Secure Console Port Server SSH User Guide

Version 1.3.8 Revision 1b

Secure Console Port Server SSH User Guide

Version 1.3.8 Revision 1b

September, 2003

We believe the information in this manual is accurate and reliable. However, we assume no responsibility, financial or otherwise, for any consequences of the use of this product or manual. This manual is published by Black Box Corporation, which reserves the right to make improvements or changes in the products described in this manual as well as to revise this publication at any time and without notice to any person of such revision or change. The operating system covered in this manual is v1.3.8. All brand and product names mentioned in this publication are trademarks or registered trademarks of their respective holders.

Black Box, Secure Console Port Server SSH, LES 2800A-48, LES 2800A-32, LES 2800A-16, LES 2800A-8, LES 2800A-4, and LES 2800A-1 are registered trademark of Black Box Corporation.
Microsoft, Windows 95, 98, XP, ME, NT, and 2K are trademarks of Microsoft Corporation.
UNIX is a trademark of UNIX System Laboratories, Inc.
Linux is a registered trademark of Linus Torvalds.

Product Version 1.3.8 Revision 1b
Document Number 1.3.8-Draft 27b

Table of Contents

Preface

Chapter 1 - Introduction and Overview

Chapter 2 - Installation, Configuration, and Usage

Table of Contents

Chapter 3 - Additional Features

Table of Contents

Table of Contents

Appendix A - New User Background Information

Appendix B - Cabling, Hardware, and Electrical Specifications

Table of Contents

Appendix C - The pslave Configuration File

Appendix D - Linux-PAM

Appendix E - Software Upgrades and Troubleshooting

Table of Contents

Appendix F - Certificate for HTTP Security

Appendix G - Web User Management

Appendix H - Connect to Serial Ports from Web

Table of Contents

Appendix I - Examples for Configuration Testing

Appendix J - Billing Feature

Appendix K - Wiz Application Parameters

Appendix L - Copyrights

Table of Contents

Preface

Purpose

The purpose of this guide is to provide instruction for users to independently install, configure, and maintain the Secure Console Port Server SSH. This manual should be read in the order written, with exceptions given in the text. *Whether or not you are a UNIX user, we strongly recommend that you follow the steps given in this manual.*

Audience and User Levels

This guide is intended for the user who is responsible for the deployment and day-to-day operation and maintenance of the Secure Console Port Server SSH. It assumes that the reader understands networking basics and is familiar with the terms and concepts used in Local and Wide Area Networking. UNIX and Linux users will find the configuration process very familiar. It is not necessary to be a UNIX expert, however, to get the Secure Console Port Server SSH up and running. There are two audiences or user levels for this manual:

New Users

These are users new to Linux and/or UNIX with a primarily PC/Microsoft background. You might want to brush up on such things as common Linux/UNIX commands and how to use the vi editor prior to attempting installation and configuration. This essential background information appears in Appendix A - New User Background Information. It is recommended that New Users configure the Secure Console Port Server SSH using a Web browser, however, New Users can also configure the Secure Console Port Server SSH with vi, the Wizard or the Command Line Interface (CLI).

Power Users

These are UNIX/Linux experts who will use this manual mostly for reference. Power Users can choose between configuring the Secure Console Port Server SSH via Web browser, vi, Wizard, or CLI.

Each configuration task will be separated into a section (a clickable link on the PDF file) for each user type. Users then can skip to the appropriate level that matches their expertise and comfort level.

Preface

How to use this Guide

This guide is organized into the following sections:

- Chapter 1 - Introduction and Overview contains an explanation of the product and its default CAS setup. It also includes safety guidelines to be followed.
- Chapter 2 - Installation, Configuration, and Usage explains how the Secure Console Port Server SSH should be connected and what each cable is used for. It describes the basic configuration process to get the Secure Console Port Server SSH up and running for its most common uses.
- Chapter 3 - Additional Features is dedicated to users wanting to explore all available features of the Secure Console Port Server SSH. It provides configuration instructions for syslog, data buffers, authentication, filters, DHCP, NTP, SNMP, clustering, and sniffing.
- Appendix A - New User Background Information contains information for those who are new to Linux/UNIX.
- Appendix B - Cabling, Hardware, and Electrical Specifications has detailed information and pinout diagrams for cables used with the Secure Console Port Server SSH.
- Appendix C - The pslave Configuration File contains example files for the various configurations as well as the master file.
- Appendix D - Linux-PAM enables the local system administrator to choose how to authenticate users.
- Appendix E - Software Upgrades and Troubleshooting includes solutions and test procedures for typical problems.
- Appendix F - Certificate for HTTP Security provides configuration information that will enable you to obtain a Signed Digital Certificate.
- Appendix G - Web User Management covers default and optional configuration, and the addition/deletion of users, groups, and access limits.
- Appendix H - Connect to Serial Ports from Web enables this process, based on how the serial port is configured.
- Appendix I - Examples for Configuration Testing provides examples for testing the Secure Console Port Server SSH after configuration.

- Appendix J - Billing Feature explains how the LES 2800A-1 can also be simply used as an intermediate buffer to collect serial data (like billing tickets from a PABX), making them available for a posterior file transfer.
- Appendix K - Wiz Application Parameters contains all basic and custom wizard parameters.
- Appendix L - Copyrights lists details about applications that were incorporated into the product.
- The Glossary provides definitions for commonly-used terms in this manual.

Conventions and Symbols

This section explains the significance of each of the various fonts, formatting, and icons that appear throughout this guide.

Fonts

This guide uses a regular text font for most of the body text and `Courier` for data that you would input, such as a command line instruction, or data that you would receive back, such as an error message. An example of this would be:

```
telnet 200.200.200.1 7001
```

Hypertext Links

References to another section of this manual are hypertext links that are underlined (and are also blue in the PDF version of the manual). When you click on them in the PDF version of the manual, you will be taken to that section.

Glossary Entries

Terms that can be found in the glossary are underlined and slightly larger than the rest of the text. These terms have a hypertext link to the glossary.

Preface

Quick Steps

Step-by-step instructions for installing and configuring the Secure Console Port Server SSH are numbered with a summarized description of the step for quick reference. Underneath the quick step is a more detailed description. Steps are numbered 1, 2, 3, etc. Additionally, if there are sub-steps to a step, they are indicated as Step A, B, C, and are nested within the Step 1, 2, 3, etc. For example:

Step 1: Modify files.

You will modify four Linux files to let the Secure Console Port Server SSH know about its local environment.

Step A: Modify pslave.conf.

Open the file plsave.conf and add the following lines . . .

Parameter Syntax

This manual uses standard Linux command syntaxes and conventions for the parameters described within it.

Brackets and Hyphens (dashes)

The brackets ([])indicate that the parameter inside them is optional, meaning that the command will be accepted if the parameter is not defined. When the text inside the brackets starts with a dash (-) and/or indicates a list of characters, the parameter can be one of the letters listed within the brackets.

Example:

```
iptables [-ADC] chain rule-specification [options]
```

Ellipses

Ellipses (...) indicate that the latest parameter can be repeated as many times as needed. Usually this is used to describe a list of subjects.

