* service client-tunnel <config_host>
 <host_port>

set line .|<number>|* service server-tunnel <server_port>

set line .|<number>|* service dslogin|printer|ppp|slip|udp|
 vmodem|modbus-master|modbus-slave|custom-app|power-management

set line .|<number>|* service trueport client-initiated off
 <config_host> <host_port> [signal-active on|off]
 [multihost all|backup <config_backup_host> <host_port>|none]

set line .|<number>|* service trueport client-initiated on
 <server_port> [signal-active on|off] [multihost on|off]
```

Options

bidir

Allows a bidirectional connection on a port.

<config_host>

The name of the target host.

<server_port>

The Device Server port number.

<host_port>

The port number the target host is listening on for incoming connections.

direct

Direct connections bypass the Device Server, enabling the user to log straight into a specific host. A direct connection is recommended where a user logging in to the Device Server is not required. It is also recommended where multiple sessions are not a requirement. The message **`Press return to continue`** is displayed on the user's screen. The user must press a key to display the host login prompt. The message is redisplayed on logout.

silent

Silent connections are the same as direct connections, except they are permanently established. The host login prompt is displayed on the screen. Logging out redisplays this prompt. Silent connections, unlike direct connections, however, make permanent use of pseudo tty resources and therefore consume host resources even when not in use.

rlogin

Sets the line for a remote login connection.

raw

Creates a connection where no authentication takes place and data is passed unchanged.

telnet

Sets the line for a telnet connection.

ssh

Sets the line for an SSH connection.

reverse

Enables a TCP/IP host to establish a login connection on an external machine attached to a port. For example, to access machines like protocol converters, statistical multiplexors, or machines like routers, firewalls, servers, etc.

client-tunnel

Sets the line for a client tunnel connection.

modbus-slave

Sets the line to act as a Modbus slave.

dslogin

The default connection. The Device Server displays a login on that line. For example, **DSLogin** is used when a System Administrator configures the Device Server, providing authentication of a user before starting a **User Service** of **SLIP**, or users starts a session(s) from the Device Server to hosts.

printer

Using the Device Server as a printer server. For example, remote printing using LPD (port 515) or RCP (port 514).

ppp

Sets the port to a dedicated PPP line.

slip

Sets the port in SLIP mode.

udp

Sets the line to listen for and/or send UDP data.

vmodem

The Device Server port behaves as if it were a modem to the attached device.

server-tunnel

Sets the line for a server tunnel connection.

modbus-slave

Sets the line to act as a Modbus master.

modbus-master

Sets the line to act as a Modbus slave.

custom-app

Sets the line to use the custom application created with the SDK.

power-management

Indicates that there is a power bar connection to this serial line.

trueport

Sets the line to communicate with the TruePort utility. You must install the TruePort utility on the host machine.

client-initiated

When this option is turned on, the Device Server will wait for a connection from the TruePort host (see the TruePort documentation for information on how to set up this feature on the TruePort host). When this option is turned off, the Device Server will initiate the connection to the TruePort host. The default is off.

signal-active

When a TruePort line becomes active, this option has the following impact:

- **TruePort Lite Mode**—When enabled, the EIA-232 signals remain high (active). When disabled, the EIA-232 signals remain low (inactive).
- **TruePort Full Mode**—Same as TruePort Lite Mode, except that when the TruePort client connects to the Device Server TruePort port, the TruePort client application can control the state of the EIA-232 signals.

Default: Enabled

multihost

Used for connections coming from the network to the serial port for TruePort or Raw services, allows multiple hosts to connect to the serial device.

multihost all|backup <*config_backup_host*> <*tcp_port*>|none

Used for connections going from the serial port to the network for TruePort or Silent Raw services, allows the serial device to communicate to either all the hosts in the multihost list or a primary/backup host schema (see ***Configuring Multiple Hosts*** on page 130 for a more detailed explanation).

Set Modem

Description Sets the modem initialization strings.

User Level Admin

Syntax `set modem <modem_name> <init_string>`

Options <***modem_name***>

Predefined modem name.

<***init_string***>

Specify the initialization string for the internal modem. This can be up to 60 characters long, but cannot include spaces.

Set Termtype

Description Sets the terminal type for the current terminal session. term1, term2, and term3 refer to the user-uploadable custom terminal definitions. If these are not present, the default is wyse60.

User Level Restricted, Normal, Admin

Syntax `set termtype [wyse60|vt100|ansi|dumb|tvi925|ibm3151te|vt320|hp700|term1|term2|term3]`

Option **wyse60|vt100|ansi|dumb|tvi925|ibm3151te|vt320|hp700|term1|term2|term3**

Specifies the type of terminal connected to the line:

- **Dumb**
- **WYSE60**
- **VT100**
- **ANSI**
- **TVI925**
- **IBM3151TE**
- **VT320** (specifically supporting VT320-7)
- **HP700** (specifically supporting HP700/44)
- **Term1, Term2, Term3** (user-defined terminals)

Show Line

Description Shows the line settings/information.

User Level Admin

Syntax `show line <number>|*`

Line Service Commands

Set Custom-App

Description You can create a custom application that can run on a specific serial line in Device Server using the Perle SDK.

User Level Admin

Syntax `set custom-app line .|<number>|* program-command-line <command>`

Options **program-command-line**

The name of the SDK program executable that has been already been downloaded to the Device Server, plus any parameters you want to pass to the program. Maximum of 80 characters. Use the **`shell`** CLI command as described in the *SDK Programmer's Guide* to manage the files that you have downloaded to the Device Server. For example, using sample **`outraw`** program, you would type:

```
outraw -s 0 192.168.2.1:10001 Acct:10001
```

if you were starting the application on the Server (notice the **`-s 0`** parameter specifies Line 1).

Set Rlogin-Client

Description Configures remote login parameters.

User Level Normal, Admin

Syntax `set rlogin-client line .|<number>|* termtype <terminal_name>`

Option **termtype**

Type of terminal attached to this line; for example, ansi or wyse60.

Set Telnet-Client

Description Configures telnet parameters.

User Level Normal, Admin

Syntax

```
set telnet-client line .|<number>|* [termtype <terminal_name>]
 [line-mode on|off] [map-cr-crlf on|off] [local-echo on|off]
 [echo <00-7f>] [eof <00-7f>] [erase <00-7f>] [intr <00-7f>]
 [quit <00-7f>] [escape <00-7f]
```

Options

termtype

Type of terminal attached to this line; for example, ANSI or WYSE60.

line-mode

When **On**, keyboard input is not sent to the remote host until **Enter** is pressed, otherwise input is sent every time a key is pressed. Default is **Off**.

map-cr-crlf

Maps carriage returns (CR) to carriage return line feed (CRLF). The default value is **Off**.

local-echo

Toggles between local echo of entered characters and suppressing local echo. Local echo is used for normal processing, while suppressing the echo is convenient for entering text that should not be displayed on the screen, such as passwords. This parameter can only be used when **Line Mode** is **On**. Default is **Off**.

echo

Defines the echo character. When **Line Mode** is **On**, typing the echo character echoes the text locally and sends only completed lines to the host. This value is in hexadecimal with a default value of **5** (ASCII value **^E**).

eof

Defines the end-of-file character. When **Line Mode** is **On**, entering the EOF character as the first character on a line sends the character to the remote host. This value is in hexadecimal with a default value of **4** (ASCII value **^D**).

erase

Defines the erase character. When **Line Mode** is **Off**, typing the erase character erases one character. This value is in hexadecimal with a default value of **8** (ASCII value **^H**).

intr

Defines the interrupt character. Typing the interrupt character interrupts the current process. This value is in hexadecimal with a default value of **3** (ASCII value **^C**).

quit

Defines the quit character. Typing the quit character closes and exits the current telnet session. This value is in hexadecimal with a default value of **1c** (ASCII value **FS**).

escape

Defines the escape character. Returns you to the command line mode. This value is in hexadecimal with a default value of **1d** (ASCII value **GS**).

Set SSH-Client

Description Configures an SSH connection.

User Level Normal, Admin

Syntax

```
set ssh-client line .|<number>|* [termtype <terminal_name>]
 [protocol ssh-1|ssh-2|ssh-2/1] [compression on|off]
 [verbose on|off] [auto-login on|off] [name <string>]
 [password <string>] [ssh-1-cipher 3des|des|blowfish]
 [authentication rsa on|off] [authentication dsa on|off]
 [authentication keyboard-interactive on|off]

set ssh-client line .|<number>|*
 ssh-2-cipher-list <3des blowfish cast aes arcfour>
```

Options

termtype

Type of terminal attached to this line; for example, ANSI or WYSE60.

protocol

Specify the SSH protocol you want to use for the connection, SSH-1, SSH-2, or either, SSH2/1.

compression

Requests compression of all data. Compression is desirable on modem lines and other slow connections, but will only slow down things on fast networks.

verbose

Displays debug messages on the terminal.

auto-login

Creates an automatic SSH login, using the **Name** and **Password** values.

name

The user's name when **Auto Login** is enabled.

password

The user's password when **Auto Login** is enabled.

ssh-1-cipher

Select the encryption method (cipher) that you want to use for your SSH version 1 connection:

- **3DES**
- **Blowfish**

ssh-2-cipher-list

Select the order of negotiation for the encryption method (ciphers) that the Device Server will use for the SSH version 2 connection:

- **3DES**
- **Blowfish**
- **AES**
- **Arcfour**
- **CAST**

authentication rsa

An authentication method used by SSH version 1 and 2. When enabled, an SSH client session will try to authenticate via RSA.

authentication dsa

An authentication method used by SSH version 2. When enabled, an SSH client session will try to authenticate via DSA.

authentication keyboard-interactive

The user types in a password for authentication.

Set PPP

Description Configures the Line's PPP settings.

User Level Admin

Syntax

```
set ppp line .|<number>|*|wireless-wan [accm <8_hex_digits>]
 [address-comp on|off] [auth-tmout <integer>]
 [challenge-interval <integer>] [cr-retry <integer>]
 [cr-timeout <integer>] [ipaddr-neg on|off]
 [ipv6-local-interface <interface_id>]
 [ipv6-remote-interface <interface_id>]
 [lipaddr <IPV4_address>] [magic-neg on|off] [mru <64-1500>]
 [nak-retry <integer>] [netmask <IPV4_address>]
 [password <string>] [proto-comp on|off] [ripaddr <IPV4_address>]
 [roaming-callback on|off] [authentication none|pap|chap]
 [routing none|send|listen|send-and-listen] [rpassword <string>]
 [ruser <string>] [tr-retry <integer>] [tr-tmout <integer>]
 [user <string>] [vj-comp on|off]
```

Options

accm

Specifies the ACCM (Asynchronous Control Character Map) characters that should be escaped from the data stream. This is entered as a 32-bit hexadecimal number with each bit specifying whether or not the corresponding character should be escaped. The bits are specified as the most significant bit first and are numbered 31-0. Thus if bit 17 is set, the 17th character should be escaped, that is, 0x11 (XON). So entering the value 000a0000 will cause the control characters 0x11 (XON) and 0x13 (XOFF) to be escaped on the link, thus allowing the use of XON/XOFF (software) flow control. If you have selected **Soft Flow Control** on the **Line**, you must enter a value of at least **000a0000** for the **ACCM**. The default value is **00000000**, which means no characters will be escaped.

address-comp

This determines whether compression of the **PPP Address** and **Control** fields take place on the link. The default is **On**. For most applications this should be enabled.

auth-tmout

The timeout, in minutes, during which successful PAP or CHAP authentication must take place (when **PAP** or **CHAP** is turned **On**). If the timer expires before the remote end has been authenticated successfully, the link will be terminated.

challenge-interval

The interval, in minutes, for which the Device Server will issue a CHAP re-challenge to the remote end. During CHAP authentication, an initial CHAP challenge takes place, and is unrelated to CHAP re-challenges. The initial challenge takes place even if re-challenges are disabled. Some PPP client software does *not* work with CHAP re-challenges, so you might want to leave the parameter disabled in the Device Server. The default value is **0** (zero), meaning CHAP re-challenge is disabled.

cr-retry

The maximum number of times a **`configure request`** packet will be re-sent before the link is terminated.

cr-timeout

The maximum time, in seconds, that LCP (Link Control Protocol) will wait before it considers a **`configure request`** packet to have been lost.

ipaddr-neg

Specifies whether or not IP address negotiation will take place. IP address negotiation is where the Device Server allows the remote end to specify its IP address. The default value is **Off**. When **On**, the IP address specified by the remote end will be used in preference to the **Remote IP Address** set for a **Line**. When **Off**, the **Remote IP Address** set for the **Line** will be used.

ipv6-local-interface

The local IPv6 interface identifier of the Device Server end of the PPP link. For routing to work, you must enter a local IP address. Choose an address that is part of the same network or subnetwork as the remote end. Do not use the Device Server's (main) IP address in this field; if you do so, routing will not take place correctly. The first 64 bits of the Interface Identifier must be zero, therefore, ::abcd:abcd:abcd:abcd is the expected format.

ipv6-remote-interface

The remote IPv6 interface identifier of the remote end of the PPP link. Choose an address that is part of the same network or subnetwork as the Device Server. If you set the **PPP** parameter **IP Address Negotiation** to **On**, the Device Server will ignore the remote IP address value you enter here and will allow the remote end to specify its IP address. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Interface-ID** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here. The first 64 bits of the Interface Identifier must be zero, therefore, ::abcd:abcd:abcd:abcd is the expected format.

lipaddr

The IPV4 IP address of the Device Server end of the PPP link. For routing to work, you must enter a local IP address. Choose an address that is part of the same network or subnetwork as the remote end; for example, if the remote end is address 192.101.34.146, your local IP address can be 192.101.34.145. Do not use the Device Server's (main) IP address in this field; if you do so, routing will not take place correctly.

magic-neg

Determines if a line is looping back. If enabled (**On**), random numbers are sent on the link. The random numbers should be different, unless the link loops back. The default is **Off**.

mru

The Maximum Receive Unit (MRU) parameter specifies the maximum size of PPP packets that the Device Server's port will accept. Enter a value between 64 and 1500 bytes; for example, 512. The default value is **1500**. If your user is authenticated by the Device Server, the **MRU** value will be overridden if you have set a **Framed MTU** value for the user. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-MTU** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

nak-retry

The maximum number of times a **`configure NAK`** packet will be re-sent before the link is terminated.

netmask

The network subnet mask. For example, 255.255.0.0. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Netmask** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

password

Complete this field only if you have specified **PAP** or **CHAP** (security protocols) in the **Security** field and:

- you wish to dedicate this line to a single remote user, who will be authenticated by the Device Server, *or*
- you are using the Device Server as a router (back-to-back with another Device Server)

Password means the following:

- When **PAP** is specified, this is the password the remote device will use to authenticate the port on this Device Server.
- When **CHAP** is specified, this is the secret (password) known to both ends of the link upon which responses to challenges shall be based.

In either case, you can enter a maximum of 16 alphanumeric characters.

proto-comp

This determines whether compression of the PPP Protocol field takes place on this link. The default is **On**.

ripaddr

The IPV4 IP address of the remote end of the PPP link. Choose an address that is part of the same network or subnetwork as the Device Server. If you set the PPP parameter IP Address Negotiation to On, the Device Server will ignore the remote IP address value you enter here and will allow the remote end to specify its IP address. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Address** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here. The exception to this rule is a **Framed-Address** value in the RADIUS file of **255.255.255.254**; this value allows the Device Server to use the remote IP address value configured here.

roaming-callback

A user can enter a telephone number that the Device Server will use to callback him/her. This feature is particularly useful for a mobile user. Roaming callback can only work when the **User Callback** parameter is set to **On**. Roaming callback therefore overrides (fixed) **User Callback**.To use **Roaming Callback**, the remote end must be a Microsoft Windows OS that supports Microsoft's Callback Control Protocol (CBCP). The user is allowed 30 seconds to enter a telephone number after which the Device Server ends the call. The default is **Off**.

routing

Determines the routing mode (RIP, Routing Information Protocol) used on the **PPP** interface as one of the following options:

- **None**—Disables RIP over the PPP interface.
- **Send**—Sends RIP over the PPP interface.
- **Listen**—Listens for RIP over the PPP interface.
- **Send and Listen**—Sends RIP and listens for RIP over the PPP interface.

This is the same function as the **Framed-Routing** attribute for RADIUS authenticated users. Default is **None**.

rpassword

Complete this field only if you have specified **PAP** or **CHAP** (security protocols) in the **Security** field, *and*

- you wish to dedicate this line to a single remote user, and this user will be authenticated by the Device Server, *or*
- you are using the Device Server as a router (back-to-back with another Device Server)

Remote password means the following:

- When **PAP** is specified, this is the password the Device Server will use to authenticate the remote device.
- When **CHAP** is specified, this is the secret (password) known to both ends of the link upon which responses to challenges will be based.

Remote Password is the opposite of the parameter **Password**. Your Device Server will only authenticate the remote device when **PAP** or **CHAP** is operating. In either case, you can enter a maximum of sixteen alphanumeric characters.

ruser

Complete this field only if you have specified **PAP** or **CHAP** (security protocols) in the **Security** field, *and*

- you wish to dedicate this line to a single remote user, who will be authenticated by the Device Server, *or*
- you are using the Device Server as a router (back-to-back with another Device Server)

When **Connection Method** is set to **In** or **Both**, the **Remote User** is the name the Device Server will use to authenticate the port on the remote device. Your Device Server will only authenticate the port on the remote device when **PAP** or **CHAP** are operating. You can enter a maximum of sixteen alphanumeric characters. When connecting together two networks, enter a dummy user name; for example, DS_SALES.

Note If you want a reasonable level of security, the user name and password should not be similar to a user name or password used regularly to login to the Device Server. This option does not work with external authentication.

authentication

The type of authentication that will be done on the link: **None**, **PAP**, or **CHAP**. The default is **CHAP**. You can use **PAP** or **CHAP** to authenticate a port or user on the Device Server, from a remote location, or authenticate a remote client/device, from the Device Server (not commonly used for **Dial Out**).

PAP is a one time challenge of a client/device requiring that it respond with a valid username and password. A timer operates during which successful authentication must take place. If the timer expires before the remote end has been authenticated successfully, the link will be terminated.

CHAP challenges a client/device at regular intervals to validate itself with a username and a response, based on a hash of the secret (password). A timer operates during which successful authentication must take place. If the timer expires before the remote end has been authenticated successfully, the link will be terminated.

When setting either **PAP** and **CHAP**, make sure the Device Server and the remote client/device have the same setting. For example, if the Device Server is set to **PAP**, but the remote end is set to **CHAP**, the connection will be refused.

tr-retry

The maximum number of times a **`terminate request`** packet will be re-sent before the link is terminated.

tr-tmout

The maximum time, in seconds, that LCP (Link Control Protocol) will wait before it considers a **`terminate request`** packet to have been lost.

user

Complete this field only if you have specified **PAP** or **CHAP** (security protocols) in the **Security** field, *and*

- you wish to dedicate this line to a single remote user, who will be authenticated by the Device Server, *or*
- you are using the Device Server as a router (back-to-back with another Device Server).

When **Connection Method** is set to **Out** or **Both**, the **User** is the name the remote device will use to authenticate a port on this Device Server. The remote device will only authenticate your Device Server's port when **PAP** or **CHAP** are operating. You can enter a maximum of sixteen alphanumeric characters; for example, tracy201. When connecting together two networks, enter a dummy user name; for example, DS_HQ.

Note If you want a reasonable level of security, the user name and password should not be similar to a user name or password used regularly to login to the Device Server. External authentication can not be used for this user.

vj-comp

This determines whether Van Jacobson Compression is used on this link. The default is **On**. If your user is authenticated by the Device Server, this VJ compression value will be overridden if you have set the **User Framed Compression On**. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Compression** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

Set PPP Dynamic-DNS

Description This option is only available when IP address negotiation (**`ipaddr-neg`**) is **`on`**. When enabled, the Device Server will automatically update the DNS server with the specified host name and negotiated IP address for the PPP session.

User Level Admin

Syntax

```
set ppp line .|<number>|* dynamic-dns [on|off]
 [hostname <hostname>] [username <username>]
 [password <password>]
```

Options

hostname

Specify the host name that will be updated with the PPP session's IP address on the DynDNS.org server.

username

Specify the user name used to access the DynDNS.org server.

password

Specify the password used to access the DynDNS.org server.

Set SLIP

Description Configures the SLIP settings.

User Level Admin

Syntax