Example:

```
ls [OPTION]... [FILE]...
```

Pipes

The pipe (|) indicates that one of the words separated by this character should be used in the command.

Example:

```
netstat {--statistics|-s} [--tcp|-t] [--udp|-u] [--raw|-w]
```

When a configuration parameter is defined, the Linux command syntax conventions will be also used, with a difference.

Greater-than and Less-than signs

When the text is encapsulated with the “<>” characters, the meaning of the text will be considered, not the literal text. When the text is not encapsulated, the literal text will be considered.

Spacing and Separators

The list of users in the following example must be separated by semicolons (;); the outlets should be separated by commas (,) to indicate a list or with dashes (-) to indicate range; there should not be any spaces between the values.

sXX.pmusers: The user access list. For example: jane:1,2;john:3,4. The format of this field is:

```
[<username>:<outlet list>][;<username>:<outlet list>...]
```

where <outlet list>'s format is:

```
[<outlet number>|<outlet start>-<outlet end>][,<outlet num-
ber>|<outlet start>-<outlet end>]...
```

Preface

Note Box Icons

Note boxes contain instructional or cautionary information that the reader especially needs to bear in mind. There are five levels of note box icons:

Tip. An informational tip or tool that explains and/or expedites the use of the Secure Console Port Server SSH.

Important! An important tip that should be read. Review all of these notes for critical information.

Warning! A very important type of tip or warning. Do not ignore this information.

DANGER! Indicates a direct danger which, if not avoided, may result in personal injury or damage to the system.

Security Issue. Indicates security-related information where it is relevant.

Introduction and Overview

The Secure Console Port Server SSH

The Secure Console Port Server SSH is line of Console Access Servers that allow both local and dial-in access for in-band and out-of-band network management. run an embedded version of the Linux operating system. Configuration of the is done by editing a few plain-text files, and then updating the versions of the files on the Secure Console Port Server SSH. The files can be edited using the vi editor provided or on another computer with the environment and text editor of your choice. The default "profile" of the Secure Console Port Server SSH is that of a Console Access Server.

You can access the Secure Console Port Server SSH via three methods:

- A console directly connected to the Secure Console Port Server SSH
- Telnet/ssh over a network
- A browser

And configure it with any of the following four options:

- vi
- Wizard
- Browser
- Command Line Interface (CLI) - only for certain configuration parameters

With the Secure Console Port Server SSH set up as a Console Access Server, you can access a server connected to the Secure Console Port Server SSH through the server's serial console port from a workstation on the LAN or WAN. There is no authentication by default, but the system can be configured for authentication to be performed by a Radius server, a TacacsPlus server, or even by a local database. Either telnet or ssh (a secure shell session) can be used. See Appendix A - New User Background Information for more information about ssh. The instructions in Chapter 2 - Installation, Configuration, and Usage will set up a fully-functional, default CAS environment. More options can be added after the initial setup, as illustrated in Chapter 3 - Additional Features.

Introduction and Overview

What’s in the box

There are several models of the Secure Console Port Server SSH. Black Box will ship either Cable Package #1 or #2 with the product according to current availability.

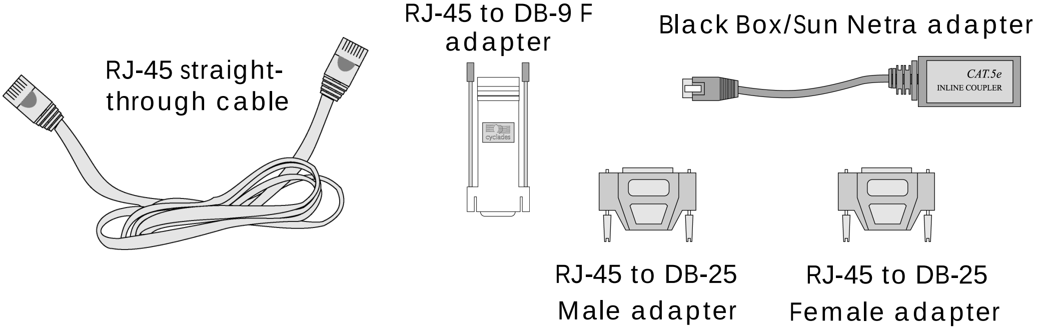

RJ-45 to DB-9 F adapter
Black Box/Sun Netra adapter
RJ-45 straight-through cable
cyclades
CAT.5e
INLINE COUPLER
RJ-45 to DB-25 Male adapter
RJ-45 to DB-25 Female adapter

Figure 1: Cable Package #1

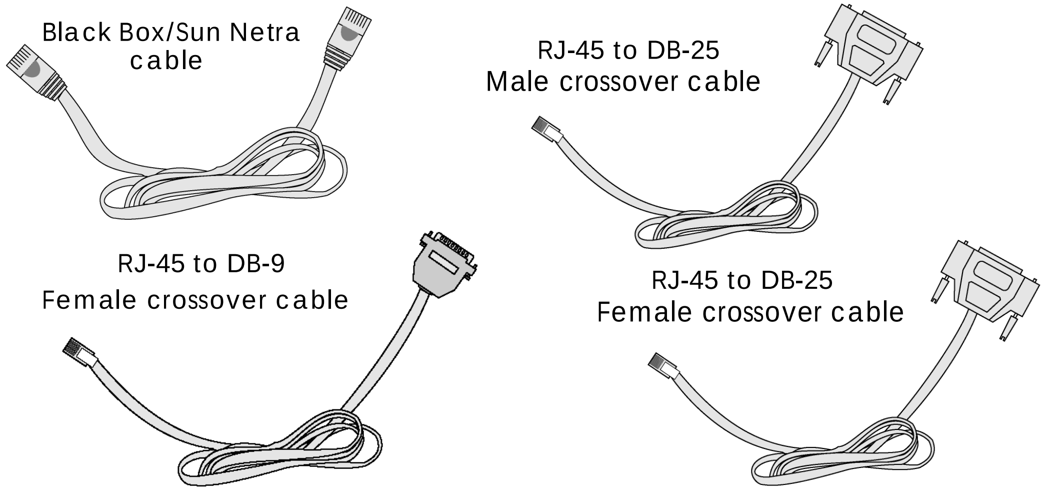

Black Box/Sun Netra cable
RJ-45 to DB-25 Male crossover cable
RJ-45 to DB-9 Female crossover cable
RJ-45 to DB-25 Female crossover cable

Figure 2: Cable Package #2

The following figures show the main units and accessories included in package.

Introduction and Overview

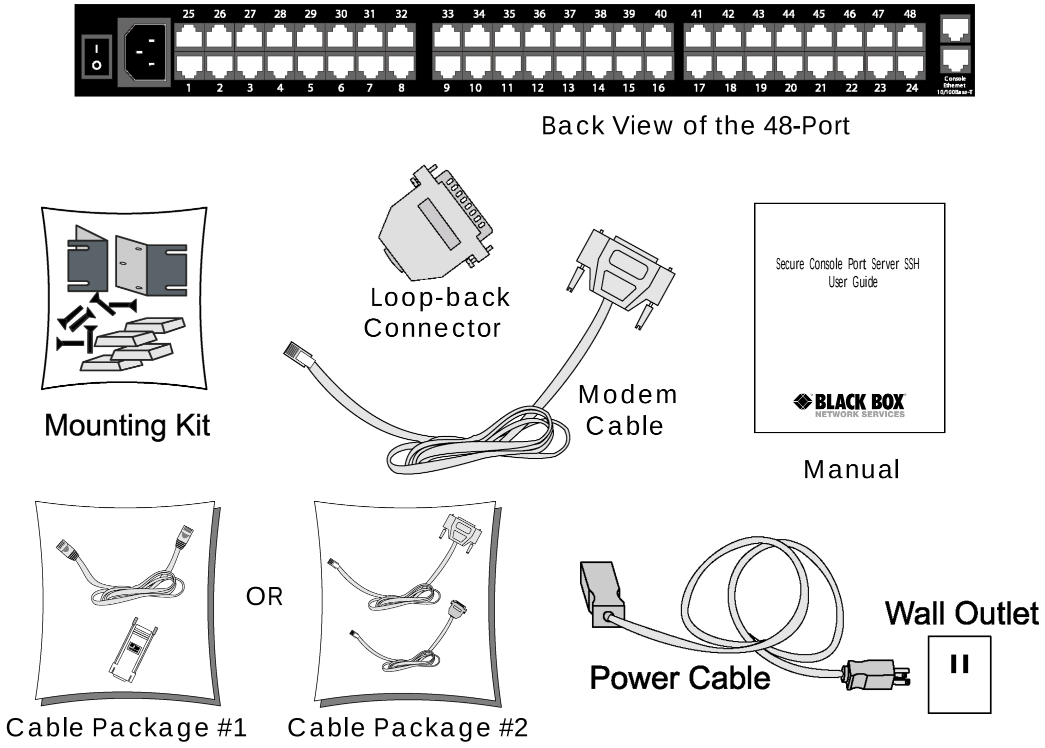


Figure 3: The Secure Console Port Server SSH 48-Port and cables

Introduction and Overview

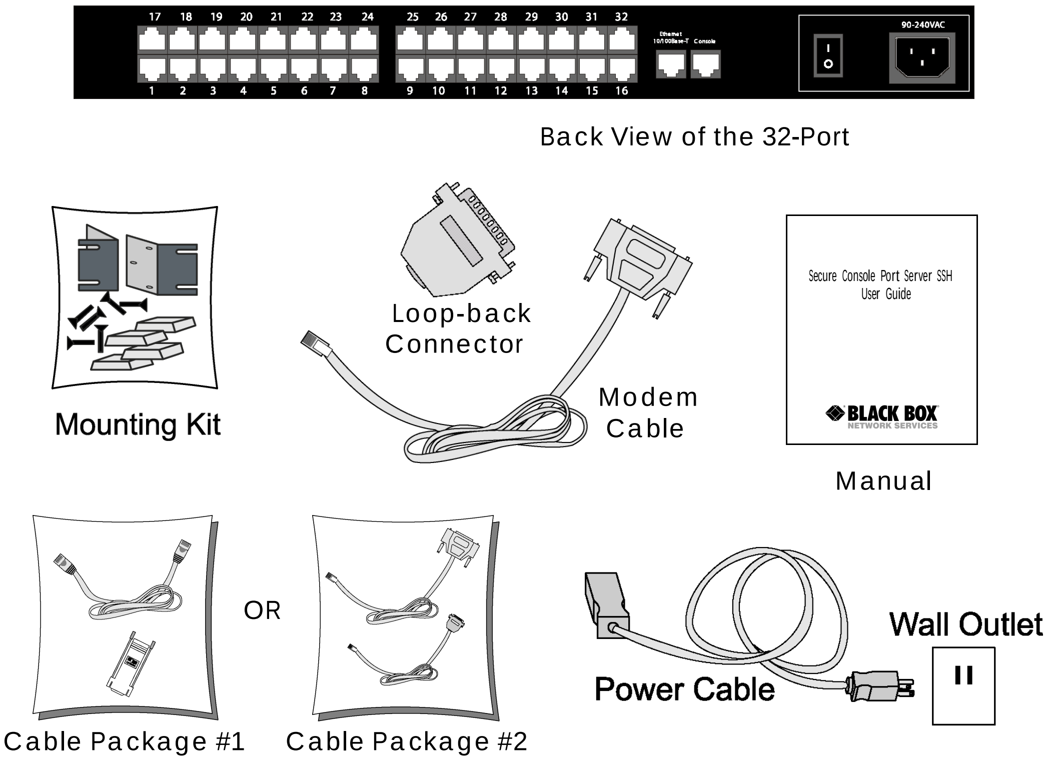


Figure 4: The Secure Console Port Server SSH 32-Port and cables

Introduction and Overview

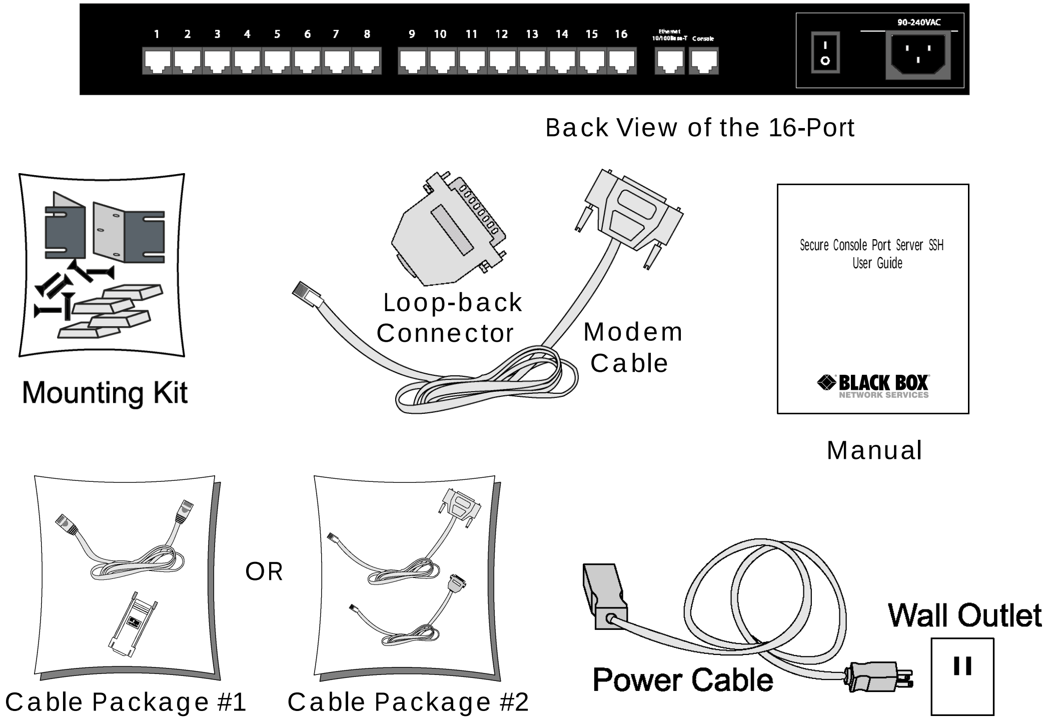


Figure 5: The Secure Console Port Server SSH 16-Port and cables

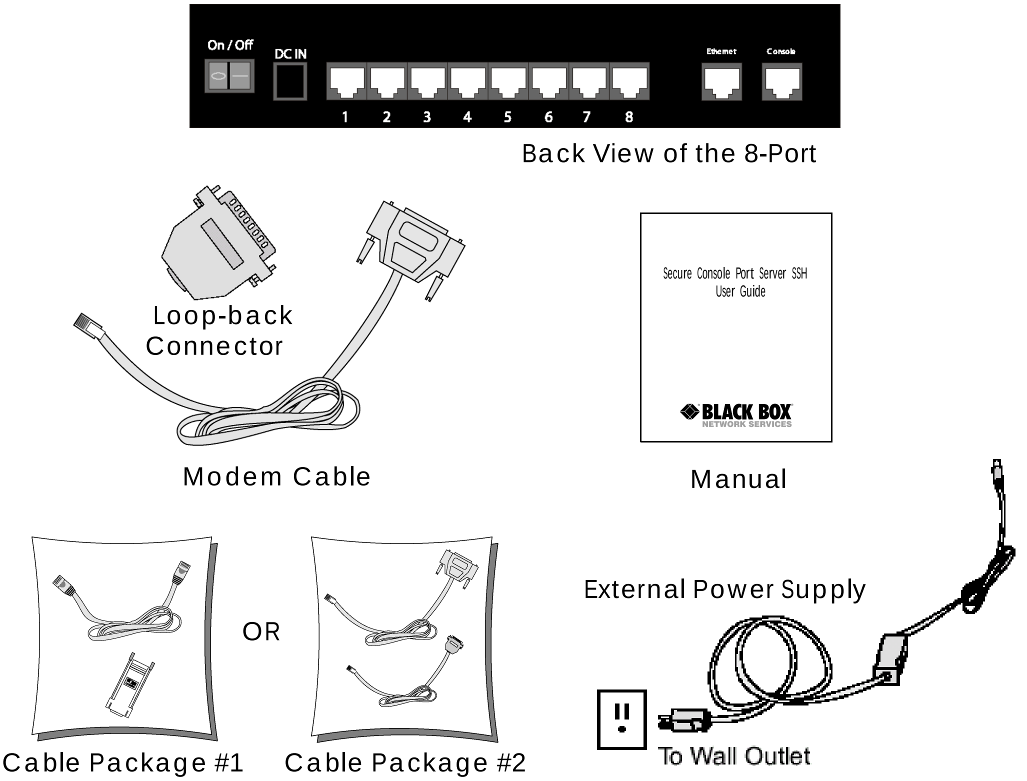