```
set slip line .|<number>|* [lipaddr <IPV4_address>]
 [mtu <256-1006>] [netmask <IPV4_address>]
 [ripaddr <IPV4_address>] [vj-comp on|off]
 [routing none|send|listen|send-and-listen]
```

Options

lipaddr

The IPv4 address of the Device Server end of the SLIP link. For routing to work you must enter an IP address in this field. Choose an address that is part of the same network or subnetwork as the remote end; for example, if the remote end is address 192.101.34.146, your local IP address can be 192.101.34.145. Do not use the Device Server's (main) IP address in this field; if you do so, routing will not take place correctly.

mtu

The Maximum Transmission Unit (MTU) parameter restricts the size of individual SLIP packets being sent by the Device Server. Enter a value between 256 and 1006 bytes; for example, 512. The default value is **256**. If your user is authenticated by the Device Server, this MTU value will be overridden when you have set a **Framed MTU** value for the user. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-MTU** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

netmask

The network subnet mask. For example, 255.255.0.0. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Netmask** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

ripaddr

The IPv4 address of the remote end of the SLIP link. Choose an address that is part of the same network or subnetwork as the Device Server. If your user is authenticated by the Device Server, this remote IP address will be overridden if you have set a **Framed IP Address** for the user. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Address** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

vj-comp

This determines whether Van Jacobson compression is used on this link; that is, whether you are using SLIP or C-SLIP (compressed SLIP). The choices are **On** (C-SLIP) or **Off** (SLIP). The default is **On**. C-SLIP greatly improves the performance of interactive traffic, such as Telnet or Rlogin.

If your user is authenticated by the Device Server, this VJ compression value will be overridden if you have set a **Framed Compression** value for a user. If your user is authenticated by RADIUS *and* the RADIUS parameter **Framed-Compression** is set in the RADIUS file, the Device Server will use the value in the RADIUS file in preference to the value configured here.

routing

Determines the routing mode (RIP, Routing Information Protocol) used on the **SLIP** interface as one of the following options:

- **None**—Disables RIP over the SLIP interface.
- **Send**—Sends RIP over the SLIP interface.
- **Listen**—Listens for RIP over the SLIP interface.
- **Send and Listen**—Sends RIP and listens for RIP over the SLIP interface.

This is the same function as the **Framed-Routing** attribute for RADIUS authenticated users. Default is **None**.

Set UDP

Description Configures the UDP settings for the serial line.

User Level Normal, Admin

Syntax

```
set udp line .|<number>|* entry 1|2|3|4
 both auto-learn|specific <UDP_port> [<start_IP_address>]
 [<end_IP_address>]

set udp line .|<number>|* entry 1|2|3|4 in
 any-port|auto-learn|specific <UDP_port> [<start_IP_address>]
 [<end_IP_address>]

set udp line .|<number>|* entry 1|2|3|4 out <UDP_port>
 [<start_IP_address>] [<end_IP_address>]

set udp line .|<number>|* entry 1|2|3|4 none
```

Options **both|in|out|none**

The direction in which information is received or relayed:

- **None**—UDP service not enabled.
- **In**—LAN to serial. The Device Server will listen on port value configured in the **DS Port** parameter for messages coming from the learned or configured port.
- **Out**—Serial to LAN. The Device Server will listen on the port value configured in the **DS Port** parameter and will send to the configured port.
- **Both**—Messages are relayed both directions. For messages coming from the LAN to the serial device, Device Server will listen on port value configured in the **DS Port** parameter for messages coming from the learned or configured port. For messages going from the serial device to the LAN, the Device Server will listen on the port value configured in the **DS Port** parameter and will send to the configured or learned (if **Auto-learn** is enabled, the Device Server must receive a UDP message before it can send one, since the port must first be ’learned’) port.

auto-learn

The Device Server will only listen to the first port that it receives a UDP packet from. Applicable when set to **In** or **Both**.

any-port

The Device Server will receive messages from any port sending UDP packets. Applicable when set to **In**.

specific

The port that the Device Server will use to relay messages to servers/hosts. This option works with any setting except **None**. The Device Server will listen for UDP packets on the port configured by the **DS Port** parameter.

<start_IP_address>

The first host IP address in the range of IP addresses (for IPV4 or IPV6) that the Device Server will listen for messages from and/or send messages to.

<end_IP_address>

The last host IP address in the range of IP addresses (for IPV4, not required for IPV6) that the Device Server will listen for messages from and/or send messages to.

Set Vmodem

Description Configures the vmodem settings for the serial line. SSL/TLS can be enabled and configured for this Line Service.

User Level Admin

Syntax

```
set vmodem line .|<number>|* [echo on|off]
 [failure-string <string>] [host <config_host>]
 [init-string <string>] [mode auto|manual]
 [port <TCP_port>|0] [respons-delay <time_ms>]
 [signals dcd always-high|follow-connection]
 [signals dtr always-high|represent-dcd|represent-ri]
 [signals rts always-high|represent-dcd|represent-ri]
 [style numeric|verbose] [success-string <string>]
 [suppress on|off]
```

Options

echo

When enabled, echoes back characters that are typed in (equivalent to ATE0/ATE1 commands). Disabled by default.

failure-string

String that is sent to the serial device when a connection fails. If no string is entered, then the string **NO CARRIER** will be sent.

host

The target host name.

init-string

You can specify additional vmodem commands that will affect how vmodem starts. The following commands are supported: ATQn, ATVn, ATEn, +++ATH, ATA, ATI0, ATI3, ATS0, AT&Z1, AT&Sn, AT&Rn, AT&Cn, AT&F, ATS2, ATS12, ATO (ATD with no phone number), and ATDS1.

See ***VModem Initialisation Commands*** on page 87 for a more detailed explanation of the support initialisation commands.

mode

Auto mode establishes the connection when the line becomes active. You must supply the AT command or phone number that will start the connection; see ***Set Vmodem-Phone*** on page 305 for the command parameters to set the AT command or phone number.

port

The port number the target host is listening on for messages.

response-delay

The amount of time, in milliseconds, before an AT response is sent to the requesting device. The default is 250 ms.

signals dcd

When you specify that the DTR and/or the RTS signal will act as a DCD signal, **`alway-high`** indicates that the signal connection will stay up and **`follow-connection`** indicates that the connection will go up/down depending on the host connection status.

signals dtr

You can specify how the DTR signal pin acts during your modem application connection, as itself (DTR), as DCD, or as RI.

signals rts

You can specify how the RTS signal pin acts during your modem application connection, as itself (RTS), as DCD, or as RI.

style

One of the following:

- **Verbose**—Return codes (strings) are sent to the connected device.
- **Numeric**—The following characters can be sent to the connected device:
  - **1** Successfully Connected
  - **2** Failed to Connect
  - **4** Error

success-string

String that is sent to the serial device when a connection succeeds. If no string is entered, then the string **`CONNECT`** will be sent with the connecting speed, for example **`CONNECT 9600`**.

suppress

When enabled, the connection success/failure indication strings are sent to the connected device, otherwise these indications are suppressed. The default is disabled.

Set Vmodem-Phone

Description Configures the VModem phone number settings. This is a universal command, meaning that all VModem lines will access to the entries defined here. 1-port models support up to 4 entries, all other desktop models support up to 8 entries, and rack-mount models support up to 48 entries.

User Level Admin

Syntax

```
set vmodem-phone entry <number> phone-number <string>
 <IP_address> <TCP_port>

set vmodem-phone entry <number> delete
```

Options

entry

Specify the entry number in the vmodem phone number table.

phone-number

Specify the phone number or AT command that your modem application sends to the modem.

<IP_address>

Specify the IP address of the Device Server that is receiving the vmodem connection.

<TCP_port>

Specify the TCP Port on the Device Server that is set to receive the vmodem connection.

delete

Deletes the specified entry from the phone number table.

Set SSL Line

Description Sets the SSL/TLS parameters for the line. SSL/TLS can be enabled for the following Line Services: DSLogin, Raw, Bidir, VModem, Server Tunnel, Client Tunnel, Modbus Master, and Custom App.

User Level Admin

Syntax

```
set ssl line .|<number>|* [enable on|off] [use-server on|off]
 [version any|tslv1|sslv3] [type client|server]
 [verify-peer on|off] [validation-criteria
 country <code>|state-province <text>|locality <text>
 |organisation <text>|organisation-unit <text>
 |common-name <text>|email <email_addr>]
```

Options

enable

Activates the SSL/TLS settings for the line.

use-server

Uses the SSL/TLS server configuration for the line.

version

Specify whether you want to use:

- **Any**—The Device Server will try a TLSv1 connection first. If that fails, it will try an SSLv3 connection. If that fails, it will try an SSLv2 connection.
- **TLSv1**—The connection will use only TLSv1.
- **SSLv3**—The connection will use only SSLv3.

The default is **Any**.

type

Specify whether the Device Server will act as an SSL/TLS client or server. The default is **Client**.

verify-peer

Enable this option when you want the Validation Criteria to match the Peer Certificate for authentication to pass. If you enable this option, you need to download an SSL/TLS certificate authority (CA) list file to the Device Server.

validation-criteria

Any values that are entered in the validation criteria must match the peer certificate for an SSL connection; any fields left blank will not be validated against the peer certificate.

country

A two character country code; for example, US. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

state-province

Up to a 128 character entry for the state/province; for example, IL. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

locality

Up to a 128 character entry for the location; for example, a city. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

organisation

Up to a 64 character entry for the organisation; for example, Accounting. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

organisation-unit

Up to a 64 character entry for the unit in the organisation; for example, Payroll. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

common-name

Up to a 64 character entry for common name; for example, the host name or fully qualified domain name. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

email

Up to a 64 character entry for an email address; for example, acct@anycompany.com. This field is case sensitive in order to successfully match the information in the peer SSL/TLS certificate.

Set SSL Line Cipher-suite

Description Sets the SSL/TLS cipher suite parameters for the line.

User Level Admin

Syntax

```
set ssl line .|<number>|* cipher-suite
 option1|option2|option3|option4|option5
 encryption any|aes|3des|des|arcfour|arctwo|none
 min-key-size 40|56|64|128|168|256
 max-key-size 40|56|64|128|168|256
 key-exchange any|rsa|edh-rsa|edh-dss|adh
 hmac any|sha1|md5
```

Options **option1**|**option2**|**option3**|**option4**|**option5**

Sets the priority of the cipher suite, with **`option1`** being highest priority and **`option5`** lowest priority.

encryption

Select the type of encryption that will be used for the SSL connection:

- Any—Will use the first encryption format that can be negotiated.
- AES
- 3DES
- DES
- ARCFOUR
- ARCTWO
- None—Removes any values defined for the cipher option.

The default value is **Any**.

min-key-size

The minimum key size value that will be used for the specified encryption type. The default is **40**.

max-key-size

The maximum key size value that will be used for the specified encryption type. The default is **256**.

key-exchange

The type of key to exchange for the encryption format:

- **Any**—Any key exchange that is valid is used (this does not, however, include ADH keys).
- **RSA**—This is an RSA key exchange using an RSA key and certificate.
- **EDH-RSA**—This is an EDH key exchange using an RSA key and certificate.
- **EDH-DSS**—This is an EDH key exchange using a DSA key and certificate.
- **ADH**—This is an anonymous key exchange which does not require a private key or certificate. Choose this key if you do not want to authenticate the peer device, but you want the data encrypted on the SSL/TLS connection.

The default is **Any**.

hmac

Select the key-hashing for message authentication method for your encryption type:

- Any
- MD5
- SHA1

The default is **Any**.

Set Modbus-Slave Line

Description Sets the Modbus slave parameters for the line. SSL/TLS can be enabled and configured for this Line Service.

User Level Admin

Syntax

```
set modbus-slave line .|<number>|* [crlf on|off]
 [protocol rtu|ascii] [uid-range <uid_range>]
```

Options

crlf

When **Modbus/ASCII** is selected, adds a CR/LF to the end of the transmission; most Modbus devices require this option. The default is **On**.

protocol

Specify the protocol that is used between the Modbus Master(s) and Modbus Slave(s), either RTU or ASCII.

uid-range

You can specify a range of UIDs (1-247), in addition to individual UIDs. The format is comma delimited; for example, 2-35, 50, 100-103.

Set Modbus-Master Line

Description Sets the Modbus master parameters for the line. SSL/TLS can be enabled and configured for this Line Service.

User Level Admin

Syntax

```
set modbus-master line .|<number>|* [crlf on|off]
 [protocol rtu|ascii]
 [[entry <number> [port <port>] [protocol udp|tcp]
  [range-mode gateway|host] [slave-ip <IP_address>]
  [uid-range <start_uid> <end_uid>]]
```

Options

crlf

When **Modbus/ASCII** is selected, adds a CR/LF to the end of the transmission; most Modbus devices require this option. The default is **On**.

protocol

Specify the protocol that is used between the Modbus Master(s) and Modbus Slave(s), either RTU or ASCII.

entry

You can specify up to 16 Modbus Slave Remote IP Mapping entries (the UIDs must not overlap).

port

The destination port of the remote Modbus TCP Slave that the Device Server will connect to.

protocol

Specify the protocol that is used between the Modbus Master and Modbus Slave(s), either TCP or UDP.

range-mode

If you specify **Host**, the IP address is used for the first UID specified in the range. The last octect in the IPv4 address is then incremented for subsequent UID's in that range. The **Host** option is not applicable for IPv6 addresses. If you specify **Gateway**, the Modbus Master Gateway will use the same IP address when connecting to all the remote Modbus slaves in the specified UID range.

slave-ip

The IP address of the TCP/Ethernet Modbus Slave.

uid-range

When **Range Mode** is **Host** and you have sequential Modbus Slave IP addresses (for example, 10.10.10.1, 10.10.10.2, 10.10.10.3, etc.), you can specify a UID range and the Device Server will automatically increment the last digit of the configured IP address. Therefore, you can specify a UID range of 1-100, and the Device Server will route Master Modbus messages to all Modbus Slaves with IP addresses of 10.10.10.1 - 10.10.10.100.

Set Power-Management Line

Description Configures the power management settings for the line.

User Level Admin

Syntax

```
set power-management line .|<number>|*
 [model rps820|rps830|rps1620|rps1630] [name <bar_name>]

set power-management line .|<number>|* plug <1-8|1-16>
 [default-state on|off] [name <plug_name>]
 [power-up-interval .5|1|2|5|15|30|60|120|180|300]
 [serial-line <number>]
```

Options

model

Specify the power bar model, either RPS820, RPS830, RPS1620, RPS1630.

name (power bar name)

Specify a name for the RPS.

plug

Specify the power bar plug number you are configuring.

default-state

Sets the default state of the plug, either **on** or **off**. The default is **off**.

name (plug name)

Specify a name for the plug to make it easier to recognize and manage.

power-up-interval

Specify the amount of time, in seconds, that the power bar will wait before powering up a plug. This can be useful if you have peripherials that need to be started in a specific order. The default is .5 seconds.

serial-line

Associate a serial line(s) connected to a serial device that is plugged into the power bar on that plug.

Set Multihost Line

Description Configures multiple hosts or a primary/backup host schema for Silent Raw, Reverse Raw, or Client-Inititated TruePort service types (multihost must be enabled by the line service type for this to take effect, see ***Set Line Service*** on page 291 for the command to enable multihost).

User Level Admin

Syntax

```
set multihost line <number> entry <number> host <host> <TCP_port>

set multihost line <number> entry <number> delete
```

Options

entry

You can specify up to 49 hosts in the multihost table.

host <***host***>

Specify the preconfigured host that will be in the multihost list.

<***TCP_port***>

Specify the TCP port that the Device Server will use to communicate to the **Host**.

delete

Deletes the specified entry from the multihost table.

Set Line Initiate-Connection

Description Determines how the connection is initiated for Direct Telnet, Direct SSH, Direct Raw, and Direct Rlogin.

User Level Admin

Syntax

```
set line <number>|* initiate-connection
 any-char|specific-char <hex>
```

Options

any-char

Initiates a Direct connection to the specified host when any data is received by the serial port.

specific-char <***hex***>

Initiates a Direct connection to the specified host only when the specified character is received by the serial port.

Show Custom-App

Description Shows the custom application line settings.

User Level Admin

Syntax

```
show custom-app line .|<number>|*
```

Show Interface

Description Shows the network interface information.

User Level Admin

Syntax

```
show interface [brief|ppp|slip|ethernet]
```

Show Power-Management

Description Shows the power management settings for a line.

User Level Admin

Syntax

```
show power-management line <number>
```

Show PPP

Description Shows the PPP line settings.

User Level Admin

Syntax

```
show ppp line <number>|wireless-wan
```

Show Rlogin-Client

Description Show the rlogin-client settings for the line.
User Level Normal, Admin
Syntax `show rlogin-client line <number>`

Show SLIP

Description Show the SLIP settings for the line.
User Level Admin
Syntax `show slip line <number>`

Show SSH-Client

Description Shows the SSH client settings for the line.
User Level Admin
Syntax `show ssh-client line <number>`

Show Telnet-Client

Description Shows the telnet client settings for a line.
User Level Admin
Syntax `show telnet-client line <number>`

Show Modbus

Description Shows the Modbus settings for a line.
User Level Admin
Syntax `show modbus master|slave <number>`

Show UDP

Description Shows the UDP settings for the line.
User Level Admin
Syntax `show udp line <number>`

Show Vmodem

Description Show the vmodem settings for the line.
User Level Normal, Admin
Syntax `show vmodem line <number>`

Show Vmodem-Phone

Description Show the vmodem-phone entries.
User Level Normal, Admin
Syntax `show vmodem-phone`

Modem Commands

Add Modem

Description Adds a modem.
User Level Admin
Syntax **`add modem <`***`modem_name`***`> <`***`initialization_string`***`>`**

Options <***modem_name***>

The name of the modem. Do not use spaces.

<***initialization_string***>

The initialisation string of the modem; see your modem's documentation.

Delete Modem

Description Deletes a modem.
User Level Admin
Syntax **`delete modem <`***`config_modem_name`***`>`**

Option <***config_modem_name***>

You can see a the list of modems that can be deleted by typing **`delete modem ?`**.

Set Modem

Description Sets the modem initialization string for the internal modem in the SCS M series models.
User Level Admin
Syntax **`set modem <`***`modem_name`***`> <`***`init_string`***`>`**

Options <***modem_name***>

Predefined modem name.

<***init_string***>

Specify the initialization string for the internal modem.

Show Modems

Description Shows the Device Server modem table.
User Level Normal, Admin
Syntax **`show modems`**

Email Commands

Set Email-Alert Line

Description This command configures email alert parameters for the line.

User Level Admin

Syntax

```
set email-alert line <number>|* [from <email_addr>]
 [level emergency|alert|critical|error|warning|notice|info|debug]
 [mode on|off] [to <email_addr>] [reply-to <email_addr>]
 [smtp-host <string>] [subject <string>] [use-server on|off]
```

Options

from

This field can contain an email address that might identify the Device Server name or some other value.

level

Choose the event level that triggers an email notification:

- **Emergency**
- **Alert**
- **Critical**
- **Error**
- **Warning**
- **Notice**
- **Info**
- **Debug**

You are selecting the lowest notification level; therefore, when you select **Debug**, you will get an email notification for all events that trigger a message.

mode

Determines whether or not email notification is turned on. Default is **Off**.

to

An email address or list of email addresses that will receive the email notification.

reply-to

The email address to whom all replies to the email notification should go.

smtp-host

The SMTP host (email server) that will process the email notification request. This can be either a host name defined in the Device Server host table or the SMTP host IP address.

subject

A text string, which can contain spaces, that will display in the **Subject** field of the email notification.

use-server

Determines whether you want the **Line** to inherit the **Email Alert** settings from the **Server Email Alert**. If this is enabled, **Server** and **Line** notification events will have the same **Email Alert** setting.

Show Email-Alert Line

Description Shows how the line email alert is configured.

User Level Admin

Syntax

```
show email-alert line <number>
```

Packet Forwarding Commands

Set Packet-Forwarding Line

Description The Packet Forwarding feature allows you to control how the data coming from a serial device is packetized before forwarding the packet onto the LAN network. This command configures packet forwarding options for serial devices attached to the serial line. The command is broken up into logical flows that can be configured; if you configure both the packet options and the frame definition options, the frame definition options will take precedence. If any of the packet options that are configured are met, the packet transmission is triggered.

User Level Admin

Syntax