Figure 6: The Secure Console Port Server SSH 8-Port and cables

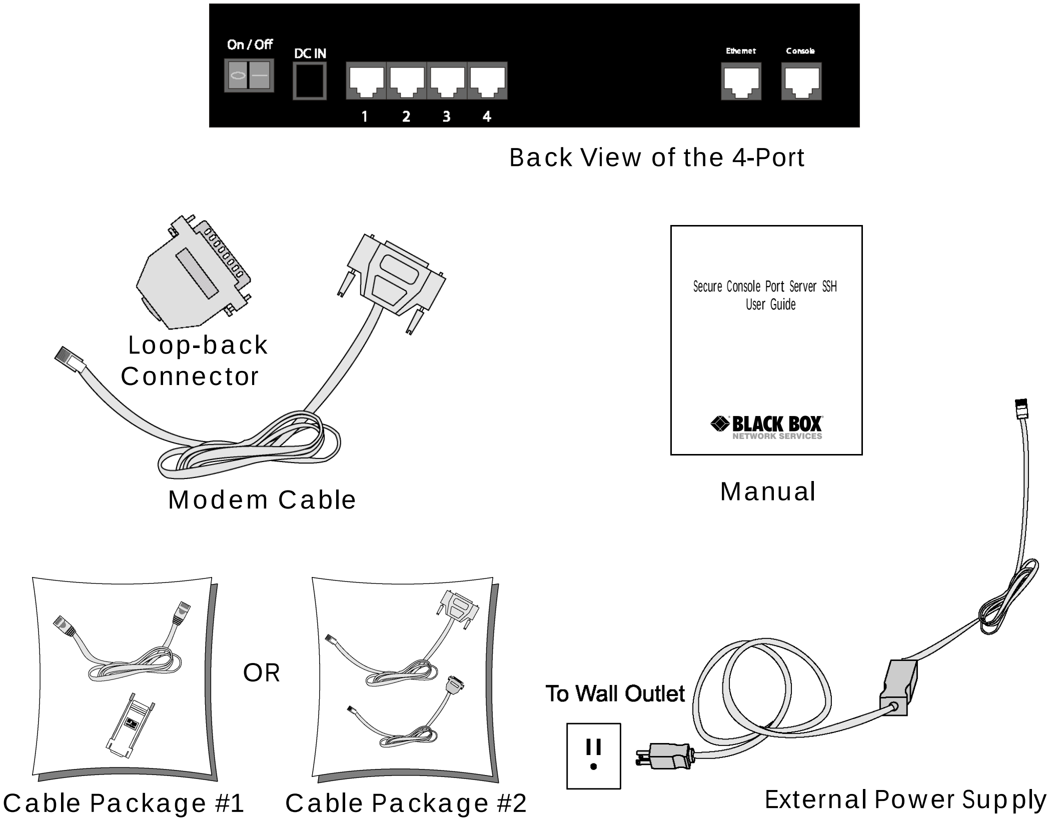


Figure 7: The Secure Console Port Server SSH 4-Port and cables

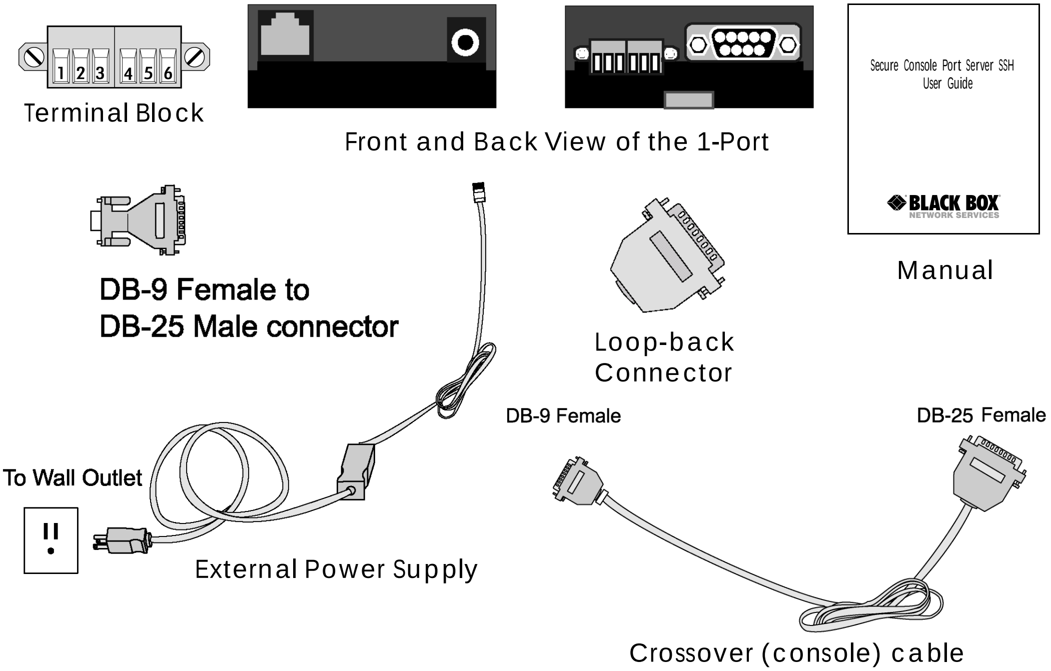


Figure 8: The Secure Console Port Server SSH 1-Port and cables

Introduction and Overview

Powering the 1-Port

There are three ways to supply power to the Secure Console Port Server SSH 1-Port:

1. External AC Desktop Power Supply: Universal AC Input (100-240VAC) / 5VDC Output. This power supply is shipped with the standard 1-Port unit (AC input)
2. External DC Supply. Three DC input options are available:
   - 12VDC nominal input (9-18 VDC)
   - 24VDC nominal input (18-36 VDC)
   - 48VDC nominal input (36-72 VDC)
3. P.O.E. (Power Over Ethernet)
   The power is supplied through the Ethernet cable. When this option is selected, the1-Port unit has to be connected to the LAN through a special hub or switch that provides DC voltage over the LAN cable. Besides these special hubs and switches, there are power injector devices available in the market which allow the users to keep using the regular hubs and switches.There are two P.O.E. standards in terms of P.O.E. feature detection circuitry. The P.O.E. supplier unit (hub, switch or power injector) can detect if the attached device supports P.O.E. One standard (old) uses capacitive load process and the second standard (new) uses resistive load process 1-Port supports both standards.

Power Supply Installation

External Desktop AC Power Supply

Step 1: Connect one end of the power cable to the 1-Port power jack (5VDC in).

Step 2: Connect the power supply end of the power cable to a standard wall outlet.

External DC Supply

Connect the two DC supply wires to the terminal block, marked as PW- and PW+. The positive voltage should be connected to PW+ and the return to PW-. If it is a -48VDC supply, the -48V signal should be connected to PW- and the return signal to PW+.

P.O.E. (Power Over Ethernet)

Notes:

- There is a label on the 1-Port unit showing the nominal DC input voltage.
- The external desktop AC Power Supply (Universal AC input / 5VDC output) is shipped with the 1-Port as a standard accessory.
- If the 5VDC input power jack is used, it will bypass the DC input from the terminal block.
- There is a protection on the terminal block's DC input. If the (PW+) and (PW-) signals are inverted, the 1-Port just won't work. It does not cause any damage to the unit.

No special setup is required. Just connect the Ethernet cable coming from the hub or switch that has support for P.O.E. or to the power injector device.

Notes:

- If the 5VDC input power jack is used, it will bypass the P.O.E. feature.
- The external desktop AC Power Supply (Universal AC input / 5VDC output) is not shipped with the 1-Port as standard accessory.

Introduction and Overview

Safety Instructions

Read all the following safety guidelines to protect yourself and your Secure Console Port Server SSH.

DANGER! In order to avoid shorting out your Secure Console Port Server SSH when disconnecting the network cable, first unplug the cable from the and then from the network jack. When reconnecting a network cable to the, first plug the cable into the network jack, and then into the.

Important! To help protect the Secure Console Port Server SSH from electrical power fluctuations, use a surge suppressor, line conditioner, or uninterruptible power supply.

Important! Be sure that nothing rests on the cables of the Secure Console Port Server SSH and that they are not located where they can be stepped on or tripped over.

Important! Do not spill food or liquids on the Secure Console Port Server SSH. If it gets wet, contact Black Box.

DANGER! Do not push any objects through the openings of the Secure Console Port Server SSH. Doing so can cause fire or electric shock by shorting out interior components.

Introduction and Overview

Important! Keep your Secure Console Port Server SSH away from heat sources and do not block cooling vents.

Important! The Secure Console Port Server SSH product (DC version) is only intended to be installed in restricted access areas (Dedicated Equipment Rooms, Equipment Closets or the like) in accordance with Articles 110-18, 110-26 and 110-27 of the National Electrical Code, ANSI/NFPA 701, 1999 Edition.

Use 18 AWG or 0.75 mm2 or above cable to connect the DC configured unit to the Centralized D.C. Power Systems.

Install the required double-pole, single-throw, DC rated UL Listed circuit breaker between the power source and the Secure Console Port Server SSH DC version. Minimum Breaker Rating: 2A. Required conductor size: 18 AWG.

Working inside the Secure Console Port Server SSH

Do not attempt to service the Secure Console Port Server SSH yourself, except when following instructions from Black Box Technical Support personnel. In the latter case, first take the following precautions:

- Turn the Secure Console Port Server SSH off.
- Ground yourself by touching an unpainted metal surface on the back of the equipment before touching anything inside it.

Replacing the Battery

A coin-cell battery maintains date and time information. The 1-Port does not have the battery, so the date and time must be kept up-to-date by ntpclient.

WARNING: There is the danger of explosion if the battery is replaced incorrectly. Replace the battery only with the same or equivalent type recommended by the manufacturer. Dispose of used batteries according to the manufacturer's instructions.

WARNUNG: Bei Einsetzen einer falschen Batterie besteht Explosionsgefahr. Ersetzen Sie die Batterie nur durch den gleichen oder vom Hersteller empfohlenen Batterietyp. Entsorgen Sie die benutzten Batterien nach den Anweisungen des Herstellers.

Предупреждение. Есть опасность взрыва, если батарея заменена неправильно. Замените батарею только тем же самым или эквивалентным типом, рекомендованным изготовителем. Избавьтесь от используемых батарей согласно инструкциям изготовителя.

Introduction and Overview

FCC Warning Statement

The Secure Console Port Server SSH has been tested and found to comply with the limits for Class A digital devices, 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 Installation & Service Manual, 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 is required to correct the problem at his or her own expense.

Canadian DOC Notice

The Secure Console Port Server SSH does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le Secure Console Port Server SSH n'émete pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la classe A prescrites dans le règlement sur le brouillage radioélectrique edicté par le Ministère des Communications du Canada.

Aviso de Precaución S-Mark Argentina

Por favor de leer todos los avisos de precaución como medida preventiva para el operador y el Secure Console Port Server SSH.

¡Peligro! No hacer funcionar el Secure Console Port Server SSH con la tapa abierta.

¡Peligro! Para prevenir un corto circuito en el Secure Console Port Server SSH al desconectarlo de la red, primero desconectar el cable del equipo y luego el cable que conecta a la red. Para conectar el equipo a la red, primero conectar el cable a la red y luego al equipo.

Introduction and Overview

¡Peligro! Asegurarse que el equipo este conectado a tierra, para prevenir un shock eléctrico. El cable eléctrico del equipo viene con tres clavijas para conectar asegurar conexión a tierra. No use adaptadores o quite la clavija de tierra. Si se tiene que utilizar una extensión, utilice una que tenga tres cables con clavija para conexión a tierra.