```
set packet-forwarding line <number>|* mode minimize-latency

set packet-forwarding line <number>|* mode
 optimize-network-throughput

set packet-forwarding line <number>|* mode
 prevent-message-fragmentation delay-between-messages <0-65535>

set packet-forwarding line <number>|*
 mode custom-on-specific-events [enable-end-trigger1 on|off]
 [enable-end-trigger2 on|off] [end-trigger1 <0x0-FF>]
 [end-trigger2 <0x0-FF>] [force-transmit-timer <number>]
 [forwarding-rule trigger1|trigger+1|trigger+2|strip-trigger]
 [idle-timer <number>] [packet-size <number>]

set packet-forwarding line <number>|*
 mode custom-on-fram-definition [enable-eof1 on|off]
 [enable-eof2 on|off] [enable-sof1 on|off] [enable-sof2 on|off]
 [eof1 <0x0-FF>] [eof2 <0x0-FF>]
 [forwarding-rule trigger|trigger+1|trigger+2|strip-trigger]
 [sof1 <0x0-FF>] [sof2 <0x0-FF>] [start-frame-transmit on|off]
```

Options

minimize-latency

This option ensures that all application data is immediately forwarded to the serial device. Select this option for timing-sensitive applications.

optimize-network-throughput

This option provides optimal network usage while ensuring that the application performance is not compromised. Select this option when you want to minimize overall packet count, such as when the connection is over a WAN.

prevent-message-fragmentation

This option detects the message, packet, or data blocking characteristics of the serial data and preserves it throughout the communication. Select this option for message-based applications or serial devices that are sensitive to inter-character delays within these messages.

delay-between-messages

The minimum time, in milliseconds, between messages that must pass before the data is forwarded by the Device Server. The range is 0-65535. The default is 250 ms.

custom-on-specific-events

This section allows you to set a variety of packet definition options. The first criteria that is met causes the packet to be transmitted. For example, if you set a **Force Transmit Timer** of **1000** ms and a **Packet Size** of **100** bytes, whichever criteria is met first is what will cause the packet to be transmitted.

custom-on-frame-definition

This section allows you to control the frame that is transmitted by defining the start and end of frame character(s). If the internal buffer (1024 bytes) is full before the EOF character(s) are received, the packet will be transmitted and the EOF character(s) search will continue. The default frame definition is SOF=00 and EOF=00.

enable-end-trigger1

Enable or disable the end trigger1 hex character.

enable-end-trigger2

Enable or disable the end trigger2 hex character.

enable-end-eof1

Enable or disable the eof1 (end of frame) hex character.

enable-end-eof2

Enable or disable the eof2 (end of frame) hex character.

enable-end-sof1

Enable or disable the sof1 (start of frame) hex character.

enable-end-sof2

Enable or disable the sof2 (start of frame) hex character.

end-trigger1

When enabled, specifies the character that when received will define when the packet is ready for transmission. The actual transmission of the packet is based on the Trigger Forwarding Rule. Valid values are in hex 0-FF. The default is 0.

end-trigger2

When enabled, creates a sequence of characters that must be received to specify when the packet is ready for transmission (if the End Trigger1 character is not immediately followed by the End Trigger2 character, the Device Server waits for another End Trigger1 character to start the End Trigger1/End Trigger2 character sequence). The actual transmission of the packet is based on the Trigger Forwarding Rule. Valid values are in hex 0-FF. The default is 0.

eof1

Specifies the End of Frame character, which defines when the frame is ready to be transmitted. The actual transmission of the frame is based on the Trigger Forwarding Rule. Valid values are in hex 0-FF. The default is 0.

eof2

When enabled, creates a sequence of characters that must be received to define the end of the frame (if the EOF1 character is not immediately followed by the EOF2 character, the Device Server waits for another EOF1 character to start the EOF1/EOF2 character sequence), which defines when the frame is ready to be transmitted. The actual transmission of the frame is based on the Trigger Forwarding Rule. Valid values are in hex 0-FF. The default is 0.

force-transmit-timer

When the specified amount of time, in milliseconds, elapses after the first character is received from the serial port sender, the packet is transmitted. A value of zero (0) ignores this parameter. Valid values are 0-65535 ms. The default is 0.

forwarding-rule

Determines what is included in the Frame (based on the EOF1 or EOF1/EOF2) or Packet (based on Trigger1 or Trigger1/Trigger2). Choose one of the following options:

- **Strip-Trigger**—Strips out the EOF1, EOF1/EOF2, Trigger1, or Trigger1/Trigger2, depending on your settings.
- **Trigger**—Includes the EOF1, EOF1/EOF2, Trigger1, or Trigger1/Trigger2, depending on your settings.
- **Trigger+1**—Includes the EOF1, EOF1/EOF2, Trigger1, or Trigger1/Trigger2, depending on your settings, plus the first byte that follows the trigger.
- **Trigger+2**—Includes the EOF1, EOF1/EOF2, Trigger1, or Trigger1/Trigger2, depending on your settings, plus the next two bytes received after the trigger.

idle-timer

The amount of time, in milliseconds, that must elapse between characters before the packet is transmitted to the network. A value of zero (0) ignores this parameter. Valid values are 0-65535 ms. The default is 0.

packet-size

The number of byte that must be received from the serial port before the packet is transmitted to the network. A value of zero (0) ignores this parameter. Valid values are 0-1024 bytes. The default is 0.

sof1

When enabled, the Start of Frame character defines the first character of the frame, any character(s) received before the Start of Frame character is ignored. Valid values are in hex 0-FF. The default is 0.

sof2

When enabled, creates a sequence of characters that must be received to create the start of the frame (if the SOF1 character is not immediately followed by the SOF2 character, the Device Server waits for another SOF1 character to start the SOF1/SOF2 character sequence). Valid values are in hex 0-FF. The default is 0.

start-frame-transmit

When enabled, the SOF1 or SOF1/SOF2 characters will be transmitted with the frame. If not enabled, the SOF1 or SOF1/SOF2 characters will be stripped from the transmission.

Show Packet-Forwarding Line

Description	Shows the packet-forwarding settings for the line.
User Level	Admin
Syntax	**`show packet-forwarding line <`*`number`*`>`**

Network Commands

SNMP Commands

The Device Server supports SNMP traps restart and SNMP community authentication error.

Add Community

Description Adds an SNMP community (version 1 and version 2).
User Level Admin
Syntax **`add community <`***`community_name`***`> <`***`config_host`***`>|<`***`IP_address`***`> none|readonly|readwrite`**

Options <***community_name***>

The name of the group that devices and management stations running SNMP belong to.

<***config_host***>|<***IP_address***>

The host name of the SNMP community that will send requests to the Device Server.

The IP address of the SNMP manager that will send requests to the Device Server. If the address is **`0.0.0.0`**, any SNMP manager with the **Community Name** can access the Device Server.

none|readonly|readwrite

Permits the Device Server to respond to SNMP requests by:

- **None**—There is no response to requests from SNMP.
- **Readonly**—Responds only to Read requests from SNMP.
- **Readwrite**—Responds to both Read and Write requests from SNMP.

Add Trap

Description Adds an SNMP trap.
User Level Admin
Syntax **`add trap <`***`trap_name`***`> <`***`config_host`***`>|<`***`IP_address`***`>`**

Options <***trap_name***>

The trap receiver is the network management system (NMS) that should receive the SNMP traps. This NMS must have the same SNMP community string as the trap sender.

<***config_host***>|<***IP_address***>

Defines the hosts (by IP address) that will receive trap messages generated by the Device Server. Up to four trap hosts can be defined.

Delete Community

Description Deletes an SNMP community (version 1 and version 2).
User Level Admin
Syntax **`delete community <`***`config_community_number`***`>`**

Option <***config_community_number***>

When you add an SNMP community, it gets assigned to a number. To delete the SNMP community, you need to specify the number of the community that you want to delete. To see which community is assigned to what number, type the **`show snmp`** command.

Delete Trap

Description Deletes an SNMP trap.
User Level Admin
Syntax `delete trap <config_trap_number>`

Option *<config_trap_number>*

When you add an SNMP trap, it gets assigned to a number. To delete the SNMP trap, you need to specify the number of the trap that you want to delete. To see which trap is assigned to what number, type the **`show snmp`** command.

Set SNMP

Description Configures SNMP settings.
User Level Admin
Syntax `set snmp [contact <string>] [location <string>] [readonly user <username>] [readwrite user <username>]`

Options **contact**

The name and contract information of the person who manages this SMNP node.

location

The physical location of the SNMP node.

readonly user

(SNMP version 3) Specified user can only view SNMP variables.

readwrite user

(SNMP version 3) Specified user can view and edit SNMP variables.

Show SNMP

Description Shows SNMP settings, including communities and traps.
User Level Admin
Syntax `show snmp`

TFTP Commands

Set Server TFTP

Description Configures the Device Server's TFTP client settings.
User Level Admin
Syntax `set server tftp [retry <integer>] [timeout <integer>]`

Options **retry**

The number of times the Device Server will retry to transmit a TPFT packet to/from a host when no response is received. Enter a value between 0 and 5. The default is **5**. A value of **0** (zero) means that the Device Server will not attempt a retry should TFTP fail.

timeout

The time, in seconds, that the Device Server will wait for a successful transmit or receipt of TFTP packets before retrying a TFTP transfer. Enter a value between 3 and 10. The default is **3** seconds.

Hosts Commands

Add Host

Description Adds a host to the Device Server host table.
User Level Admin
Syntax **`add host <`*`hostname`*`> <`*`IP_address`*`>`**

`add host <`*`config_host`*`> fqdn <`*`text`*`>`

Options <***hostname***>

The name of the host.

<***IP_address***>

The host IP address.

fqdn

When you have DNS defined in the Device Server, you can enter a DNS resolvable fully qualified domain name (note: FQDN's are excluded as accessible hosts when **IP Filtering** is enabled).

Delete Host

Description Deletes a host from the Device Server host table.
User Level Admin
Syntax **`delete host <`*`config_host`*`>`**

Option <***config_host***>

You can see a list of hosts that can be deleted by typing **`delete host ?`**.

Set Host

Description Configures a host in the Device Server host table.
User Level Admin
Syntax **`set host <`*`config_host`*`> <`*`IP_address`*`>`**

`set host <`*`config_host`*`> fqdn <`*`text`*`>`

Options <***config_host***>

The name of the host.

<***IP_address***>

The host IP address.

fqdn

When you have DNS defined in the Device Server, you can enter a DNS resolvable fully qualified domain name (note: FQDN's are excluded as accessible hosts when **IP Filtering** is enabled).

Show Hosts

Description Shows the Device Server host table.
User Level Normal, Admin
Syntax **`show hosts`**

DNS/WINS Commands

Add DNS

Description Adds a DNS entry.
User Level Admin
Syntax **`add dns <`*`IP_address`*`>`**

Option *`<IP_address>`*

You can specify the IP addresses for up to four DNS (Domain Name Servers) hosts in your network.

Add WINS

Description Adds a WINS entry.
User Level Admin
Syntax **`add wins <`*`IP_address`*`>`**

Option *`<IP_address>`*

You can specify the IP addresses for up to four WINS (Windows Internet Naming Service) hosts in your network.

Delete DNS

Description Deletes a DNS entry.
User Level Admin
Syntax **`delete dns <`*`config_dns_addr`*`>`**

Option *`<config_dns_addr>`*

You can view a list of configured DNS server IP addresses to choose from by typing **`delete dns ?`**.

Delete WINS

Description Deletes a WINS entry.
User Level Admin
Syntax **`delete wins <`*`config_wins_addr`*`>`**

Option *`<config_wins_addr>`*

You can view a list of configured WINS server IP addresses to choose from by typing **`delete wins ?`**.

Show DNS

Description Shows all DNS entries, even those supplied by DHCP/BOOTP when applicable.
User Level Admin, Normal
Syntax **`show dns`**

Show Server

Description Shows the server configuration, including configured WINS or DNS servers.
User Level Admin, Normal
Syntax **`show server`**

Show WINS

Description Shows all WINS entries, even those supplied by DHCP/BOOTP when applicable.
User Level Admin, Normal
Syntax **`show wins`**

Gateway Commands

Add Gateway

Description Adds a gateway. You can configure up to twenty gateways.
User Level Admin
Syntax

```
add gateway <config_host> default

add gateway <config_host> host <dest_IP_addr>

add gateway <config_host> network
 <dest_IPv4_addr>|<dest_IPv6_addr>
 [<subnet_bits_0-32>|<subnet_bits_0-128>]
```

Options <***config_host***>

You can specify up to 20 hosts on desktop models and 49 hosts on rack mount models to act as gateways in your network. Each gateway host must be defined in the Device Server's host table.

default|host|network

Specify the type of gateway:

- **Default**—A gateway which provides general access beyond your local network.
- **Host**—A gateway reserved for accessing a specific host external to your local network.
- **Network**—A gateway reserved for accessing a specific network external to your local network.

<***dest_IP_addr***>

When the gateway is a **Host** or **Network** gateway, you must specify the IP address of the target host machine/network.

<***subnet_bits***>

When the gateway is a **Network** gateway, you must specify the network's subnet mask.

Delete Gateway

Description Deletes a gateway.
User Level Admin
Syntax

```
delete gateway <config_gateway_host>
```

Option <***config_gateway_host***>

You can view the configured gateways that can be deleted by typing **`delete gateway ?`**.

Set Gateway

Description Configures the gateway.
User Level Admin
Syntax

```
set gateway <config_gateway_host> default

set gateway <config_gateway_host> host <destination_ip>

set gateway <config_gateway_host>
 network <dest_IPv4_addr>|<dest_IPv6_address> <prefixbits_mask>
```

Options *<config_gateway_host>*

You can view the configured gateways that can be deleted by typing **`delete gateway ?`**.

default|host|network

Specify the type of gateway:

- **Default**—A gateway which provides general access beyond your local network.
- **Host**—A gateway reserved for accessing a specific host external to your local network.
- **Network**—A gateway reserved for accessing a specific network external to your local network.

<destination_ip>

When the gateway is a **Host** or **Network** gateway, you must specify the IP address of the target host machine/network.

<prefixbits_mask>

When the gateway is a **Network** gateway, you must specify the network's subnet mask for an IPv4 destination IP address (the address is in the form of 123.123.123.123) or prefix bits for an IPv6 destination IP address (valid values are 0-128).

Show Gateways

Description Shows configured gateways.
User Level Normal, Admin
Syntax **`show gateways`**

Logging Commands

Set Syslog

Description Configures the system log.
User Level Admin
Syntax

```
set syslog
 [level emergency|alert|critical|error|warning|notice|info|debug]
 [primary-host <config_host>] [secondary-host <config_host>]
```

Options **level**

Choose the event level that triggers a syslog entry:

- **Emergency**
- **Alert**
- **Critical**
- **Error**
- **Warning**
- **Notice**
- **Info**
- **Debug**

When you select a **Level**, all the levels that appear above it in the list also trigger a syslog entry. For example, if you select **Error**, all **Error**, **Critical**, **Alert**, and **Emergency** events will be logged.

primary-host

The first preconfigured host that the Device Server will attempt to send system log messages to; messages will be displayed on the host's monitor.

secondary-host

If the Device Server cannot communicate with the primary host, then the Device Server will attempt to send system log messages to this preconfigured host; messages will be displayed on the host's monitor.

Show Syslog

Description Shows the syslog settings.
User Level Admin
Syntax `show syslog`

RIP Commands

Add RIP

Description Adds a RIP MD5 key. After pressing **Enter**, you will be prompted for the MD5 key value.

User Level Admin

Syntax **`add rip md5 <`*`integer_md5_id`*`> <`*`start_date`*`> <`*`start_time`*`> <`*`end_date`*`> <`*`end_time`*`>`**

Options *`<integer_md5_id>`*

The **MD5** identification key.

`<start_date>`

The start date that the MD5 key becomes valid. The date format is dependent on your system's settings.

`<start_time>`

The time that the MD5 key becomes valid. The time format is dependent on your system's settings.

`<end_date>`

The last day that the MD5 key is valid. The date format is dependent on your system's settings.

`<end_time>`

The time that the MD5 key becomes invalid. The time format is dependent on your system's settings.

Delete RIP

Description Deletes a RIP MD5 key.

User Level Admin

Syntax **`delete rip md5 <`*`integer_md5_id`*`>`**

Option *`<integer_md5_id>`*

You can see a list of MD5 IDs available for deletion by typing **`delete rip md5 ?`**.

Set RIP

Description Configures the RIP MD5 key. After pressing Enter, you will be prompted for the MD5 key value.

User Level Admin

Syntax

```
set rip [authentication none|password|md5]
 [ethernet-mode none|send|listen|send-and-listen]

set rip password

set rip md5 <config_md5_id> [end <date> <time>]
 [start <date> <time>] [key]
```

Options **authentication**

Specify the type of RIP authentication:

- **None**—No authentication for RIP.
- **Password**—Simple RIP password authentication.
- **MD5**—Use MD5 RIP authentication.

ethernet-mode

Enable/disable RIP (Routing Information Protocol) mode for the Ethernet interface with one of the following options:

- **None**—Disables RIP over the Ethernet interface.
- **Send**—Sends RIP over the Ethernet interface.
- **Listen**—Listens for RIP over the Ethernet interface.
- **Send and Listen**—Sends RIP and listens for RIP over the Ethernet interface.

password

When you type the **`set rip password`** command and press **Enter**, you will be prompted to type in a password and then re-enter that password.

<configured_md5_id>

The **MD5** identification key.

end *<date>* *<time>*

The last day that the MD5 key is valid. Specify as ***dd/mm/yyyy***.

The time that the MD5 key becomes invalid. Specify as ***hh*:*mm*:[*ss*]**.

start *<date>* *<time>*

The start date that the MD5 key becomes valid. Specify as ***dd/mm/yyyy***.

The time that the MD5 key becomes valid. Specify as ***hh*:*mm*:[*ss*]**.

key

When you press **Enter** after typing the **`key`** command, you will be prompted to enter the MD5 key value and then re-enter the key value.

Show RIP

Description Shows the RIP settings.

User Level Normal, Admin

Syntax **`show rip`**

Show RIP Peers

Description Shows current information about IPv4 or IPv6 RIP peers.

User Level Normal, Admin

Syntax **`show rip peers [ipv6]`**

Time Commands

Server Commands

Set Time

Description Sets the Device Server's system clock.
User Level Admin
Syntax `set time <hh:mm[:ss]>`

Option <***hh***:***mm***[:***ss***]>

Sets the Device Server's system time, using the 24-hour clock time format (00:00-23:59).

Set Timezone

Description Sets the Device Server's time zone name and its offset from Greenwich Mean Time (UTC).
User Level Admin
Syntax `set timezone [name <string>] [offset +|-<hh[:mm]>]`

Options <*name*>

The name of the time zone to be displayed during standard time. Maximum 4 characters and minimum 3 characters (do not use angled brackets < >).

offset

The offset from UTC for your local time zone. Specify in the format of hours *hh* (valid -12 to +14) and minutes *mm* (valid 0 to 59 minutes) for the offset from UTC.

Show Time

Description Shows the Device Server's system clock.
User Level Normal, Admin
Syntax `show time`

Show Timezone

Description Shows the time zone settings.
User Level Admin
Syntax `show timezone`

SNTP Commands

Add SNTP

Description Adds an SNTP server.
User Level Admin
Syntax **`add sntp [server-1 <`*`config_host`*`>] [server-2 <`*`config_host`*`>]`**

Options **server-1**

The name of the primary SNTP server from the Device Server host table. Valid with **Unicast** and **Multicast** modes, although in **Multicast** mode, the Device Server will only accept broadcasts from the specified host SNTP server.

server-2

The name of the secondary SNTP server from the Device Server host table. Valid with **Unicast** and **Multicast** modes, although in **Multicast** mode, the Device Server will only accept broadcasts from the specified host SNTP server.

Delete SNTP

Description Deletes an SNTP server.
User Level Admin
Syntax **`delete sntp server-1|server-2`**

Options **server-1**

The name of the primary SNTP server from the Device Server host table. Valid with **Unicast** and **Multicast** modes, although in **Multicast** mode, the Device Server will only accept broadcasts from the specified host SNTP server.

server-2

The name of the secondary SNTP server from the Device Server host table. Valid with **Unicast** and **Multicast** modes, although in **Multicast** mode, the Device Server will only accept broadcasts from the specified host SNTP server.

Set SNTP

Description Configures an SNTP server.
User Level Admin
Syntax