¡Importante! Para proteger al Secure Console Port Server SSH de fluctuaciones en corriente eléctrica, utilice una fuente eléctrica de respaldo.

¡Importante! Asegurarse de que nada descanse sobre los cables del Secure Console Port Server SSH, y que los cables no obstruyan el paso.

¡Importante! Asegurarse de no dejar caer alimentos o bebidas en el Secure Console Port Server SSH. Si esto ocurre, avise a Black Box.

¡Peligro! No empuje ningún tipo de objeto en los compartimientos del Secure Console Port Server SSH. Hacer esto podría ocasionar un incendio o causar un corto circuito dentro del equipo.

Introduction and Overview

¡Importante! Mantenga el Secure Console Port Server SSH fuera del alcancé de calentadores, y asegurarse de no tapar la ventilación del equipo.

¡Importante! El Secure Console Port Server SSH con alimentación de corriente directa (CD) solo debe ser instalado en áreas con restricción y de acuerdo a los artículos 110-18, 110-26, y 110-27 del National Electrical Code, ANSI/NFPA 701, Edición 1999.

Para conectar la corriente directa (CD) al sistema, utilice cable de 0.75 mm (18 AWG).

Instalar el interruptor corriente directa (CD) aprobado por UL entre la fuente de alimentación y el Secure Console Port Server SSH. El limite mínimo del interruptor deberá ser 2 amperes, con conductor de 0.75 mm (18 AWG).

Trabajar dentro del Secure Console Port Server SSH

No intente dar servicio al Secure Console Port Server SSH, solo que este bajo la dirección de Soporte Técnico de Black Box. Si este es el caso, tome las siguientes precauciones:

Apague el Secure Console Port Server SSH. Asegurase que este tocando tierra antes de tocar cualquier otra cosa, que puede ser al tocar la parte trasera del equipo.

Batería

¡Peligro! Una batería nueva puede explotar, si no esta instalada correctamente. Remplace la batería cuando sea necesario solo con el mismo tipo recomendado por el fabricante de la batería. Deshacerse de la batería de acuerdo a las instrucciones del fabricante de la batería.

.

This page has been left intentionally blank.

Chapter 2 - Installation, Configuration, Usage

Introduction

This chapter will allow you to install and configure the Secure Console Port Server SSH as the default CAS configuration. *Please read the entire chapter before beginning*. A basic installation and configuration should take a half hour at the most, either done manually or with the Wizard.

The Secure Console Port Server SSH operating system is embedded Linux. If you are fairly new to Linux, you will want to brush up prior to proceeding with this chapter with the essential background information presented in Appendix A - New User Background Information. *Even if you are a UNIX user and find the tools and files familiar, do not configure this product as you would a regular Linux server.*

The chapter is divided into the following sections:

- System Requirements
- Default Configuration Parameters
- Pre-Install Checklist
- Task List
- The Wizard
- Quick Start
- The Installation and Configuration Process

System Requirements

Black Box recommends either of the following specifications for configuration of the Secure Console Port Server SSH:

- A workstation with a console serial port, or
- A workstation with Ethernet and TCP/IP topology

Chapter 2 - Installation, Configuration, Usage

The following table shows the different hardware required for various configuration methods:

Table 1: Hardware vs. Configuration Methods

Hardware	Configuration Method
Console, Console Cable (constructed from RJ-45 straight-through cable + adapter)	vi, Wizard, or CLI
Workstation, Hub, Ethernet Cables	vi, Wizard, CLI, or browser

If you will be using vi, the files that need to be changed are discussed in Configuration using Telnet in this chapter. If you will be using the Wizard, basic Wizard access can be found under Configuration Wizard - Basic Wizard in Chapter 3 - Additional Features and specifics of this method are discussed under the appropriate option title in the same chapter. If you choose the browser method, the Quick Start in this chapter shows the screen flow and input values needed for this configuration mode. If you choose the CLI (Command Line Interface) method, this allows you to configure certain parameters for a specified serial port or some network-related parameters. Specifics of this method are discussed under the appropriate option title in Chapter 3 - Additional Features.

Default Configuration Parameters

- DHCP enabled (if there is no DHCP Server, IP for Ethernet is 192.168.160.10 with a Netmask of 255.255.255.0)
- CAS configuration
- socket_server in all ports (access method is telnet)
- 9600 bps, 8N1
- No Authentication

Chapter 2 - Installation, Configuration, Usage

Pre-Install Checklist

There are several things you will need to confirm prior to installing and configuring the Secure Console Port Server SSH:

Root Access — You will need Root Access on your local UNIX machine in order to use the serial port.

HyperTerminal, Kermit, or Minicom — If you are using a PC, you will need to ensure that HyperTerminal is set up on your Windows operating system. If you have a UNIX operating system, you will be using Kermit or Minicom.

IP Address of: PC or terminal, Secure Console Port Server SSH, NameServer, and Gateway — You will need to locate the IP address of your PC or workstation, the Secure Console Port Server SSH, and the machine that resolves names on your network. Your Network Administrator can supply you with these. If there is outside access to the LAN that the Secure Console Port Server SSH will be connected with, you will need the gateway IP address as well.

Network Access — You will need to have a NIC card installed in your PC to provide an Ethernet port, and have network access.

Chapter 2 - Installation, Configuration, Usage

Task List

There are eight key tasks that you will need to perform to install and configure the Secure Console Port Server SSH:

Task 1: Connect the Secure Console Port Server SSH to the Network and other Devices.

Task 2: Configure the COM Port Connection and Log In.

Task 3: Modify the System Files.

Task 4: Edit the pslave.conf file.

Task 5: Activate the changes.

Task 6: Test the configuration.

Task 7: Save the changes.

Task 8: Reboot the Secure Console Port Server SSH

The Wizard

The eight key tasks can also be done through a wizard in the 1.3.4 plus versions of the Secure Console Port Server SSH.

Basic Wizard

The Basic Wizard will configure the following parameters:

- Hostname
- DHCP enabled/disabled
- System IP (if DHCP is disabled)
- Netmask (if DHCP is disabled)
- Default Gateway
- DNS Server
- Domain

Chapter 2 - Installation, Configuration, Usage

Basic Wizard access is covered in the Quick Start in this chapter and also in Configuration Wizard - Basic Wizard in Chapter 3 - Additional Features.

Custom Wizard

Further configuration of the Secure Console Port Server SSH can be done through one of several customized wizards. These procedures are explained under their respective topic heading in Chapter 3 - Additional Features. There are custom wizards for the following optional configurations:

- Access Method
- Generating Alarms
- Authentication
- Data Buffering
- Help
- Serial Settings
- Session Sniffing
- Syslog
- Terminal Appearance

Important! If you are installing and configuring the Secure Console Port Server SSH 1-Port, there are special requirements and instructions. Be sure to read 1-Port-specific background information at the end of this chapter.