```
set sntp mode none|unicast|anycast|multicast
 [server-1 <config_host>] [server-2 <config_host>]
 [version 1|2|3|4]
```

Options **mode**

The SNTP mode. Valid modes are:

- **None**—SNTP is turned off.
- **Unicast**—Sends a request packet periodically to the Primary host. If communication with the Primary host fails, the request will be sent to the Secondary host.
- **Multicast**—Listen for any broadcasts from an SNTP server and then synchronizes its internal clock to the message.
- **Anycast**—Sends a request packet as a broadcast on the LAN to get a response from any SNTP server. The first response that is received is used to synchronize its internal clock and then operates in **Unicast** mode with that SNTP server.

server-1

The name of the primary SNTP server from the Device Server host table. Valid with **Unicast** and **Multicast** modes, although in **Multicast** mode, the Device Server will only accept broadcasts from the specified host SNTP server.

server-2

The name of the secondary SNTP server from the Device Server host table. Valid with **Unicast** and **Multicast** modes, although in **Multicast** mode, the Device Server will only accept broadcasts from the specified host SNTP server.

version

Version of SNTP. Valid values are 1 to 4. Default value is **4**.

Show SNTP

Description Shows the SNTP settings.
User Level Admin
Syntax `show sntp`

Show SNTP-Info

Description Shows current SNTP information.
User Level Admin
Syntax `show sntp-info`

Time/Date Setting Commands

Set Date

Description Sets the Device Server's system clock.
User Level Admin
Syntax `set date <dd/mm/yyyy>`

Set Summertime

Description Sets the summertime clock.
User Level Admin
Syntax `set summertime [mode none|fixed|recurring] [name <text>] [offset <hh:mm>]`

Options **mode**

You can configure the summer time to take effect:

- **None**—No summer time change.
- **Fixed**—The summer time change goes into effect at the specified time every year. For example, April 15 at 1:00 pm.
- **Recurring**—The summer time changes goes into effect every year at same relative time. For example, on the third week in April on a Tuesday at 1:00 pm.

<name>

The name of the configured summer time zone; this will be displayed during the summer time setting. Maximum 4 characters and minimum 3 characters (do not use angled brackets < >). If this parameter is not set, then the summertime feature will not work.

offset

The offset from UTC for your local time zone. Specify in the format of hours *hh* (valid -12 to +14) and minutes *mm* (valid 0 to 59 minutes) for the offset from UTC.

Set Summertime Fixed

Description Sets the summertime clock to start on the same date each year, for example, April 15 at 1:00 pm.
User Level Admin
Syntax
```
set summertime fixed
 [start-date january|february |...  <0-31>] [start-time <hh:mm>]
 [end-date january|february|... <0-31>] [end-time <hh:mm>]
```

Options **start-date**

The date to change to summer time and end standard time.

start-time *<hh:mm>*

The time to change to summertime. Valid values are 00:00 to 23:59.

end-date

The date to end summer time and start standard time.

end-time *<hh:mm>*

The time to change to standard time. Valid values are 00:00 to 23:59.

Set Summertime Recurring

Description Sets the summertime clock to start at the same relative time each year; for example, on the third week in April on a Tuesday at 1:00 pm.

User Level Admin

Syntax

```
set summertime recurring [start-day monday|tuesday|...]
 [start-month january|february|...] [start-time <hh:mm>]
 [start-week 1|2|3|4|5|last] [end-day monday|tuesday|...]
 [end-month january|february|...] [end-time <hh:mm>]
 [end-week 1|2|3|4|5|last]
```

Options

start-day

The day to change to summer time from standard time.

start-month

The month to change to summer time from standard time.

start-time

The time to change to summer time from standard time; uses the format hh:mm for a 24-hour clock (00:00-23:59).

start-week

The week to change to summer time from standard time.

end-day

The day to end summer time and start standard time.

end-month

The month to end summer time and start standard time.

end-time

The time to end summer time and start standard time; uses the format hh:mm for a 24-hour clock (00:00-23:59).

end-week

The week to end summer time and start standard time.

Show Date

Description Shows the date, according to the Device Server system clock.

User Level Normal, Admin

Syntax `show date`

Show Summertime

Description Shows the summertime settings.

User Level Admin

Syntax `show summertime`

Administration Commands

Bootup Commands

Reboot

Description Reboots the Device Server. You will be prompted to save configuration to FLASH, if there have been unsaved configuration changes.
User Level Admin
Syntax `reboot`

Reset

Description Resets the user profile or serial line to the default factory configuration.
User Level Admin
Syntax `reset user .|<username>|*`

`reset line <number>|*`

Reset Factory

Description Resets the Device Server to the factory configuration.
User Level Admin
Syntax `reset factory`

Save

Description Saves the configuration to FLASH.
User Level Admin
Syntax `save`

Set Bootup

Description Specifies remote the TFTP host and pathname for files to be loaded after a Device Server reboot.
User Level Admin
Syntax `set bootup firmware host <hostname> [file <path_filename>]`

`set bootup configuration host <hostname> [file <path_filename>]`

Options **firmware file**

The path and file name, relative to the default path of your TFTP server software, of the update software for the Device Server that will be loaded when the Device Server is rebooted.

configuration file

The path and file name, relative to the default path of your TFTP server software, of the configuration software for the Device Server that will be loaded when the Device Server is rebooted.

host

The host name or IP address of the server that contains the configuration or firmware file. If you use a host name, it must exist in the Device Server's host table or be resolved by DNS.

Show ARP

Description Shows the current contents of the ARP cache.
User Level Admin
Syntax `show arp`

Show Bootup

Description Shows the Firmware and Configuration files specified for Device Server bootup.
User Level Admin
Syntax `show bootup`

TFTP File Transfer Commands

Netload

Description Transfers a file from a remote host to the Device Server using the TFTP protocol.
User Level Admin
Syntax `netload firmware|configuration|customlang|term1|term2|term3| customapp-file|wan-driver <hostname/IP_address> <filename>`

Options **firmware**

Specifies that you are going to download a new firmware file to the Device Server.

configuration

Specifies that you are going to download a new configuration file to the Device Server.

customlang

Specifies that you are going to download a custom language file to the Device Server.

term1|term2|term3

You can create and download up to three custom terminal definitions to the Device Server.

customapp-file

You can download multiple SDK program executables and ancillary files using this command by running the command multiple times to download multiple files. Use the `shell` CLI command as described in the *SDK Programmer's Guide* to manage the files that you download.

wan-driver

Download wireless WAN custom drivers to the Device Server that have been downloaded from the Perle website.

<hostname/IP_address>

The IP address or host name where the file you are downloading to the Device Server resides. If you are using a host name, it must be resolved in either the Device Server's **Host Table** or a DNS server.

<filename>

The complete path and file name of the file you are downloading to the Device Server (this path should be relative to the default path of your TFTP server, which may or may not allow drive letters).

Netsave

Description Transfers a file from the Device Server to a remote host using the TFTP protocol.
User Level Admin
Syntax **`netsave configuration|crash <`***`hostname/IP_address`***`> <`***`filename`***`>`**

Options **configuration**

Specifies that you are going to upload a configuration file from the Device Server to the specified host or IP address.

crash

Specifies that you are going to upload a crash file from the Device Server to the specified host or IP address.

<***hostname/IP_address***>

The IP address or host name for where the file you are uploading from the Device Server is going. If you are using a host name, it must be resolved in either the Device Server's **Host Table** or a DNS server.

<***filename***>

The complete path and file name for the file you are uploading from the Device Server (this path should be relative to the default path of your TFTP server, which may or may not allow drive letters).

Keys and Certificates Commands

Netload

Description Loads certificates and keys into the Device Server.
User Level Admin
Syntax

```
netload https certificate|private-key <hostname/IP_address>
 <filename>

netload ldap certificate <hostname/IP_address> <filename>

netload ssh-client host <config_host> public-key ssh-1 rsa
 <hostname/IP_address> <filename>

netload ssh-client host <config_host> public-key ssh-2 rsa|dsa
 <hostname/IP_address> <filename>

netload ssh-client user <config_user> private-key ssh-1 rsa
 <hostname/IP_address> <filename>

netload ssh-client user <config_user> private-key ssh-2 rsa|dsa
 <hostname/IP_address> <filename>

netload ssh-server user <config_user> public-key ssh-2 rsa|dsa
 <hostname/IP_address> <filename>
```

Options **https certificate|private-key**

If you are using the secure version of the WebManager (HTTPS), then you need to download the SSL/TLS private key and CA list to make a secure connection.

ldap certificate

If you are using LDAP authentication with TLS, you need to download the certificate of the CA who signed the LDAP certificate to the Device Server for authentication to work properly.

ssh-client host

The public key for the host that is being authenticated by the Device Server's SSH server.

public-key ssh-1

Specify ssh-1 when you are using SSH version 1.

public-key ssh-2

Specify ssh-2 when you are using SSH version 2.

rsa|dsa

When downloading keys to the Device Server, specify the authentication method used by the key.

ssh-client user

The user that the SSH key is for.

ssh-server user

The user that the SSH key is for.

<hostname/IP_address>

Enter the host or IP address that contains the certificate/key you are downloading to the Device Server. If you are using a host name, If you are using a host name, it must be resolved in either the Device Server's **Host Table** or a DNS server.

<filename>

Enter the complete path and file name of the certificate/key you are downloading to the Device Server.

Netsave

Description Uploads certificates and keys from the Device Server to a remote host using TFTP.

User Level Admin

Syntax **`netsave ssh-server public-key ssh-2 rsa|dsa <`*`hostname/IP_address`*`> <`*`filename`*`>`**

Options **rsa|dsa**

When uploading SSH keys from the Device Server, specify the SSH authentication method used by the SSH key.

<hostname/IP_address>

The IP address or host name for where the SSH key you are uploading from the Device Server is going. If you are using a host name, it must be resolved in either the Device Server's **Host Table** or a DNS server.

<filename>

The complete path and file name for the file you are uploading from the Device Server (this path should be relative to the default path of your TFTP server, which may or may not allow drive letters).

MOTD Commands

Set MOTD

Description Specifies the server/file that contains the message of the day (MOTD) that is displayed when users log into the Device Server.

User Level Normal, Admin

Syntax **`set motd host <`*`hostname`*`> file <`*`path_filename`*`>`**

Options **host**

The host that the Device Server will be getting the Message of the Day file from.

file

The path and file name, relative to the default path of your TFTP server software, of the file that contains a string that is displayed when a user connects to the Device Server.

Show MOTD

Description Show the Message of the Day (MOTD) settings.

User Level Admin

Syntax **`show motd`**

Statistic Commands

Configuration Statistics

Show Netstat

Description Shows currently used TCP/UDP sockets/ports.

User Level Admin

Syntax **`show netstat [all] [listening] [tcp] [udp] [tcpv6] [updv6]`**

Options **all**

Displays all ports, including server (listening) ports; by default, listening ports are not displayed.

listening

Displays server (listening) ports; by default, listening ports are not displayed.

tcp

Displays TCP port statistics.

udp

Displays UDP port statistics.

tcpv6

Displays TCPv6 port statistics.

udpv6

Displays UDPv6 port statistics.

Show Netstat Statistics

Description Shows protocol (IP/ICMP/TCP/UDP) counters.

User Level Admin

Syntax **`show netstat statistics [ip] [ipv6] [icmp] [icmpv6] [tcp] [udp] [udp6]`**

Show Modbus Statistics

Description Shows the Modbus statistics.
User Level Admin
Syntax

```
show modbus statistics master-tcp line *|<number>

show modbus statistics master-udp line  *|<number>

show modbus statistics slave-tcp line  *|<number>

show modbus statistics slave-udp line *|<number>
```

Show Routes

Description Shows current information about IPv4 or IPv6 network routes.
User Level Admin
Syntax `show routes [ipv6]`

Run-Time Statistics

Delete Arp

Description Delete entries from the Device Server's ARP cache. Takes effect immediately; not related to configuration.
User Level Admin
Syntax `delete arp`

Show Arp

Description Shows the current contents of the ARP cache.
User Level Admin
Syntax `show arp`

Show Serial

Description Shows statistics on the serial port.
User Level Admin
Syntax `show serial [<line_number>]`

Uptime

Description Displays the elapsed time (in days, hours, minutes, and seconds) since the last reboot/power cycle.
User Level Admin
Syntax `uptime`

IOLAN+ User Commands

You can configure the Device Server using the IOLAN+ menu. See the *IOLAN+ User's Guide* for the command line interface and menu parameters. See ***IOLAN+ Interface*** on page 71 for a list of changes to the IOLAN+ menu.

IOLAN+

Description Displays the IOLAN+ configuration menu.
User Level Admin
Syntax `iolan+`

I/O Commands

Global I/O Commands

Set IO UDP

Description Sets the UDP settings for I/O unicast messages.

User Level Admin

Syntax

```
set io udp [mode on|off]
 [broadcast-interval <broadcast_interval>]

set io udp entry 1|2|3|4 disabled

set io udp entry 1|2|3|4 <udp_port> <start_ip> [<end_ip>]
```

Options

mode

Enables/disables UDP broadcast of I/O channel status (data).

broadcast-interval

Enter the interval, in seconds, for UDP broadcasts of I/O channel status (data). Valid values are 1-9999. Default value is 30 seconds.

entry

You can specify up to four sets of UDP IP address that will receive the I/O unicast.

udp_port

The UDP port that the Device Server will use to relay messages to servers/hosts.

start_ip

The first host IP address in the range of IP addresses (for IPV4 or IPV6) that the Device Server will listen for messages from and/or send messages to.

end_ip

The last host IP address in the range of IP addresses (for IPV4, not required for IPV6) that the Device Server will listen for messages from and/or send messages to.

Set IO Failsafe

Description Sets the failsafe (watchdog) settings for I/O.

User Level Admin

Syntax

```
set io failsafe [mode on|off] [timeout <seconds>]
```

Options

mode

Enables/disables the **Failsafe Timer**. This is the global setting that must be enabled to set the **Failsafe Action** on the channel for digital outputs and relays. When this timer expires because of no I/O activity within the specified time interval, the **Failsafe Action** set for the channel determines the action on the output.

timeout

The number of seconds that must elapse with no I/O activity before the channel **Failsafe Action** is triggered. Valid values are 1-9999. The default is 30 seconds.

Set IO Modbus

Description Enabling the Modbus option makes the Device Server act as a Modbus Slave, allowing Modbus Masters to communicate with the Device Server to control and/or retrieve I/O data.

User Level Admin

Syntax `set io modbus [mode on|off] [uid <1-255>]`

Options **mode**

Enables/disables Modbus as the communication protocol for all the I/O channels.

uid

This is the UID you are assigning to the Device Server, which is acting as a Modbus slave.

Set IO Temperature-Scale

Description Sets the temperature scale that will be used for all I/O temperature readings.

User Level Admin

Syntax `set io temperature-scale celsius|fahrenheit`

Option **temperature-scale**

Select the temperature scale that will be used to display temperature data, either Fahrenheit or Celsius. The default is Celsius.

Set Line

Set Line Service

Description Sets the **Line Service** settings for signal I/O.

User Level Admin

Syntax `set line <number> service signal-io`

Option **signal-io**

Sets the line to use signal I/O. You still need to define the serial pins for digital input (CTS, DSR, or DCD) or digital output (RTS or DTR). See ***Set IOChannel Digital Input (Serial Pins)*** on page 341 or ***Set IOChannel Digital Output (Serial Pins)*** on page 343 for configuration options.

Set IOChannel

Set IOChannel Mode

Description Sets general I/O channel settings for the specified channel, these settings are available to all channels and I/O serial pins.

User Level Admin

Syntax `set iochannel <i/o_channel> [mode enabled|disabled] [description <string>]`

Options ***i/o_channel***

Specify the channel number, for example, d2 or a4. Temperature models use Analog input, so the channel numbers are a1-a4.

mode

Enables the channel, allowing the settings to become active.

description

Provide a description of the channel, making it easier to identify. The channel description can be up to 20 characters.

Set IOChannel Digital I/O

Description Sets up the Digital I/O channel to act as either an output or input channel.

User Level Admin

Syntax

```
set iochannel <digital_channel> source-type input|output
```

Options ***digital_channel***

Specify the Digital channel number, for example, d2.

source-type

Specify whether the channel will drive the line (output) or will be reading the status of the line (input). The default is **Input**. The internal jumpers must match the software configuration, so if you change this setting to **Output**, you will have to also change the internal hardware jumpers.

Set IOChannel Digital Input

Description Sets the Digital input settings for the channel.

User Level Admin

Syntax

```
set iochannel <digital_channel>
 [alarm [trigger disabled|inactive-input|active-input]
   [clear auto|manual] [email on|off] [syslog on|off]
   [snmp on|off]]
 [description <string>] [invert-signal on|off]
 [latch disabled|inactive-to-active|active-to-inactive]
```

Options ***digital_channel***

Specify the Digital channel number, for example, d2.

alarm

Configures alarm settings when the Digital input trigger is activated.

trigger

When the trigger condition is met, triggers the specified alarm action. Triggers can be:

- **Disabled**—No alarm settings. This is the default.
- **Inactive**—When the expected Digital input is active, going inactive will trigger an alarm.
- **Active**—When the expected Digital input is inactive, going active will trigger an alarm.

clear

Specify **Manual** to manually clear an alarm. Specify **Auto** to automatically clear the alarm when the trigger condition changes; for example, if the **Trigger** is **Inactive** and the alarm is triggered, once the input becomes active again, the alarm will be cleared when **Auto** is set. The default is **Auto**.

email

Sends an email alert to an email account(s) set up in the Server settings (the Line Email Alert settings are not used with this feature) when an alarm is triggered or cleared. The email alert data includes the severity level and the value that caused the alarm to trigger or clear. The **Email Alert** is associated with **Level Critical**.

syslog

Sends a message to syslog when an alarm is triggered or cleared. The syslog entry includes the severity level and the value that caused the alarm to trigger or clear. The syslog message is associated with **Level Critical**.

snmp

Sends an SNMP trap when an alarm is triggered or cleared. The trap consists of the severity level and whether the alarm was triggered or cleared.

description

Provide a description of the channel, making it easier to identify. The channel description can be up to 20 characters.

invert-signal

Inverts the actual condition of the I/O signal in the status; therefore, an inactive status will be displayed as active.

latch

Latches (remembers) the activity transition (active-to-inactive or inactive-to-active). The default is disabled.

Set IOChannel Digital Input (Serial Pins)

Description Sets the Digital input settings for serial pins CTS, DSR, and DCD. This option is only available when the **Line Service** is set to **Signal I/O**.

User Level Admin

Syntax

```
set iochannel cts|dsr|dcd
 [alarm [trigger disabled|inactive-input|active-input]
   [clear auto|manual] [email on|off] [syslog on|off]
   [snmp on|off]]
 [description <string>] [invert-signal on|off]
 [latch disabled|inactive-to-active|active-to-inactive]
```

Options ***digital_channel***

Specify the Digital channel number, for example, d2.

alarm

Configures alarm settings when the Digital input trigger is activated.

trigger

When the trigger condition is met, triggers the specified alarm action. Triggers can be:

- **Disabled**—No alarm settings. This is the default.
- **Inactive**—When the expected Digital input is active, going inactive will trigger an alarm.
- **Active**—When the expected Digital input is inactive, going active will trigger an alarm.

clear

Specify **Manual** to manually clear an alarm. Specify **Auto** to automatically clear the alarm when the trigger condition changes; for example, if the **Trigger** is **Inactive** and the alarm is triggered, once the input becomes active again, the alarm will be cleared when **Auto** is set. The default is **Auto**.

email

Sends an email alert to an email account(s) set up in the Server settings (the Line Email Alert settings are not used with this feature) when an alarm is triggered or cleared. The email alert data includes the severity level and the value that caused the alarm to trigger or clear. The **Email Alert** is associated with **Level Critical**.

syslog

Sends a message to syslog when an alarm is triggered or cleared. The syslog entry includes the severity level and the value that caused the alarm to trigger or clear. The syslog message is associated with **Level Critical**.

snmp

Sends an SNMP trap when an alarm is triggered or cleared. The trap consists of the severity level and whether the alarm was triggered or cleared.

description

Provide a description of the channel, making it easier to identify. The channel description can be up to 20 characters.

invert-signal

Inverts the actual condition of the I/O signal in the status; therefore, an inactive status will be displayed as active.

latch

Latches (remembers) the activity transition (active-to-inactive or inactive-to-active). The default is disabled.

Set IOChannel Digital Output

Description Sets the Digital output channel settings.

User Level Admin

Syntax