Chapter 2 - Installation, Configuration, Usage

Quick Start

This Quick Start gives you all the necessary information to quickly configure and start using the Secure Console Port Server SSH as a Console Access Server (CAS). The complete version of this process is listed later in this chapter under The Installation and Configuration Process. New Users may wish to follow the latter instruction set, as this Quick Start does not contain a lot of assumed knowledge. You can configure the Secure Console Port Server SSH by any one of four methods:

- Console
- Browser
- Telnet
- CLI (Command Line Interface)

If you have a serial port that you can use as a console port, use the Console method. If you have access to telnet, you can use this method, while New Users may prefer the Browser method for its user-friendliness.

Important! Take care when changing the IP address of the Secure Console Port Server SSH. Confirm the address you are changing it to. (You may want to write it down.)

Configuration using a Console

Step 1: Connect the console cable.

Connect the console cable (created from the RJ-45 straight-through cable and the appropriate console adapter) to the port labeled “Console” on the Secure Console Port Server SSH with the RJ-45 connector end, and to your PC’s available COM port with the serial port end.

Step 2: Power on the Secure Console Port Server SSH.

After the Secure Console Port Server SSH finishes booting, you will see a login prompt on the console screen.

Step 3: Enter *root* as login name and *tslinux* as password.

Step 4: Type *wiz* and press Enter.

A configuration wizard screen will appear in your Hyperterminal session, asking you a series of questions.

```
***********************************************************
********* CONFIGURATION  WIZARD *********
***********************************************************

INSTRUCTIONS for using the Wizard:
You can:
1) Enter the appropriate information for your system
and press ENTER or
2) Press ENTER if you are satisfied with the value
within the brackets [ ] and want to go on to the
next parameter or
3) Press ESC if you want to exit.
NOTE: For some parameters, if there is nothing within
the brackets, it will continue to ask for a value.
In that case, you must enter a valid value or # if you
do not wish to configure the value.


Press ENTER to continue...
```

You will want to configure the following settings:

- Hostname
- DHCP enabled/disabled
- System IP (if DHCP is disabled)
- Domain Name
- Primary DNS Server

- Gateway IP
- Network Mask (if DHCP is disabled)

After you input the requested parameters you will receive a confirmation screen:

```
Current configuration:

Hostname : CAS

DHCP : enabled

Domain name : mycompany.com

Primary DNS Server : 197.168.160.200

Gateway IP : 192.168.160.1
```

If the parameters are correct, “y” should be typed; otherwise, type “n” and then “c” when asked to change the parameters or quit the program. After the parameters are confirmed, the next question will be whether to save the configuration to flash. Select “y” to make the new configuration permanent in non-volatile memory.

After you confirm and save the basic parameters, you will be presented with the shell prompt. From there, either select to continue configuration using the vi editor or use the browser or CLI method (if appropriate).

The Secure Console Port Server SSH is now configured as a CAS with its new IP address, with no authentication, and accepting telnet to the serial ports. You can telnet the CAS IP + serial port 1 with the following command:

```
telnet <IP assigned by DHCP Server or by you> 7001
```

Note. Serial port 1 is configured as 9600, 8N1 by default. The server connected to this serial port has to have the same configuration for its serial port.

To explore the Secure Console Port Server SSH features, either continue configuration using the vi editor from the console or use a browser from a workstation and point to the Secure Console Port Server SSH.

Chapter 2 - Installation, Configuration, Usage

Configuration using a Web browser

The Secure Console Port Server SSH comes with DHCP client enabled. If you have a DHCP Server installed on your LAN, you can skip Step 2 below. If not, the DHCP request will fail and an IP address pre-configured on the Console server's Ethernet interface (192.168.160.10) will be used instead. To access the using your browser:

Step 1: Connect Hub to workstation and Secure Console Port Server SSH.

Your workstation and your Secure Console Port Server SSH must be on the same physical LAN. Connect one RJ-45 cable from the Ethernet port of the Secure Console Port Server SSH to a spare port from a hub, and another cable from another spare port of that same hub to the workstation used to manage the servers.

Step 2: If you do not have a DHCP Server in your LAN, add a route pointing to the Secure Console Port Server SSH IP.

From the workstation, issue a command to add a route pointing to the network IP address of the Secure Console Port Server SSH (192.168.160.0) accessed through the workstation's Ethernet interface.

For Linux, the command would be:

```
route add -net 192.168.160.0/24 gw <IP address assigned to
the workstation’s Ethernet interface>
```

Example: if the workstation has IP address 200.246.93.150 the command would be:

```
route add -net 192.168.160.0/24 gw 200.246.93.150
```

For Windows, the command would be:

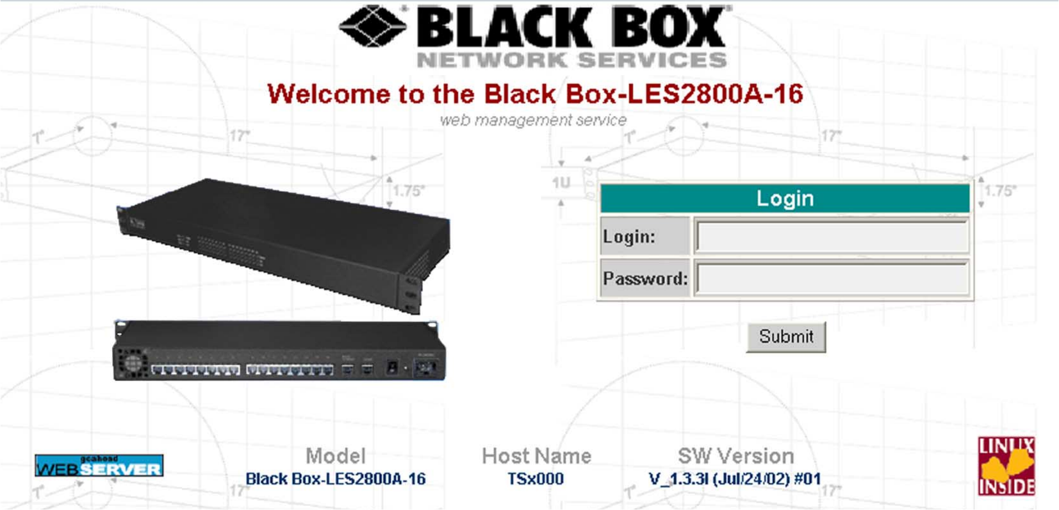
```
route add 192.168.160.0 mask 255.255.255.0 <IP address
assigned to the workstation’s Ethernet interface>
```

Example: if the workstation has IP address 200.246.93.150 the command would be:

```
route add 192.168.160.0 mask 255.255.255.0 200.246.93.150
```

Step 3: Point your browser to the IP address assigned by the DHCP Server (or to 192.168.160.10 if there is no DHCP Server in your LAN).

The login page shown in the following figure will appear.