```
set iochannel <digital_channel>
 [type sink|source|sink-and-source] [active-signal-width <width>]
 [inactive-signal-width <width>]
 [failsafe-action none|activate-output|deactivate-output]

set iochannel <digital_channel>
 output [pulse continuous|counted <pulse_count>]
 [active-to-inactive-delay <delay>]
 [inactive-to-active-delay <delay>]
```

Options ***digital_channel***

Specify the Digital channel number, for example, d2.

type

Specify the type of digital output:

- **Sink**—Specifies that the channel will be grounded when active.
- **Source**—Specifies that the channel will provide voltage when active.
- **Sink and Source**—Specifies that channel will be grounded when it is inactive and will provide voltage when it is active.

The default is **Sink**.

active-signal-width

How long the channel will be active during the pulse mode. Valid values are 1-9999 x 100 ms. The default is 100 ms.

inactive-signal-width

How long the channel will remain inactive during pulse mode. Valid values are 1-9999 x 100 ms. The default is 100 ms.

failsafe-action

When there has been no I/O activity within the specified time (set in the Global Settings) and the **Failsafe Timer** is triggered, you can set the **Failsafe Action** to:

- **None**—The state of the Digital/Relay output remains the same, no change.
- **Activate Output**—Activates the channel.
- **Deactivate Output**—Deactivates the channel.

output

Specify how the channel output will be handled:

- **Manual**—You must manually manipulate the channel output.
- **Pulse**—Activates and deactivates the channel output activity in intervals after it is manually activated.
- **Inactive-to-Active Delay**—The channel output will remain inactive for the specified time interval after it is manually started.
- **Active-to-Inactive Delay**—The channel output will go inactive after the specified time interval after it is manually started.

The default is **Manual**.

pulse

When the **Output** is **Pulse**, you can have it pulse in a **Continuous** manner or specify a pulse **Count** (each count consists of an active/inactive sequence). The default is **Continuous**.

active-to-inactive-delay

How long to delay an active-to-inactive or inactive-to-active setting after it is manually started. Valid values are 1-9999 x 100 ms. The default is 100 ms.

inactive-to-active-delay

How long to delay an active-to-inactive or inactive-to-active setting after it is manually started. Valid values are 1-9999 x 100 ms. The default is 100 ms.

Set IOChannel Digital Output (Serial Pins)

Description Sets the Digital output for serial pins RTS and DTR. This option is only available when the **Line Service** is set to **Signal I/O**.

User Level Admin

Syntax

```
set iochannel rts|dtr [description <string>]
 [failsafe-action none|activate-outut|deactivate-output]
 [mode enabled|disabled]
```

Options

description

Provide a description of the channel, making it easier to identify. The channel description can be up to 20 characters.

failsafe-action

When there has been no I/O activity within the specified time (set in the Global Settings) and the **Failsafe Timer** is triggered, you can set the **Failsafe Action** to:

- **None**—The state of the Digital/Relay output remains the same, no change.
- **Activate Output**—Activates the channel.
- **Deactivate Output**—Deactivates the channel.

mode

Enables the channel, allowing the settings to become active.

Set IOChannel Relay

Description Sets the Relay output channel settings.

User Level Admin

Syntax

```
set iochannel <relay_number> output
 [pulse continuous|counted <pulse_count>]
 [active-to-inactive-delay <delay>]
 [inactive-to-active-delay <delay>]

set iochannel <relay_number>
 [active-signal-width <width>] [inactive-signal-width <width>]
 [failsafe-action none|activate|deactivate]
```

Options ***relay_number***

Specify the Relay channel number, for example, r2.

output

Specify how the channel output will be handled:

- **Manual**—You must manually manipulate the channel output.
- **Pulse**—Activates and deactivates the channel output activity in intervals after it is manually activated.
- **Inactive-to-Active Delay**—The channel output will remain inactive for the specified time interval after it is manually started.
- **Active-to-Inactive Delay**—The channel output will go inactive after the specified time interval after it is manually started.

The default is **Manual**.

pulse

When the **Output** is **Pulse**, you can have it pulse in a **Continuous** manner or specify a pulse **Count** (each count consists of an active/inactive sequence). The default is **Continuous**.

active-to-inactive-delay

How long to delay an active-to-inactive or inactive-to-active setting after it is manually started. Valid values are 1-9999 x 100 ms. The default is 100 ms.

inactive-to-active-delay

How long to delay an active-to-inactive or inactive-to-active setting after it is manually started. Valid values are 1-9999 x 100 ms. The default is 100 ms.

active-signal-width

How long the channel will be active during the pulse mode. Valid values are 1-9999 x 100 ms. The default is 100 ms.

inactive-signal-width

How long the channel will remain inactive during pulse mode. Valid values are 1-9999 x 100 ms. The default is 100 ms.

failsafe-action

When there has been no I/O activity within the specified time (set in the Global Settings) and the **Failsafe Timer** is triggered, you can set the **Failsafe Action** to:

- **None**—The state of the Digital/Relay output remains the same, no change.
- **Activate Output**—Activates the channel.
- **Deactivate Output**—Deactivates the channel.

Set IOChannel Analog (True Analog)

Description Sets the Analog input channel settings.

User Level Admin

Syntax

```
set iochannel <analog_channel> type current|voltage
 range <range_specifier>

set iochannel <analog_channel> alarm
 [level 1|2|3|4|5 [mode on|off] [trigger-type disabled|low|high]
  [trigger-level <decimal_value>] [clear-mode auto|manual]
  [clear-level <decimal_value>] [email on|off] [snmp on|off]
  [syslog on|off]]
```

Options ***analog_channel***

Specify the Analog channel number, for example, a2 or a4 (this also applies to Temperature models).

type

Select the type of input being measured, either **Current** or **Voltage**. The default is **Current**.

range

Select the range for the measurement type. For current, the range is:

- 0-20 (0-20mA) This is the default.
- 4-20 (04-20mA)

For voltage, the range is:

- 1 (+/-1V)
- 5 (+/-5V)
- 10 (+/-10V) This is the default.
- 150 (+/-150mV)
- 500 (+/-500mV)

alarm

Configures alarm settings when the Analog input trigger is activated.

level

You can specify up to five alarm trigger/clear severity levels. If the **Trigger Type** is **Low**, an alarm is triggered when the input drops below the specified **Trigger** value; other severity level trigger values must decrease in value with each subsequent level. If the **Trigger Type** is **High**, an alarm is triggered when the input is higher than the specified **Trigger** value; other severity level trigger values must increase in value with each subsequent level. To clear an alarm, the input must drop below the specified value when **Trigger Type** is **High** or go above the specified value when **Trigger Type** is **Low**.

mode

Enables/disables an alarm level. The default is off.

trigger-type

If the **Trigger Type** is **Low**, an alarm is triggered when the input drops below the specified **Trigger** value; other severity level trigger values must decrease in value with each subsequent level. If the **Trigger Type** is **High**, an alarm is triggered when the input is higher than the specified **Trigger** value; other severity level trigger values must increase in value with each subsequent level.

trigger-level

Specify the value that will trigger an alarm, the measurement is based on the **Type** and **Range** that you specify. This value must not fall within the scope of the value used to clear an alarm.

clear-mode

Specifies whether an activated alarm must be **Manually** cleared, or can be cleared when the input drops below the specified value (when **Trigger Type** is **High**) or goes above the specified value (when **Trigger Type** is **Low**).

clear-level

Specify that value that will clear an alarm, the measurement is based on the **Type** and **Range** that you specify. This value must not fall within the scope of the value used to trigger an alarm.

email

Sends an email alert to an email account(s) set up in the Server settings (the **Line Email Alert** settings are not used with this feature) when an alarm is triggered or cleared. The email alert data includes the severity level and the value that caused the alarm to trigger or clear. The Email Alert is associated with **Level Critical**.

snmp

Sends an SNMP trap when an alarm is triggered or cleared. The trap consists of the severity level and whether the alarm was triggered or cleared.

syslog

Sends a message to syslog when an alarm is triggered or cleared. The syslog entry includes the severity level and the value that caused the alarm to trigger or clear. The syslog message is associated with **Level Critical**.

Set IOChannel Analog (Temperature)

Description Sets the Analog input channel settings for Temperature models.

User Level Admin

Syntax

```
set iochannel <analog_channel> type rtd|thermocouple
 range <range_specifier>

set iochannel <analog_channel> alarm
 [level 1|2|3|4|5 [mode on|off] [trigger-type disabled|low|high]
  [trigger-level <decimal_value>] [clear-mode auto|manual]
  [clear-level <decimal_value>] [email on|off] [snmp on|off]
  [syslog on|off]]
```

Options ***analog_channel***

Specify the Analog channel number, for example, a2 or a4 (this also applies to Temperature models).

type

Specify the type of sensor you are using to measure temperature, either RTD or thermocouple. The default is RTD.

range

Specify the temperature range that you want to measure. For RTD, the range is:

- 1 (Pt100 a=385 -50 to 150C) This is the default.
- 2 (Pt100 a=385 0 to 100C)
- 3 (Pt100 a=385 0 to 200C)
- 4 (Pt100 a=385 0 to 400C)
- 5 (Pt100 a=385 -200 to 200C)
- 6 (Pt100 a=392 -50 to 150C)
- 7 (Pt100 a=392 0 to 100C)
- 8 (Pt100 a=392 0 to 200C)
- 9 (Pt100 a=392 0 to 400C)
- 10 (Pt100 a=392 -200 to 200C)
- 11 (Pt1000 a=385 -40 to 160C)
- 12 (NiFe604 a=518 -80 to 100C)
- 13 (NiFe604 a=518 0 to 100C)

For thermocouple, the range is:

- b (B 500 to 1800C)
- e (E 0 to 1000C)
- j (J 0 to 760C) This is the default.
- k (K 0 to 1370C)
- r (R 500 to 1750C)
- s (S 500 to 1750C)
- t (T -100 to 400C).

alarm

Configures alarm settings when the Analog input trigger is activated.

level

You can specify up to five alarm trigger/clear severity levels. If the **Trigger Type** is **Low**, an alarm is triggered when the input drops below the specified **Trigger** value; other severity level trigger values must decrease in value with each subsequent level. If the **Trigger Type** is **High**, an alarm is triggered when the input is higher than the specified **Trigger** value; other severity level trigger values must increase in value with each subsequent level. To clear an alarm, the input must drop below the specified value when **Trigger Type** is **High** or go above the specified value when **Trigger Type** is **Low**.

mode

Enables/disables an alarm level. The default is off.

trigger-type

If the **Trigger Type** is **Low**, an alarm is triggered when the input drops below the specified **Trigger** value; other severity level trigger values must decrease in value with each subsequent level. If the **Trigger Type** is **High**, an alarm is triggered when the input is higher than the specified **Trigger** value; other severity level trigger values must increase in value with each subsequent level.

trigger-level

Specify the value that will trigger an alarm, the measurement is based on the **Type** and **Range** that you specify. This value must not fall within the scope of the value used to clear an alarm.

clear-mode

Specifies whether an activated alarm must be **Manually** cleared, or can be cleared when the input drops below the specified value (when **Trigger Type** is **High**) or goes above the specified value (when **Trigger Type** is **Low**).

clear-level

Specify that value that will clear an alarm, the measurement is based on the **Type** and **Range** that you specify. This value must not fall within the scope of the value used to trigger an alarm.

email

Sends an email alert to an email account(s) set up in the Server settings (the **Line Email Alert** settings are not used with this feature) when an alarm is triggered or cleared. The email alert data includes the severity level and the value that caused the alarm to trigger or clear. The Email Alert is associated with **Level Critical**.

snmp

Sends an SNMP trap when an alarm is triggered or cleared. The trap consists of the severity level and whether the alarm was triggered or cleared.

syslog

Sends a message to syslog when an alarm is triggered or cleared. The syslog entry includes the severity level and the value that caused the alarm to trigger or clear. The syslog message is associated with **Level Critical**.

Kill IOChannel

Description Kills the I/O channel.

User Level Admin

Syntax

```
kill iochannel <i/o_channel>

kill iochannel line <number> rts|cts|dtr|dsr|dcd
```

Options ***i/o_channel***

Specify the channel number, for example, d2 or a4. Temperature models use Analog input, so the channel numbers are a1-a4.

rts|cts|dtr|dsr|dcd

Specify the Digital output pins (RTS or DTR) or Digital input pins (CTS, DSR, or DCD).

Show IO

Description Shows global I/O information (for example, UDP, TruePort, Modbus). Temperature I/O is Analog.

User Level Admin

Syntax

```
show iochannel <i/o_channel>

show iochannel rts|cts|dtr|dsr|dcd
```

Options ***i/o_channel***

Specify the channel number, for example, d2 or a4. Temperature models use Analog input, so the channel numbers are a1-a4.

rts|cts|dtr|dsr|dcd

Specify the Digital output pins (RTS or DTR) or Digital input pins (CTS, DSR, or DCD).

Show IOChannel

Description Shows I/O channel information. Temperature I/O is Analog.
User Level Admin
Syntax `show iochannel <i/o_channel>`

`show iochannel line <number> rts|cts|dtr|dsr|dcd`

Options ***i/o_channel***

Specify the channel number, for example, d2 or a4. Temperature models use Analog input, so the channel numbers are a1-a4.

rts|cts|dtr|dsr|dcd

Specify the Digital output pins (RTS or DTR) or Digital input pins (CTS, DSR, or DCD).

I/O Channel Control Commands

The I/O commands in this section are used to manually manage the I/O channels.

Digital Output

Description Manages the Digital output channel status. Not all models have four Digital channels, most have just two.
User Level Admin
Syntax `iochannel d1|d2|d3|d4|cts|dsr|dcd clear alarm|input-latch`

Options **alarm**

Clears the alarm. Note that if the condition that tripped the alarm still exists, the alarm will not look like it's cleared, but will reflect the appropriate alarm level severity. Alarm Level 0 means that the alarm has not been triggered.

latch-input

Clears the latch value.

Digital Input

Description Manages the Digital input channel status.
User Level Admin
Syntax `iochannel d1|d2|d3|d4|rts|dtr output activate|deactivate`

Option **output**

Manually activates/deactivates the I/O channel.

Relay

Description Manages the Relay output channel status.
User Level Admin
Syntax `iochannel r1|r2 output activate|deactivate`

Option **output**

Manually activates/deactivates the I/O channel.

Analog Input

Description Manages the Analog input channel status.
User Level Admin
Syntax `iochannel a1|a2|a3|a4 clear alarm|min|max`

Options **alarm**

Clears the alarm. Note that if the condition that tripped the alarm still exists, the alarm will not look like it's cleared, but will reflect the appropriate alarm level severity. Alarm Level 0 means that the alarm has not been triggered.

min

Clears the minimum value.

max

Clears the maximum value.

Calibrating Analog Input (Analog/Temperature)

Calibrate Analog

Description Calibrates the Analog input channel. When this command is issued, a script will automatically start, requesting that the minimum and maximum calibration values be applied to the requested Analog/Temperature channel. See ***Calibrating Analog Input*** on page 121 for more information.
User Level Admin
Syntax `iochannel a1|a2|a3|a4 calibrate`

Reset Calibration

Description Resets the calibration to factory defaults.
User Level Admin
Syntax `reset io calibration`

Power Commands

Description Actively controls the RPS plug power.

User Level Admin, Normal

Syntax

```
power cycle line <number> [plug <number|range|*>]

power on line <number> [plug <number|range|*>]

power off line <number> [plug <number|range|*>]

power reset line <number>

power status line <number>
```

Options

cycle

Turns the specified plug(s) off and then on.

on

Turns the specified plug(s) on.

off

Turns the specified plug(s) off.

reset

Resets all the RPS plugs to the default state as defined in the Power Management line settings.

status

Displays the status (on/off) of the plug(s).

A RADIUS

Introduction

Although RADIUS can be used strictly for external authentication, it can also be used to configure line and user parameters. Therefore, when a user is being authenticated using RADIUS, it is possible that the user's configuration is a compilation of the parameters passed back from RADIUS, the Device Server parameters if the user has also been set up as a local user in the Device Server, and the Default User's parameters for any parameters that have not been set by either RADIUS or the user's local configuration.

Supported RADIUS Parameters

This section describes the attributes which will be accepted by the Device Server from a RADIUS server in response to an authentication request.

Type	Name	Description
1	User-Name	The name of the user to be authenticated.
2	User-Password	The password of the user to be authenticated.
6	Service-Type	Indicates the service to use to connect the user to the Device Server. A value of 6 indicates administrative access to the Device Server. Supported values are: ● 1—Login ● 3—Callback-Login Equivalent to the Device Server **User Service** set by Type 15, Login-Service. ● 2—Framed ● 4—Callback-Framed Equivalent to the Device Server **User Service** set by Type 7, Framed-Protocol. ● 7—NAS prompt ● 9—Callback NAS-prompt Equivalent to Device Server **User Service DSLogin**. ● 6—Administrative User ● 11—Callback Administrative User Equivalent to Device Server **User Service DSLogin** and the User gets Admin privileges.

Type	Name	Description
7	Framed-Protocol	The link layer protocol to be used by this user. Determines the User Service when Service-Type is set to Framed or Callback-Framed. Supported values are: ● 1—PPP ● 2—SLIP
8	Framed-IP-Address	The IP Address to be assigned to this user for PPP or SLIP.
9	Framed-IP-Netmask	The subnet to be assigned to this user for PPP or SLIP.
12	Framed-MTU	Attribute indicates the Maximum Transmission Unit (MTU) to be configured for the user, when it is not negotiated by some other means such as PPP.
13	Framed-Compression	Indicates a compression protocol to be used for the PPP or SLIP link. Supported value is: ● 1—Van Jacobson TCP/IP compression.
14	Login-Host	Indicates the host with which the user can connect to when the Service-Type is set to **1** (Login) or **3** (Callback-Login).
15	Login-Service	Indicates the Device Server **User Service** to use to connect the user a host. Supported values are: ● 0—Telnet ● 1—Rlogin ● 2—TCP Clear ● 5—SSH ● 6—SSL Raw
16	Login-TCP-Port	Indicates the TCP port with which the user is to be connected when the Service-Type is set to **1** (Login) or **3** (Callback-Login).
19	Callback-Number	Specifies the callback phone number. This is the same implementation as **20** (Callback-ID), but takes precedence if **20** is set.
20	Callback-ID	Specifies the callback phone number. This is the same implementation as **19** (Callback-Number), but **19** takes precedence if both are set.

Type	Name	Description
26	Vendor-Specific	Perle's defined attributes for line access rights and user level. See ***Perle RADIUS Dictionary Example*** on page 358 for an example of this file. Line Access Rights for port ***n*** (where ***n*** is the line number): Name: Perle-Line-Access-Port-***n*** Type: 100 + ***n*** Data Type: Integer Value: Disabled (0), ReadWrite(1), ReadInput(2), ReadInputWrite (3), ReadOutput (4), ReadOutputWrite (5), ReadOutputInput (6), ReadOutputInputWrite (7) Name: Perle-User-Level Type: 100 Data Type: Integer Value: Admin(1), Normal(2), Restricted(3), Menu(4)
27	Session-Timeout	Maximum number of seconds the user will be allowed to stay logged on.
28	Idle-Timeout	Use this timer to close a connection because of inactivity. When the Idle-Timeout expires, the Device Server will end the connection. The maximum value is 4294967 seconds (about 49 days). A value of 0 (zero) means the Idle-Timeout will not expire, so the connection is permanently open.
96	Framed-Interface-Id	The remote IPv6 interface identifier for the remote end of the PPP link.

Accounting Message

This section describes the attributes which will be included by the Device Server when sending an accounting message to the RADIUS server.

Type	Name	Description
1	User-Name	The name of the user to be authenticated.
4	NAS-IP-Address	IP Address of Device Server LAN interface.
5	NAS	Port Line number of Device Server.
6	Service-Type	Indicates the service to use to connect the user to the Device Server. A value of 6 indicates administrative access to the Device Server. Supported values are: ● 1—Login ● 3—Callback-Login Equivalent to the Device Server **User Service** set by Type 15, Login-Service. ● 2—Framed ● 4—Callback-Framed Equivalent to the Device Server **User Service** set by Type 7, Framed-Protocol. ● 7—NAS prompt ● 9—Callback NAS-prompt Equivalent to Device Server **User Service DSLogin**. ● 6—Administrative User ● 11—Callback Administrative User Equivalent to Device Server **User Service DSLogin** and the User gets Admin privileges.
40	Acct-Status-Type	Indicates if this is the beginning or end of a session. Supported values are: 1 = Start 2 =Stop.
42	Acct-Input-Octets	Number of bytes which were received from the user during this session.
43	Acct-Output-Octets	Number of bytes where were transmitted to the user during this session.
44	Acct-Session-ID	A string which identifies the session. The same string must be used in the start and stop messages.
45	Acct-Authentic	Indicates how the user was authenticated. Supported values are: 1 = Local 2 = RADIUS.
46	Acct-Session-Time	Number of seconds for which the user has been connected to a specific session.
47	Acct-Input-Packets	Number of packets which were received from the user during this session.
48	Acct-Output-Packets	Number of packets which were transmitted to the user during this session.

Type	Name	Description
49	Acct-Terminate-Cause	Indicates how the session was terminated: Supported values include: 1 = User Request 2= Lost Carrier 3=Lost Service 4= Idle Timeout 5= Session Timeout 14 = Port Suspended 16 = Callback.

Mapped RADIUS Parameters to Device Server Parameters

When authentication is being done by RADIUS, there are several **Line** and **User** parameters that can be set by the RADIUS server. Any parameters sent by that RADIUS server that are not supported by the Device Server are discarded. Below is a list of the RADIUS parameters and their Device Server parameters:

RADIUS Parameter	Device Server Parameter
User-Service	This has no Device Server field, although it needs to be set to **`Framed-User`** in the RADIUS server.
Framed-Protocol	Set to SLIP or PPP service.
Framed-Address	Remote IP Address field under either SLIP or PPP. *Caution:* the exception to the above rule is a **`Framed-Address`** value of 255.255.255.254. When this value is specified in the RADIUS file, the unit will use the Remote IP address configured for a PPP line in the Device Server.
Framed-Netmask	**Subnet/Prefix Bits** field under either **SLIP** or **PPP**.
Framed-Compression	**VJ Compression** field under either **SLIP** or **PPP**.
Framed-MTU	**MTU** field under **SLIP**. **MRU** field under **PPP**.
Idle-Timeout	**Idle Timer** under **Line** settings.
Login-Service	Corresponds to one of the following **User Service** parameters: **Telnet**, **Rlogin**, **TCP Clear**, **SSH**, or **SSL Raw**.
Session-Timeout	**Session Timer** under **Line** settings.
Callback-Number	Combination of the **Callback On** and **Phone Numbe**r fields under **User** settings.
Callback-ID	Combination of the **Callback On** and **Phone Numbe**r fields under **User** settings.

Perle RADIUS Dictionary Example

The Device Server has defined Vendor Specific RADIUS attributes in order for the RADIUS server to be configured to support the Device Server features of Line Access Rights and User Level. These attributes have been defined in ***Supported RADIUS Parameters*** on page 353 to allow the RADIUS server to be configured for RADIUS users to have this level of configuration.

See below for an example of the Perle defined attributes for the RADIUS server for a 4-port Device Server (although the dictionary can contain 48 ports, even if they are not all defined):

```
# Perle dictionary.
#
#       Perle Systems Ltd.
#       http://www.perle.com/
#
#       Enable by putting the line "$INCLUDE dictionary.perle" into
#       the main dictionary file.
#
# Version:  1.20  30-Nov-2005  Add new line access right values for ports
#                              up to 49.
# Version:  1.10  11-Nov-2003  Add new line access right values
# Version:  1.00  17-Jul-2003  original release for vendor specific field
#                                 support
#

VENDOR  Perle       1966

#   Perle Extensions

ATTRIBUTE   Perle-User-Level            100 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-1    101 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-2    102 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-3    103 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-4    104 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-5    105 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-6    106 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-7    107 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-8    108 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-9    109 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-10   110 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-11   111 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-12   112 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-13   113 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-14   114 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-15   115 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-16   116 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-17   117 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-18   118 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-19   119 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-20   120 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-21   121 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-22   122 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-23   123 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-24   124 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-25   125 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-26   126 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-27   127 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-28   128 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-29   129 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-30   130 integer Perle
```

```
ATTRIBUTE   Perle-Line-Access-Port-31   131 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-32   132 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-33   133 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-34   134 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-35   135 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-36   136 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-37   137 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-38   138 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-39   139 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-40   140 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-41   141 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-42   142 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-43   143 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-44   144 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-45   145 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-46   146 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-47   147 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-48   148 integer Perle
ATTRIBUTE   Perle-Line-Access-Port-49   149 integer Perle

#   Perle User Level Values

VALUE   Perle-User-Level    Admin           1
VALUE   Perle-User-Level    Normal          2
VALUE   Perle-User-Level    Restricted      3
VALUE   Perle-User-Level    Menu            4

#   Perle Line Access Right Values

VALUE   Perle-Line-Access-Port-1    Disabled                0
VALUE   Perle-Line-Access-Port-1    Read-Write              1
VALUE   Perle-Line-Access-Port-1    Read-Input              2
VALUE   Perle-Line-Access-Port-1    Read-Input-Write        3
VALUE   Perle-Line-Access-Port-1    Read-Output             4
VALUE   Perle-Line-Access-Port-1    Read-Output-Write       5
VALUE   Perle-Line-Access-Port-1    Read-Output-Input       6
VALUE   Perle-Line-Access-Port-1    Read-Output-Input-Write 7

VALUE   Perle-Line-Access-Port-2    Disabled                0
VALUE   Perle-Line-Access-Port-2    Read-Write              1
VALUE   Perle-Line-Access-Port-2    Read-Input              2
VALUE   Perle-Line-Access-Port-2    Read-Input-Write        3
VALUE   Perle-Line-Access-Port-2    Read-Output             4
VALUE   Perle-Line-Access-Port-2    Read-Output-Write       5
VALUE   Perle-Line-Access-Port-2    Read-Output-Input       6
VALUE   Perle-Line-Access-Port-2    Read-Output-Input-Write 7

VALUE   Perle-Line-Access-Port-3    Disabled                0
VALUE   Perle-Line-Access-Port-3    Read-Write              1
VALUE   Perle-Line-Access-Port-3    Read-Input              2
VALUE   Perle-Line-Access-Port-3    Read-Input-Write        3
VALUE   Perle-Line-Access-Port-3    Read-Output             4
VALUE   Perle-Line-Access-Port-3    Read-Output-Write       5
VALUE   Perle-Line-Access-Port-3    Read-Output-Input       6
VALUE   Perle-Line-Access-Port-3    Read-Output-Input-Write 7
```

```
VALUE   Perle-Line-Access-Port-4    Disabled                0
VALUE   Perle-Line-Access-Port-4    Read-Write              1
VALUE   Perle-Line-Access-Port-4    Read-Input              2
VALUE   Perle-Line-Access-Port-4    Read-Input-Write        3
VALUE   Perle-Line-Access-Port-4    Read-Output             4
VALUE   Perle-Line-Access-Port-4    Read-Output-Write       5
VALUE   Perle-Line-Access-Port-4    Read-Output-Input       6
VALUE   Perle-Line-Access-Port-4    Read-Output-Input-Write 7

...
```

B TACACS+

Introduction

Although TACACS+ can be used strictly for external authentication, it can also be used to configure Line and User parameters. Therefore, when a user is being authenticated using TACACS+, it is possible that the user's configuration is a compilation of the parameters passed back from the TACACS+ authentication server, the User's Device Server parameters if the user has also been set up as a local user in the Device Server, and the Default User's parameters for any parameters that have not been set by either TACACS+ or the User's local configuration.

TACACS+ Parameter Values

User and Line parameters can be passed to the Device Server after authentication for Direct (users accessing the Device Server from the serial side) and Reverse (users accessing the Device Server from the Ethernet side) line connections.

Direct Users

This section describes the attributes which will be accepted by the Device Server from a TACACS+ server in response to an authentication request for Direct Users.

Name	Value(s)	Description
priv-lvl	12-15 (Admin) 8-11 (Normal) 4-7 (Restricted) 0-3 (Menu)	The Device Server privilege level. See ***User Levels*** on page 94 for more information.
Perle_User_Service	0 (Telnet) 1 (Rlogin) 2 (TCP_Clear) 3 (SLIP) 4 (PPP) 5 (SSH) 6 (SSL_Raw)	Corresponds to the User Service setting in the Device Server. If no value is specified, DSLogin is the default User Service.

service = telnet
{
addr =
port =
}
	
	IPv4 or IPv6 address
	TCP port number
		Settings when Perle_User_Service is set to 0.

Name	Value(s)	Description
service = rlogin		Settings when Perle_User_Service is set to 1.
{		
addr =	IPv4 or IPv6 address	
}		
service = tcp_clear		Settings when Perle_User_Service is set to 2.
{		
addr =	IPv4 or IPv6 address	
port =	TCP port number	
}		
service = slip		Settings when Perle_User_Service is set to 3.
{		
routing =	true (Send and Listen) false (None)	
addr =	IPv4 or IPv6 address	
}		
service = ppp		Settings when Perle_User_Service is set to 4.
{		
routing =	true (Send and Listen) false (None)	
addr =	IPv4 or IPv6 address	
port =	TCP port number	
ppp-vj-slot-compression	true or false	
callback-dialstring	phone number, no punctuation	
}		
service = ssh		Settings when Perle_User_Service is set to 5.
{		
addr =	IPv4 or IPv6 address	
port =	TCP port number	
}		
service = ssl_raw		Settings when Perle_User_Service is set to 6.
{		
addr =	IPv4 or IPv6 address	
port =	TCP port number	
}		

Direct User Example Settings

The following example shows the parameters that can be set for users who are accessing the Device Server from the serial side. These settings should be included in the TACACS+ user configuration file.

```
Service = EXEC
{
priv-lvl = x            // x = 12-15 (Admin)
                        // x = 8-11  (Normal)
                        // x = 4-7   (Restricted)
                        // x = 0-3   (Menu)

timeout=x                       // x = session timeout in seconds

idletime=x                      // x = Idle timeout in seconds

Perle_User_Service = x          // x = 0 Telnet
                                // x = 1 Rlogin
                                // x = 2 TCP_Clear
                                // x = 3 SLIP
                                // x = 4 PPP
                                // x = 5 SSH
                                // x = 6 SSL_RAW
                                // If not specified, command prompt
}

// Depending on what Perle_User_Service is set to

service = telnet
{
addr = x.x.x.x       // ipv4 or ipv6 addr
port = x             // tcp_port #
}

service = rlogin
{
addr = x.x.x.x        // ipv4 or ipv6 addr
}

service = tcp_clear
{
addr = x.x.x.x       // ipv4 or ipv6 addr
port = x             // tcp_port #
}


service = slip
{
routing=x         // x = true (Send and Listen)
                  // x = false (None)
addr = x.x.x.x    // ipv4 addr
}
```

```
service = ppp
{
routing=x         // x = true (Send and Listen)
                  // x = false (None)
addr = x.x.x.x    // ipv4 or ipv6 addr
ppp-vj-slot-compression = x // x =true or false
callback-dialstring = x // x=number to callback on
}

service = ssh
{
addr = x.x.x.x   // ipv4 or ipv6 addr
port = x             // tcp_port #
}

service = ssl_raw
{
addr = x.x.x.x   // ipv4 or ipv6 addr
port = x             // tcp_port #
}
```

Reverse Users

This section describes the attributes which will be accepted by the Device Server from a TACACS+ server in response to an authentication request for Reverse Users. The TACACS+ **service** needs to be set to **EXEC**/**raccess** or just **raccess** on the well known port .

Name	Value(s)	Description
priv-lvl	12-15 (Admin) 8-11 (Normal) 4-7 (Restricted) 0-3 (Menu)	The Device Server privilege level. See ***User Levels*** on page 94 for more information.
Perle_Line_Access_#	# = port number 0 (Disabled) 1 (ReadWrite) 2 (ReadInput) 3 (ReadInputWrite) 4 (ReadOuptut) 5 (ReadOutputWrite) 6 (ReadOutputInput) 7 (ReadOuputWrite)	For the specified line, provides the User's Line Access rights.
timeout	0-4294967	Session timeout in seconds.
idletime	0-4294967	Idle timeout in seconds

Reverse User Example Settings

The following example shows the parameters that can be set for users who are accessing the Device Server from the Ethernet side. These settings should be included in the TACACS+ user configuration file.

```
service = raccess
{
priv-lvl = x            // x = 12-15 (Admin)
                        // x = 8-11 (Normal)
                        // x = 4-7 (Restricted)
                        // x = 0-3 (Menu)

Perle_Line_Access_i=x   // i = port number
                        // x = 0 (Diasabled)
                        // x = 1 (ReadWrite)
                        // x = 2 (ReadInput)
                        // x = 3 (ReadInputWrite)
                        // x = 4 (ReadOuptut)
                        // x = 5 (ReadOutputWrite)
                        // x = 6 (ReadOutputInput)
                        // x = 7 (ReadOuputWrite)
timeout=x               // x = session timeout in seconds

idletime=x              // x = Idle timeout in seconds
}
```

C SSL/TLS Ciphers

Introduction

This appendix contains a table that shows valid SSL/TLS cipher combinations.

Valid SSL/TLS Ciphers

This chart displays all of the valid SSL/TLS combinations.

Full Name	SSL Ver.	Key-Exchange	Authentication	Encryption	Key-Size	HMAC
ADH-AES256-SHA	SSLv3	Kx=DH	Au=None	Enc=AES	256	Mac=SHA1
DHE-RSA-AES256-SHA	SSLv3	Kx=DH	Au=RSA	Enc=AES	256	Mac=SHA1
DHE-DSS-AES256-SHA	SSLv3	Kx=DH	Au=DSS	Enc=AES	256	Mac=SHA1
AES256-SHA	SSLv3	Kx=RSA	Au=RSA	Enc=AES	256	Mac=SHA1
EDH-RSA-DES-CBC3-SHA	SSLv3	Kx=DH	Au=RSA	Enc=3DES	168	Mac=SHA1
EDH-DSS-DES-CBC3-SHA	SSLv3	Kx=DH	Au=DSS	Enc=3DES	168	Mac=SHA1
DES-CBC3-SHA	SSLv3	Kx=RSA	Au=RSA	Enc=3DES	168	Mac=SHA1
DES-CBC3-MD5	SSLv2	Kx=RSA	Au=RSA	Enc=3DES	168	Mac=MD5
ADH-AES128-SHA	SSLv3	Kx=DH	Au=None	Enc=AES	128	Mac=SHA1
DHE-RSA-AES128-SHA	SSLv3	Kx=DH	Au=RSA	Enc=AES	128	Mac=SHA1
DHE-DSS-AES128-SHA	SSLv3	Kx=DH	Au=DSS	Enc=AES	128	Mac=SHA1
AES128-SHA	SSLv3	Kx=RSA	Au=RSA	Enc=AES	128	Mac=SHA1
RC2-CBC-MD5	SSLv2	Kx=RSA	Au=RSA	Enc=RC2	128	Mac=MD5
DHE-DSS-RC4-SHA	SSLv3	Kx=DH	Au=DSS	Enc=RC4	128	Mac=SHA1
RC4-SHA	SSLv3	Kx=RSA	Au=RSA	Enc=RC4	128	Mac=SHA1
RC4-MD5	SSLv3	Kx=RSA	Au=RSA	Enc=RC4	128	Mac=MD5
RC4-MD5	SSLv2	Kx=RSA	Au=RSA	Enc=RC4	128	Mac=MD5
RC4-64-MD5	SSLv2	Kx=RSA	Au=RSA	Enc=RC4	64	Mac=MD5
EXP1024-DHE-DSS-DES-CBC-SHA	SSLv3	Kx=DH(1024)	Au=DSS	Enc=DES	56	Mac=SHA1
EXP1024-DES-CBC-SHA	SSLv3	Kx=RSA(1024)	Au=RSA	Enc=DES	56	Mac=SHA1
EXP1024-RC2-CBC-MD5	SSLv3	Kx=RSA(1024)	Au=RSA	Enc=RC2	56	Mac=MD5

Full Name	SSL Ver.	Key-Exchange	Authentication	Encryption	Key-Size	HMAC
EDH-RSA-DES-CBC-SHA	SSLv3	Kx=DH	Au=RSA	Enc=DES	56	Mac=SHA1
EDH-DSS-DES-CBC-SHA	SSLv3	Kx=DH	Au=DSS	Enc=DES	56	Mac=SHA1
DES-CBC-SHA	SSLv3	Kx=RSA	Au=RSA	Enc=DES	56	Mac=SHA1
DES-CBC-MD5	SSLv2	Kx=RSA	Au=RSA	Enc=DES	56	Mac=MD5
EXP1024-DHE-DSS-RC4-SHA	SSLv3	Kx=DH(1024)	Au=DSS	Enc=RC4	56	Mac=SHA1
EXP1024-RC4-SHA	SSLv3	Kx=RSA(1024)	Au=RSA	Enc=RC4	56	Mac=SHA1
EXP1024-RC4-MD5	SSLv3	Kx=RSA(1024)	Au=RSA	Enc=RC4	56	Mac=MD5
EXP-EDH-RSA-DES-CBC-SHA	SSLv3	Kx=DH(512)	Au=RSA	Enc=DES	40	Mac=SHA1
EXP-EDH-DSS-DES-CBC-SHA	SSLv3	Kx=DH(512)	Au=DSS	Enc=DES	40	Mac=SHA1
EXP-DES-CBC-SHA	SSLv3	Kx=RSA(512)	Au=RSA	Enc=DES	40	Mac=SHA1
EXP-RC2-CBC-MD5	SSLv3	Kx=RSA(512)	Au=RSA	Enc=RC2	40	Mac=MD5
ADH-DES-CBC3-SHA	SSLv3	Kx=DH	Au=None	Enc=3DES	168	Mac=SHA1
ADH-DES-CBC-SHA	SSLv3	Kx=DH	Au=None	Enc=DES	56	Mac=SHA1
EXP-ADH-DES-CBC-SHA	SSLv3	Kx=DH(512)	Au=None	Enc=DES	40	Mac=SHA1
ADH-RC4-MD5	SSLv3	Kx=DH	Au=None	Enc=RC4	128	Mac=MD5
EXP-ADH-RC4-MD5	SSLv3	Kx=DH(512)	Au=None	Enc=RC4	40	Mac=MD5
EXP-RC2-CBC-MD5	SSLv2	Kx=RSA(512)	Au=RSA	Enc=RC2	40	Mac=MD5
EXP-RC4-MD5	SSLv3	Kx=RSA(512)	Au=RSA	Enc=RC4	40	Mac=MD5
EXP-RC4-MD5	SSLv2	Kx=RSA(512)	Au=RSA	Enc=RC4	40	Mac=MD5

D Troubleshooting

Introduction

This chapter provides information that can help resolve problems with the Device Server.

Hardware Problems

If the Device Server Power/Ready LED is red and stays red for over 10 seconds, you have a hardware problem that might to require factory service. First, try the following:

- In Console mode for desktop models or viewing the Console port in rack mount models, see if you need to reload the firmware, which can be found either on the CD-ROM that came with the Device Server or on the Perle website, **`www.perle.com/downloads/serial.shtml`**.
- If the bootloader option does not appear when you reboot the Device Server (to load new firmware), you need to make arrangements to return the Device Server.

If you purchased the Device Server less than 30 days before this problem appears, contact your distributor; otherwise, see the Perle web site (**`www.Perle.com`**) for factory service information. Note: no factory service can be done on a Device Server that has not been registered.

Power/Ready LED continues to flash green in Desktop models

This is not an error, the Power/Ready LED will flahs green when serial port 1 is in Console Mode.

Communication Issues

General communication checks and practices are as follows:

- Are your cables connected and correctly configured? If you are using EIA-232, see ***EIA-232 Cabling Diagrams*** on page 63 to verify that your cables are correctly configured.
- Ping your host? If you can ping but packet loss is reported, ping another host/device on the same network. This will tell you whether the problem is specific to the host/device or general to the network.
- After entering or changing IP information for your Device Server, *reboot* the Device Server (does not apply when using BOOTP or DHCP). Once the Device Server has rebooted, other network devices should be able to communicate with it (ping, telnet, etc.). Also, protocols such as ARP and proxy-ARP will work properly.
- Use the **`show routes`** command (command line only) or view the **Routes** statistics. Is there a route to the host?
- If the WebManager or DeviceManager cannot communicate with the Device Server, verify that the **Server Services HTTP** and/or **HTTPS** are enabled for WebManager and **DeviceManagerD** is enabled for DeviceManager. If you are using only HTTPS, the connection URL must start with **`https://`**.

DeviceManager Problems

Error Message: **16 bit Windows Subsystem - C:\WINDOWS\SYSTEM32\AUTOEXEC.NT. The system file is not suitable for running MS-DOS and Microsoft Windows applications. Choose 'Close' to terminate the application.**

The error message can be misleading, because it is displayed even if the **`AUTOEXEC.NT`** file is actually missing.

To verify whether you have the file, type **`%windir%/system32/`** in the address bar of an Explorer window. If there is no **`AUTOEXEC.NT`** file proceed as follows:

1. Browse to **`%windir%/repair/`** (usually **`C:\WINDOWS\repair`**).
2. Right-click and Copy the **`AUTOEXEC.NT`** file.
3. Browse to **`%windir%/system32/`** (usually **`C:\WINDOWS\System32`**).
4. Right-click inside the window and Paste the file.

The error condition described here may also be the result of corruption of the **`AUTOEXEC.NT`** file, in which case the above procedure may be helpful to restore a valid file.

If the above procedure does not fix the DeviceManager installation problem, see http://support.microsoft.com/?kbid=324767 for the official Microsoft explanation.

Host Problems

Cannot access a host by name:

- If using DNS or if DNS is required, ensure a nameserver is configured on your Device Server and is accessible (ping it).
- If not using DNS, verify that the host is configured in the **Host Table**. Check access to the host by pinging it using the host's IP address.

Cannot access a host on a local network, verify:

- The network address is correct.
- The subnet mask is set correctly and reflects the network configuration.
- The broadcast address is set correctly and reflects the network configuration.

Cannot access a host on a remote network:

- Use the **`show route`** command to verify that there is a route to the remote host. If no gateway is specified, verify that a default gateway is specified. Ping the default gateway to check if it is working.
- Consider the situation beyond the gateway; for example, are intermediate gateways and the remote host available? Also, check the messages returned by the **`ping`** command; for example, that a particular host or gateway is unreachable.

Gateways added into the gateway table are ignored by the Device Server:

- Have you used BOOTP and entered a single static gateway in the bootptab file entry? If yes, the other gateways will be ignored.

Access to host lost after a few minutes.

- If the route to this host goes through routers, make sure those routers are all sending RIP packets across the networks.

RADIUS Authentication Problems

User is waiting up to 60 seconds before login is accepted or denied and Authentication is set to RADIUS. User has entered User Name and Password, and has pressed Enter.

- Check RADIUS configuration of primary and secondary authentication/accounting hosts specified, if you have retry and timeout values greater than the default, the Device Server will be spending time trying each of these hosts and keeping the user waiting.
- Adjust RADIUS configuration: specify just one host, reduce **Timeout** and **Retry** values to the default or less than default.

You cannot progress beyond the login and password prompts when authentication is set to RADIUS:

- On the RADIUS host, check the secret (password), you should see it displayed in clear text in the RADIUS clients file. If you are unsure whether it is the same secret which you entered in the Device Server, go to the Device Server and re-enter a new secret.
- On the RADIUS host, verify that there is only one entry for a particular user; do not have multiple entries of the same user name (even if the passwords are different).

Login Problems

You cannot obtain a login on *any* of the serial ports

- Connect via the Admin port and check the settings of the front-mounted ports; they have probably been set to ‘direct’ or ‘silent’ telnet/rlogin.

You have lost or don't know your password (as Admin user).

- You must reset the Device Server to its factory default settings using the **Reset** switch on the rear panel. There is no procedure to access the Device Server without a password.

Problems with Terminals

The following section concerns problems with the appearance of data on your terminal screen.

The Device Server logs me out after a few minutes:

- Check the **Idle Timer** value set for the user. The default setting for the **Idle Timer** for all users is 0 seconds (does not timeout).

Corrupt data.

- Check your line settings (baud rate, stop bits, etc.)

Missing data.

- Verify that the same type of flow control is set in both your terminal and on the Device Server’s port.

Error message `not permitted on a dumb terminal` after typing the CLI command `screen`.

- Set your **Line** to **`Termtype`** VT100, ANSI or WYSE60 (or other form of terminal emulation, if you have downloaded one). The default line type in the Device Server is **Dumb**, which does not support the graphics characters necessary to view the text-based menus.

Screen corruption when using the text-based menu system.

- Verify that the terminal setup in the Device Server matches your terminal.
- Verify that entries in the term file match your terminal setup.
- If using a PC/computer, verify that the type of terminal emulation selected in your application matches those supported by the Device Server.

When using the function keys on your keyboard, nothing happens or your sessions keep swapping.

- Change your **Hotkey Prefix** character. The function keys on the keyboards of some terminals (like WYSE60) send character sequences which begin with **^a**; unfortunately, **^a** is also the default **Hotkey Prefix**, which you use to switch between sessions. A valid alternative would be **^b** (hex=02). If you are the system administrator, you can change any user's **Hotkey Prefix** character.

When using a downloaded terminal definition, you are having problems using arrow keys.

- Use Ctrl-K, Ctrl-J, Ctrl-H and Ctrl-L for up, down, left and right respectively.

When switching from a session back to the text menus, both screen images are superimposed.

- Press **^r** to redraw the screen.

`INIT: Error in terminal file <`*`filename`*`>`

- This error indicates that you have exceeded the 80 character limit for one or more of the terminal capabilities defined in the reported file.

`INIT: Error on line `*`n`*` in terminal file <`*`filename`*`>`

- You have omitted the **=** sign from the reported line.

Unknown IP Address

You have a Device Server already configured and you do know your password, but have lost, misconfigured, or don't know the IP address of the Device Server, and you cannot obtain a login.

- If the Device Server resides within the local network segment, you can use DeviceManager to find the Device Server.
- You can connect directly to the serial port of the Device Server, as explained in ***Using a Direct Connection*** on page 55.

DHCP/BOOTP Problems

Messages: `host name too long` or `filename too long`.

- The Device Server can only accept host names of 14 characters or file names of 64 characters, so verify that you are not attempting to pass a string that is longer than those maximums.

DHCP or BOOTP have been set up to configure my Device Server, but does not seem to have done anything.

- Check that the server DHCP/BOOTP service is set to on, if not set it to on and reboot.
- Check that your BOOTP server is configured for your Device Server or that your DHCP server has an active lease pool (scope) with at least 1 free IP address.

You observe TFTP errors when the Device Server boots, for example:

```
TFTP: File not found : filename
TFTP: Timed out
```

This has a number of causes, including:

- The file names you specified to DHCP/BOOTP do not exist or are in the wrong place.
- The server for any of the downloadable files in your bootfile has no TFTP server running.
- Verify that lease data in your DHCP server manager is correct.
- Reset or restart the DHCP server.

Callback Problems

User Callback is On, and a number is configured for the line, but the Device Server is not calling the user back:

- Verify that the phone number is entered under the user (not the line).
- Verify that the callback **Phone Number** is valid.
- Verify that the modem at the user's end is set to 'auto-answer'.

Language Problems

In a customised language, the text strings appear in the wrong place in the Menu, CLI, or WebManager.

- Check the original ASCII text file you used to translate to your customised language. The sequence of the line much match exactly (be aware that comments don't affect line sequence, but can affect the actual line that the strings appear on). So, if you strip out all comments, if the original file says line 1000 should be string **none**, then line 1000 (stripped of comments) should be the translated version of **none**.

Modem problems

The Device Server is not initializing the modem.

- Check your **Line Service** is set to **SLIP** or **PPP**. If your line service is set to any other type, the Device Server will not initialize a modem. You will need to configure the modem manually.

PPP problems

The link fails on start-up when there are remote IP addresses set for both a user (Framed IP value) and a line (Remote IP address).

- Check the IP address set for the user; this is used in preference to the IP address set for a line. If there is a problem with the user's IP address, negotiation will fail; the Device Server will *not* use the line's IP address as an alternative.

The link fails on start-up and security (either PAP or CHAP) is enabled on the line.

- Check the remote client/device has the same setting; that is, PAP if the Device Server is using PAP. The Device Server does not perform negotiation with the remote end over PAP or CHAP.

At the remote end, the client software locks up when security (CHAP) is enabled on the line.

- Disable CHAP re-challenge parameter (challenge_interval) in the Device Server. Some PPP client software does not work when receiving CHAP re-challenges.

PPP is not running successfully over your 485 half-duplex environment.

- PPP is incompatible with half-duplex; it must be run over a full-duplex environment.

Printing Problems

The print job fails to print on the device attached to the serial port.

- On the line where the printer is attached, set **Line Service** to **Printer**. Print jobs will not print when the line service is set incorrectly.

When using RCP, the network host receives a rejection message from the Device Server. The result is that the print job does not take place.

- Print using LPD

  or

- Modify the printer interface scripts on the network host to overcome this weakness of RCP. The modification will force the network host to continue trying to send the print job when the Device Server's printer port is busy.

Long Reboot Cycle

Rebooting the Device Server takes a long time.

If you are not using DHCP/BOOTP, disable this within the Server Services; otherwise, the Device Server waits to timeout for a request to DHCP/BOOTP.

SSL/TLS

If you are experiencing problems obtaining a successful SSL/TLS connection, you can set your **Syslog Level** to **Notice** and view the syslog for the following messages:

Line not SSL enabled. Abort connection when a user who is configured for **Service SSL_RAW** tries to login on the serial port.

The user has been configured for an **SSL_RAW** connection, but the line has not been configured to enable SSL. To resolve this, either enable the line for SSL or change the user's **Service** to **TCP_CLEAR** if SSL is not wanted.

Could not obtain peer's certificate.

- User has selected a cipher key exchange of ADH (anonymous Diffie-Hellman) and enabled Peer verification. ADH does not use certificates so they will not be sent in an SSL/TLS handshake. Disable Peer Verification or change to a cipher suite that uses certificates.
- User has selected Peer Verification on the configured SSL/TLS server and has not configured a certificate for the client. Either disable peer verification on the SSL/TLS server or configure a certificate for the SSL/TLS client.

SSL_accept failed on the SSL/TLS server device.

- The device has failed to accept an SSL/TLS connection on top of a TCP connection that has just been established. This could indicate that the peer from which TruePort is trying to accept a connection from is not configured for SSL/TLS. Verify that the peer has been configured for an SSL/TLS client connection.

Certificate did not match configuration

- The message is displayed when **Validate Peer Certificate** has been enabled, but the configured **Validation Criteria** does not match the corresponding data in the certificate received from the peer. The data configured must match exactly to the data in the certificate. The data is also case sensitive.

unknown protocol message when trying to make an SSL/TLS connection

- This will be displayed when both sides of the TCP connection are configured as SSL/TLS clients. Change one of the end points to act as an SSL/TLS server.
- One of the endpoints is not configured for SSL/TLS. Make sure both endpoints are configured for SSL/TLS, verify that one is a client and the other is a server.

tlsv1 alert handshake failure or **sslv3 alert handshake failure**

- The remote site has an SSL/TLS error and is sending this message with an alert message. Look at the error messages on the remote end and fix the problem indicated.

I/O Models

An I/O Digital or Relay controlled motor is starting/stopping

- Digital and Relay channels have automatically resetting fuses, meaning that if the circuit gets overloaded and the fuse blows, it will automatically reset when the circuit cools down.

An A4R2 model is starting/stopping

- The A4R2 model can run at 55 degrees Celsius ambient temperature when the input voltage is 22VDC or below. If the input voltage exceeds 22VDC, the maximum ambient temperature will drop into the range of 45-50 degrees Celsius to run successfully.

E Utilities

Introduction

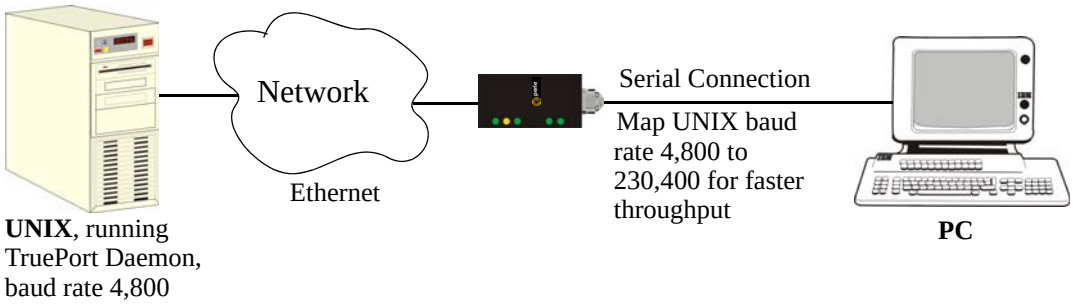
This chapter provides information on the TruePort and Decoder utilities.

TruePort

TruePort is a com port redirector utility for the Device Server. It can be run in two modes:

- **TruePort Full mode**—This mode allows complete device control and operates exactly like a directly connected serial port. It provides a complete COM port interface between the attached serial device and the network.
- **TruePort Lite mode**—This mode provides a simple raw data interface between the device and the network. Although the port will still operate as a COM port, control signals are ignored. In this mode, the serial communications parameters must be configured on the Device Server.

You use TruePort when you want to connect extra terminals to a server using a Device Server rather than a multi-port serial card. TruePort is especially useful when you want to improve data security, as you can enable an SSL/TLS connection between the TruePort host port and the Device Server. When run on UNIX, TruePort allows you to print directly from a terminal to an attached printer (transparent printing). You can also remap the slow baud rate of your UNIX server to a faster baud rate, as shown below.



Currently, TruePort is supported on Linux, Windows, SCO, Solaris, and others. For a complete list of of supported operating systems, see the Perle website.

For more information, see the *TruePort User Guide* or the *TruePort Installation and Configuration Guide for Windows NT* on the CD-ROM.

Accessing I/O Data Via TruePort

Introduction

Analog and Digital I/O data, as well as output control, can be accessed in several ways. To have access from an application running on a workstation or server, the I/O Applications Program Interface (API) provided within Trueport can be used. This API uses a command/response format to get or set data on each individual I/O channel register. A sample program (**`ioapiotp.c`**) demonstrating typical usage can be found on the IOLAN product CD-ROM.

Setup

After TruePort has been properly installed and configured on the workstation or server and initiated from the application, it will setup a connection to the appropriate IOLAN. It will then be available to relay commands to the IOLAN and communicate responses back the application. TruePort will create a COM port to which the application can write commands to and read responses from. Since all communications are done via this COM port, the application need only use standard serial communication interface calls.

The following steps should be taken:

1. Install the Trueport software on the server or workstation on which the application will be running.
2. Configure the virtual communication port (COM) (see Trueport User Guide for details)
3. Run the application. Typically the application will:
   1. Open the COM port.
   2. Send Commands, to the COM port using standard write commands.
   3. Read Responses from the COM port using standard read commands.

      Note: All commands are forwarded to the IOLAN over the network where the specific I/O channel registers are modified or read, and then responses are sent back to TruePort where they will be made available to be read from the COM port.

   4. Once the desired operations are completed, the COM port can be closed.

Format of API Commands

There are two groups of commands:

- **Get Commands**—Retrieve values of the I/O channel registers
- **Set Commands**—Set \values on the I/O channel registers.

Note: All commands need to be written to the COM port as a single write.

I/O Channel registers are all assigned unique addresses, which need to be referenced in all of the commands. Please refer to the documentation specific you the applicable mode, for the list and addresses of all the registers.

Model	Go to...
A4	***A4/T4 Registers* on page 111**
T4	***A4/T4 Registers* on page 111**
A4D2	***A4D2/A4R2 Registers* on page 112**
A4R2	***A4D2/A4R2 Registers* on page 112**
D4	***D4/D2R2 Registers* on page 113**
D2/R2	***D4/D2R2 Registers* on page 113**

Get Commands

The following tables show the general structure to be used for Get commands.

Note: Numeric values provided in the API documentation are in Hexadecimal (Hex) format.

Command Format

Byte(s)	# of Bytes	Value
1	1	Command Code: • 0x01 – Get “coils” (Boolean register) • 0x03 – Get “holding registers” (R/W registers) • 0x04 – Get “input registers” (R only register)
2-3	2	Starting register number (see ***A4/T4 Registers*** on page 111, ***A4D2/A4R2 Registers*** on page 112, or ***D4/D2R2 Registers*** on page 113 for this value).
4-5	2	Number of registers to read. If this value is greater than 1, the response will contain the values of multiple consecutive registers.

Response Format

Byte(s)	# of Bytes	Value
1	1	Command that this is a response to. If an error has been detected, the command value will have the high bit set (OR with 0x80). For example: The command is 0x04, so the command field in the response would be 0x84.
2	1	Length of data (in bytes) starting in next byte.
3-n	n	Requested register values.

Example 1: Read the status of the first digital input (DI1) on a D2R2 unit.

DI1 sensor is a coil register with the decimal value of 6145 (hex 0x1801).

Request: 0x01 0x18 0x01 0x00 0x01

Response: 0x01 0x01 0x**01** (Digital input 1 is active)

Example 2: Read the values for the Inactive Signal Width, Active Signal Width, and Pulse count for the second digital output (DO2) on a D4 unit.

DO2, Inactive Signal Width is a holding register with the decimal value of 6210 (hex 0x1842).

Request: 0x03 0x18 0x42 0x00 0x03

Response: 0x03 0x06 0x00 0x0A 0x00 0x11 0x00 0x0F
(Inactive =10*100ms, Active= 17*100ms, and Pulse count = 15)

Example 3: Read the raw current, minimum and maximum values of the third Analog input (A3) on an A4D2 unit.

A3 current raw value is an input register with the decimal value of 2150 (hex 0x0866).

Request: 0x04 0x08 0x86 0x00 0x03

Response: 0x04 0x06 0x10 0x03 0x0F 0x30 0x10 0x20
(Current = 0x1003, Minimum = 0x0F30, and Maximum = 0x1020)

Set Commands

The following tables show the general structure to be used for set commands.

Note: Numeric values provided in the API documentation are in Hexadecimal (Hex) format.

Command Format

Byte(s)	# of Bytes	Value
1	1	Command Code (in hex): ● 0x0F – Set “Boolean registers” (R/W coils) ● 0x10 – Set “holding registers” (read/write registers)
2-3	2	Starting register number (see ***A4/T4 Registers*** on page 111, ***A4D2/A4R2 Registers*** on page 112, or ***D4/D2R2 Registers*** on page 113 for this value).
4-5	2	Number of registers to set. If this value is greater than 1, the response will contain the values of multiple consecutive registers.
6	1	The length of the data (in bytes) to be written to the registers.
7-n	n	Data to be written to the registers. If accessing registers which are 2 or 4 bytes, the data is in Network order (Big endian) format (that is, MSB, LSB). For Boolean registers, the value field will be a bit field with the LSBit corresponding to the IO channel referenced by the starting register.

Successful Response Format

Byte(s)	# of Bytes	Value
1	1	Command code (from request).
2	2	Starting register number (see ***A4/T4 Registers*** on page 111, ***A4D2/A4R2 Registers*** on page 112, or ***D4/D2R2 Registers*** on page 113 for this value) from request.
4	2	Number of registers written.

Unsuccessful Response Format

Byte(s)	# of Bytes	Value
1	1	Command that this is a response to. If an error has been detected, the command value will have the high bit set (OR with 0x80). For example: The Command is 0x10, so the command field in the response would be 0x90.
1	1	Error code, see ***Error Codes*** on page 382.

Example 1: Turn on the first relay on a D2R2 unit.

The first relay (R1) is a digital out coil register with a decimal value of 6659 (hex 0x1A03).

Request: 0x0F 0x1A 0x03 0x00 0x01 0x01 0x01

Response: 0x0F 0x1A 0x03 0x00 0x01

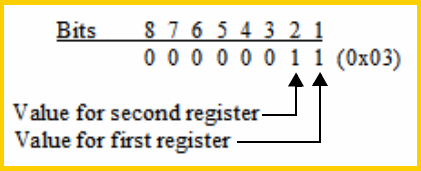
Example 2: Turn on the first and second relay on a D2R2 unit.

The first relay (R1) is a digital out coil register with a decimal value of 6659 (hex 0x1A03).

Request: 0x0F 0x1A 0x03 0x00 0x02 0x01 0x03 (03 = “00000011” which sets R1 and R2 to 1)

Response: 0x0F 0x1A 0x03 0x00 0x02

Note: When reading or writing consecutive “Boolean” (coils) registers, the values of the registers are combined into a single byte as shown by the example above. Two registers (coils) are being written but the length of the data is 1 byte. The one byte contains the value for both registers as follows:



Error Codes

Code	Name	Description
01	Illegal Function	The function code received in the query is not an allowable action for the server (or slave).
02	Illegal Data Address	The data address received in the query is not an allowable address for the server (or slave).
03	Illegal Data Value	A value contained in the query data field is not an allowable value for server (or slave).
04	Slave Device Failure	An unrecoverable error occurred while the server (or slave) was attempting to perform the requested action.

Decoder

If you are using **Port Buffering NFS Encryption**, you need to run the Decoder utility to view the port buffering logs. See the Readme file to install the Decoder utility on any of the following operating systems:

- Windows 98/NT/ME/2000/Server 2003/XP/Vista
- DOS
- Solaris x86
- Solaris Sparc 32-bit/64-bit

Linux x86 v2.4.x

F Accessories

Introduction

This chapter provides information about peripheral Device Server options that can be ordered separately from the product. Contact your sales representative to find out how to order the products listed in this appendix.

Installing a Perle PCI Modem Card

This sections describes how to install the Perle IOLAN PCI modem card in your SCS rack mount model. The location and brackets are slightly different for the 32-port and 48-port SCS rack mount models, but the basic installation concept is the same. The PCI modem bracket is found on the serial side of the 32-port model and the LED side of the 48-port model.

Note: Do not touch any of the components within the SCS Device Server while performing the PCI modem card installation.

1. Unscrew the six screws on the top of the SCS Device Server.

2. Unscrew the four screws along the bottom of the serial side of the SCS Device Server. On the SCS 32-port model, this includes the screw that is at the bottom of the PCI face plate.

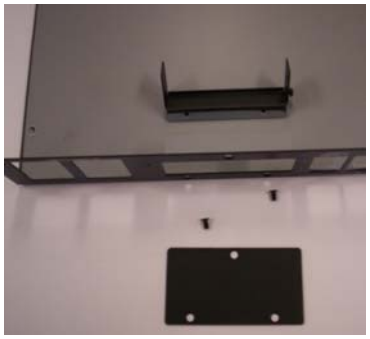
3. Slide the top of the Device Server off of the chassis.

4. Carefully holding the bracket just behind the face plate, unscrew the two screws at the top of the 32-port removable face plate or the two side screws of the 48-port removable face plate of the piece you just took off.

32-port model

48-port model

The 32-port model is displayed below with the face plate and bracket taken apart.



5. Unscrew the screw in the bracket. The 32-port bracket is shown below.

6. Slide the PCI modem card into the bracket.

32-port model

48-port model

7. The black bracket should then fit on the inside of the PCI modem bracket. Align the modem card bracket and then insert the screw and tighten it to keep it firmly in place.

32-port model

48-port model

Note: You must attach the bracket to the PCI modem card before you slide it into the PCI slot.

8. Slide the PCI modem card into the PCI slot.

9. You can now replace the top of the Device Server chassis by aligning it and sliding it into the base. You can throw away the face plate, as you will not be needing it.

32-port model

48-port model

10. Replace all the screws on the top and the serial side of the Device Server.

Starter Kit (Adapters/Cable)

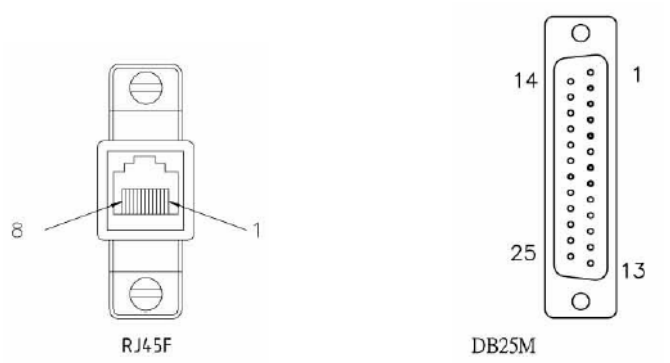
The IOLAN Starter Kit includes the following for every model except the SCS48C (see ***SCS48C Starter Kit (Adapters/Cable)*** on page 391 for the Device Server Cisco model):

- ***RJ45F to DB25M DTE Crossover Adapter***
- ***RJ45F to DB25M DCE Modem Adapter***
- ***RJ45F to DB25F DTE Crossover Adapter***
- ***RJ45F to DB9M DTE Crossover Adapter***
- ***RJ45F to DB9F DTE Crossover Adapter***
- ***Sun/Cisco RJ45MgRJ45F Adapter for Rack Mount Models***

The adapters/cable can be purchased as a kit or individually.

RJ45F to DB25M DTE Crossover Adapter

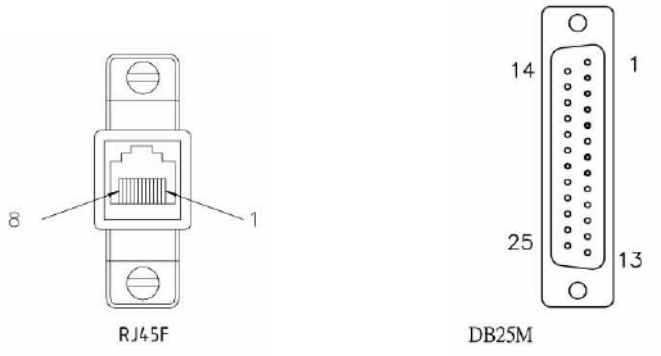
The following diagram shows the IOLAN Device Server RJ45F→DB25M DTE crossover adapter pinouts. This is model number DBA0011.



RJ45F	DB25M DTE
(TxD) 4	3 (RxD)
(RxD) 5	2 (TxD)
(GND) 6	7 (GND)
(DTR) 8	6 (DSR) 8 (DCD)
(DSR) 3	20 (DTR)
(RTS) 2	5 (CTS)
(CTS) 7	4 (RTS)

RJ45F to DB25M DCE Modem Adapter

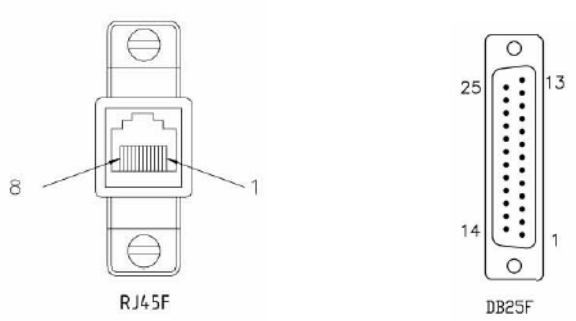
The following diagram shows the IOLAN Device Server RJ45F→DB25M DCE modem adapter pinouts. This is model number DBA0013.



RJ45F	DB25M
(TxD) 4	2 (TxD)
(RxD) 5	3 (RxD)
(GND) 6	7 (GND)
(DTR) 8	20 (DTR)
(DSR) 3	6 (DSR)
(DCD) 1	8 (DCD)
(RTS) 2	4 (RTS)
(CTS) 7	5 (CTS)

RJ45F to DB25F DTE Crossover Adapter

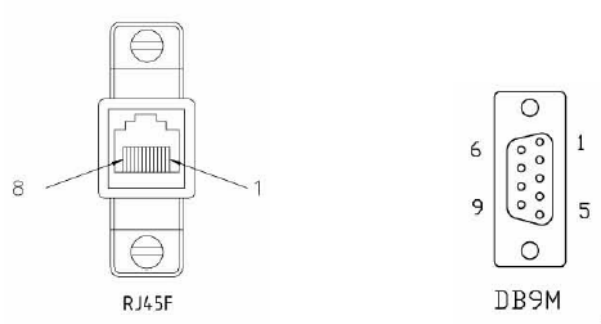
The following diagram shows the IOLAN Device Server RJ45→DB25F DTE crossover adapter pinouts. This is model number DBA0010.



RJ45F	DB25F
(TxD) 4	3 (RxD)
(RxD) 5	2 (TxD)
(GND) 6	7 (GND)
(DTR) 8	6 (DSR) 8 (DCD)
(DSR) 3	20 (DTR)
(RTS) 2	5 (CTS)
(CTS) 7	4 (RTS)

RJ45F to DB9M DTE Crossover Adapter

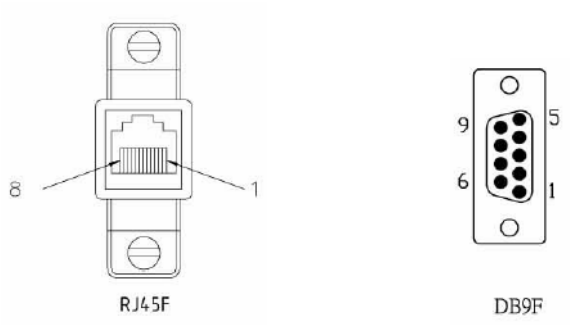
The following diagram shows the IOLAN Device Server RJ45→DB9M crossover adapter pinouts. This is model number DBA0021.



RJ45F	DB9M
(TxD) 4	2 (RxD)
(RxD) 5	3 (TxD)
(GND) 6	5 (GND)
(DTR) 8	1 (DCD) 6 (DSR)
(DSR) 3	4 (DTR)
(RTS) 2	8 (CTS)
(CTS) 7	7 (RTS)

RJ45F to DB9F DTE Crossover Adapter

The following diagram shows the IOLAN Device Server RJ45F→DB9F crossover adapter pinouts. This is model number DBA0020.



RJ45F	DB9F
(TxD) 4	2 (RxD)
(RxD) 5	3 (TxD)
(GND) 6	5 (GND)
(DTR) 8	1 (DCD) 6 (DSR)
(DSR) 3	4 (DTR)
(RTS) 2	8 (CTS)
(CTS) 7	7 (RTS)

Sun/Cisco RJ45M→RJ45F Adapter for Rack Mount Models

This is an RJ45M→RJ45F Sun/Cisco adapter. The RJ45M end connects to a serial port on the Device Server. Use a straight-through cable between the RJ45F end of the adapter and the Sun/Cisco router Console port. This model number is DBA0031.

IOLAN DS RJ45M	Sun/Cisco RJ45F
(RTS) 2	8 (CTS)
(DSR) 3	2 (DTR)
(TxD) 4	6 (RxD)
(RxD) 5	3 (TxD)
(GND) 6	4 (GND) 5 (GND)
(CTS) 7	1 (RTS)
(DTR) 8	7 (DSR)

SCS48C Starter Kit (Adapters/Cable)

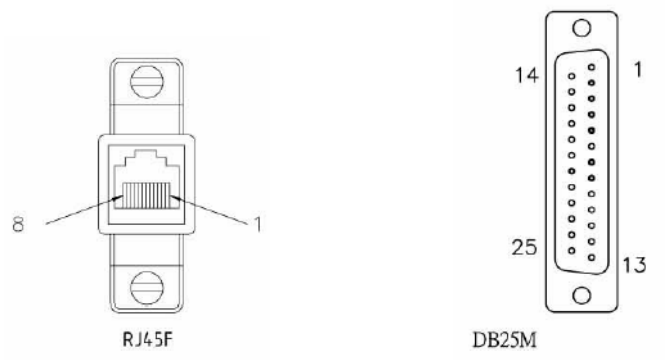
The IOLAN Starter Kit includes the following for the SCS48C (Cisco) model:

- ***RJ45F to DB25M DTE Crossover Adapter***
- ***RJ45F to DB25M DCE Modem Adapter***
- ***RJ45F to DB25F DTE Crossover Adapter***
- ***RJ45F to DB9M DTE Crossover Adapter***
- ***RJ45F to DB9F DTE Crossover Adapter***
- ***Sun/Cisco Roll-Over Adapter for Rack Mount Models***

The adapters/cable can be purchased as a kit or individually.

RJ45F to DB25M DTE Crossover Adapter

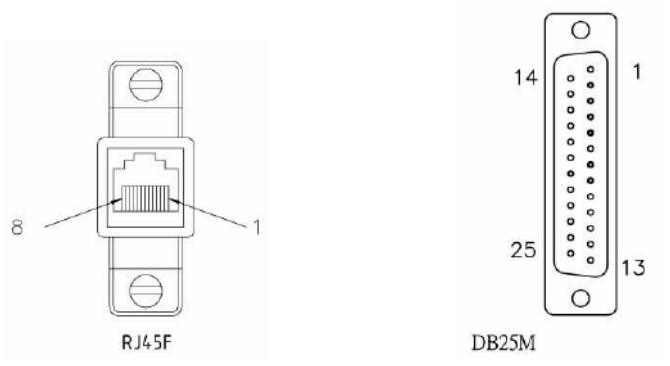
The following diagram shows the IOLAN Device Server RJ45F→DB25M DTE crossover adapter pinouts. This is model number DBA0011C.



RJ45F	DB25M DTE
(TxD) 3	3 (RxD)
(RxD) 6	2 (TxD)
(GND) 4 (GND) 5	7 (GND)
(DTR) 2	6 (DSR) 8 (DCD)
(DSR) 7	20 (DTR)
(RTS) 1	5 (CTS)
(CTS) 8	4 (RTS)

RJ45F to DB25M DCE Modem Adapter

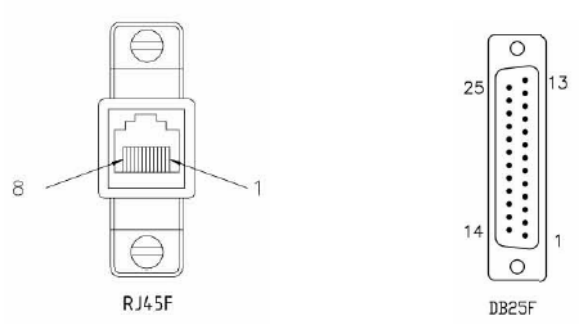
The following diagram shows the IOLAN Device Server RJ45F→DB25M DCE modem adapter pinouts. This is model number DBA0013C.



RJ45F	DB25M
(TxD) 3	2 (TxD)
(RxD) 6	3 (RxD)
(GND) 4 (GND) 5	7 (GND)
(DTR) 2	20 (DTR)
(DSR) 7	8 (DCD)
(RTS) 1	4 (RTS)
(CTS) 8	5 (CTS)

RJ45F to DB25F DTE Crossover Adapter

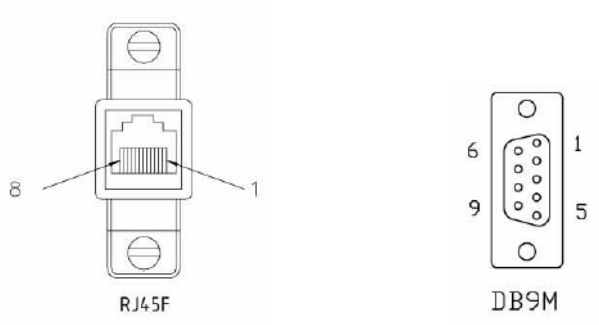
The following diagram shows the IOLAN Device Server RJ45→DB25F DTE crossover adapter pinouts. This is model number DBA0010C.



RJ45F		DB25F
(TxD) 3	———	3 (RxD)
(RxD) 6	———	2 (TxD)
(GND) 4 (GND) 5	———	7 (GND)
(DTR) 2	———	6 (DSR) 8 (DCD)
(DSR) 7	———	20 (DTR)
(RTS) 1	———	5 (CTS)
(CTS) 8	———	4 (RTS)

RJ45F to DB9M DTE Crossover Adapter

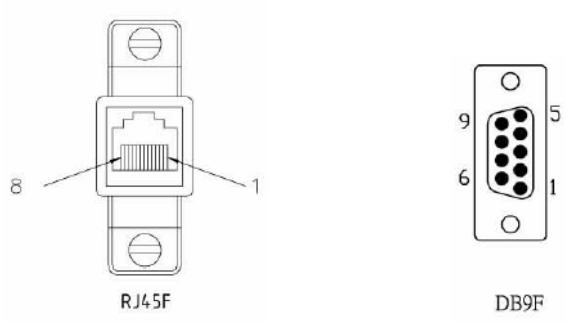
The following diagram shows the IOLAN Device Server RJ45→DB9M crossover adapter pinouts. This is model number DBA0021C.



RJ45F	DB9M
(TxD) 3	2 (RxD)
(RxD) 6	3 (TxD)
(GND) 4 (GND) 5	5 (GND)
(DTR) 2	1 (DCD) 6 (DSR)
(DSR) 7	4 (DTR)
(RTS) 1	8 (CTS)
(CTS) 8	7 (RTS)

RJ45F to DB9F DTE Crossover Adapter

The following diagram shows the IOLAN Device Server RJ45F→DB9F crossover adapter pinouts. This is model number DBA0020C.



RJ45F		DB9F
(TxD) 3	———	2 (RxD)
(RxD) 6	———	3 (TxD)
(GND) 4 (GND) 5	———	5 (GND)
(DTR) 2	———	1 (DCD) 6 (DSR)
(DSR) 7	———	4 (DTR)
(RTS) 1	———	8 (CTS)
(CTS) 8	———	7 (RTS)

Sun/Cisco Roll-Over Adapter for Rack Mount Models

This is a RJ45M→RJ45F Sun/Cisco adapter. This model number is DBA0031C.

IOLAN SDS RJ45F		Sun/Cisco RJ45M*
1	———	8
2	———	7
3	———	6
4	———	5
5	———	4
6	———	3
7	———	2
8	———	1

*The Sun/Cisco RJ45M connector attaches to the Sun/Cisco Console port.

Glossary

This chapter provides definitions for Device Server terms.

BOOTP (BOOTstrap Protocol)	An Internet protocol that enables a diskless workstation to discover its own IP address, the IP address of a BOOTP server on the network, and a file to be loaded into memory to boot the machine. This enables the workstation to boot without requiring a hard or floppy disk drive.
Callback	A security feature where the Device Server calls back the User at a predetermined number defined in the User's account.
CHAP (Challenge Handshake Authentication Protocol)	Standard authentication protocol for PPP connections. It provides a higher level of security than PAP and should be used whenever possible. *see PAP*
Community (SNMP)	An SNMP community is the group that devices and management stations running SNMP belong to. It helps define where information is sent.
DHCP (Dynamic Host Configuration Protocol)	A TCP/IP protocol that provides static and dynamic address allocation and management.
Direct Connection	Connections that bypass the Device Server enabling the user to log straight into a specific host. A direct connection is recommended where a user logging in to the Device Server is not required.
Ethernet	A high-speed (10Mbps,100Mbps) cable technology that connects devices to a LAN, using one or more sets of communication protocols.
Fixed Callback	A method where there is a specific number defined to callback a user.
Local Authentication	Uses the user ID and password stored within the Device Server User database.
LPD	Line Printer Daemon. A printer protocol that uses TCP/IP to establish connections between printers and workstations on a network. The technology was developed originally for BSD UNIX and has since become the de facto cross-platform printing protocol.
Modem Initialization String	A series of commands sent to the modem by a communications program at start up. These commands tell a modem how to set itself up in order to communicate easily with another modem.
MOTD	Message of the day. This is defined by a file whose contents display when users log into the Device Server.
Multicast	The broadcasting of messages to a specified group of workstations on a LAN, WAN, or internet.
NAK (Negative Acknowledgment)	A communication control character sent by the receiving destination indicating that the last message was not received correctly.

PAP (Password Authentication Protocol)	Standard authentication protocol for PPP connections. *see CHAP*
RADIUS (Remote Authentication Dial In Users Services)	An open standard network security server that communicates with the PAP protocol.
Reverse Connection	Connections that originate from a host that go directly to a serial device through the Device Server.
RIP (Routing Information Protocol)	A protocol that allows gateways and hosts to exchange information about various routes to different networks.
Roaming Callback	A method where the client supplies the number for callback when they dial in.
RPC	Remote Procedure Call. A type of protocol that allows a program on one computer to execute a program on a server computer.
Silent Connection	Silent connections are the same as direct connections except that they are permanently established. The host login prompt is displayed on the screen. Logging out redisplays this prompt. Silent connections, unlike direct connections, however, make permanent use of pseudo tty resources and therefore consume host resources even when not in use.
SNMP (Simple Network Management Protocol)	A protocol for managing network devices.
Subnet/Prefix Bits	Identifies the device's IP address, which portion constitutes the network address and which portion constitutes the host address.

Index

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W