Welcome to the Black Box-LES2800A-16

web management service

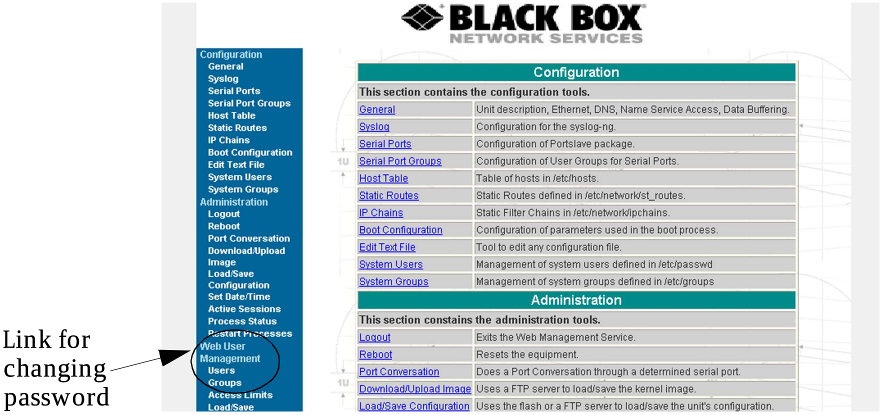
Model	Host Name	SW Version
Black Box-LES2800A-16	TSx000	V_1.3.3I (Jul/24/02) #01

Figure 9: Login page of the Web Configuration Manager

Step 4: Enter *root* **as login name and** *tslinux* **as password.**

Step 5: Click the Submit button.

This will take you to the Configuration & Administration Menu page, shown in the following figure:



Configuration	
This section contains the configuration tools.	
General	Unit description, Ethernet, DNS, Name Service Access, Data Buffering.
Syslog	Configuration for the syslog-ng.
Serial Ports	Configuration of Portslave package.
Serial Port Groups	Configuration of User Groups for Serial Ports.
Host Table	Table of hosts in /etc/hosts.
Static Routes	Static Routes defined in /etc/network/st_routes.
IP Chains	Static Filter Chains in /etc/network/ipchains.
Boot Configuration	Configuration of parameters used in the boot process.
Edit Text File	Tool to edit any configuration file.
System Users	Management of system users defined in /etc/passwd
System Groups	Management of system groups defined in /etc/groups
Administration	
This section constains the administration tools.	
Logout	Exits the Web Management Service.
Reboot	Resets the equipment.
Port Conversation	Does a Port Conversation through a determined serial port.
Download/Upload Image	Uses a FTP server to load/save the kernel image.
Load/Save Configuration	Uses the flash or a FTP server to load/save the unit's configuration.

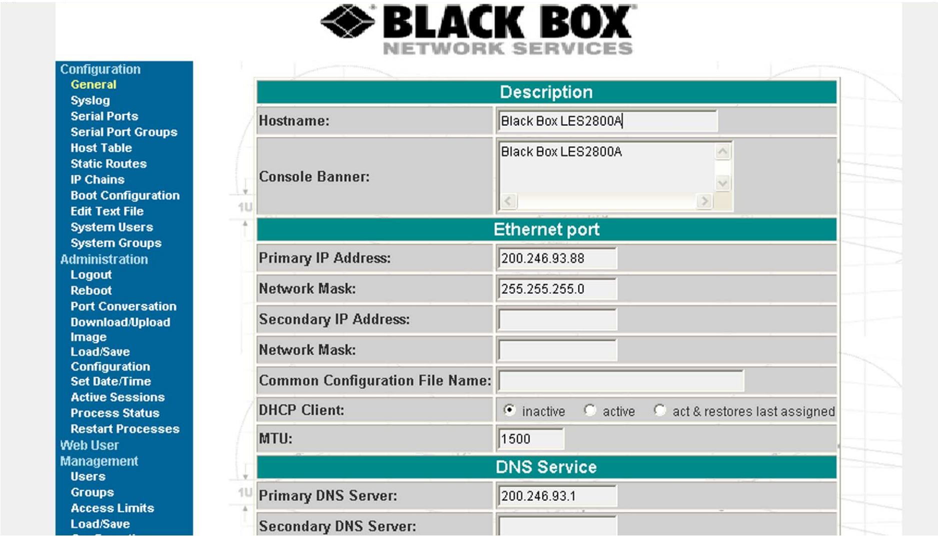
Link for changing password

Figure 10: Configuration & Administration Menu page

This page gives a brief description of all menu options. A menu of links is provided along the left side of the page. A summary of what each link leads to is shown on Table 3: Configuration Section through Table 6: Information Section.

Security Issue. Change the password of the Web root user as soon as possible. The user database for the Web Configuration Manager is different than the system user database, so the root password can be different. See Changing the Root Password in Appendix G - Web User Management.

Step 6: Click on the General link.



Description	
Hostname:	Black Box LES2800A
Console Banner:	Black Box LES2800A
Ethernet port	
Primary IP Address:	200.246.93.88
Network Mask:	255.255.255.0
Secondary IP Address:	
Network Mask:	
Common Configuration File Name:	
DHCP Client:	● inactive ○ active ○ act & restores last assigned
MTU:	1500
DNS Service	
Primary DNS Server:	200.246.93.1
Secondary DNS Server:	

Figure 11: General page

Step 7: Configure parameters presented in the fields.

Step 8: Click on the Submit button.

Step 9: Make the changes effective.

Click on the Administration > Run Configuration link, check the Serial Ports/ Ethernet/Static Routes box and click on the Activate Configuration button.
If you disabled DHCP and changed your Ethernet IP, you will lose your connection. You will need to use your browser to connect to the new IP.

Step 10: Click on the Save Configuration to Flash button.

The configuration was saved in flash. The new configuration will be valid and running. The Secure Console Port Server SSH is now configured as a CAS with its assigned (by DHCP Server or you) IP address, with no authentication, and accepting telnet to the serial ports. You can telnet the CAS IP + serial port 1 with the following command:

```
telnet <IP assigned> 7001
```

Note. Serial port 1 is configured as 9600, 8N1 by default. The server connected to this serial port has to have the same configuration for its serial port.

To explore the Secure Console Port Server SSH features, either continue configuration using browser, use the vi editor from the console, or use CLI, if appropriate. A description of each of the links on the five sections of the Configuration and Administration menu page is provided on the following five tables:

Table 2: Applications Section

Link Name	Description of Page Contents
Logout	Exits the Web Management Service
Connect to Serial Ports	Telnet/SSH connection to Portslave

Table 3: Configuration Section

Link Name	Description of Page Contents
Configuration	This section contains the configuration tools
General	Unit Description, Ethernet, DNS, Name Service Access, Data Buffering
Syslog	Configuration for the syslog-ng
SNMP	Configuration for the SNMP server
Serial Ports	Configuration of Portslave package
Serial Port Groups	Configuration of User Groups for Serial Ports
Host Table	Table of hosts in /etc/hosts
Static Routes	Static routes defined in /etc/network/st_routes
IP Chains	Shows IP Chains entries
Boot Configuration	Configuration of parameters used in the boot process
Edit Text File	Tool to edit a configuration file
System Users	Management of system users defined in /etc/password
System Groups	Management of system groups defined in /etc/groups

Table 4: Administration Section

Link Name	Description of Page Contents
Reboot	Resets the equipment
Download/ Upload Image	Uses an FTP server to load/save a kernel image
Load/Save Configuration	Uses flash memory or an FTP server to load or save theSecure Console Port Server SSH's configuration
Run Configuration	Makes the configuration changes effective
Set Date/Time	Set theSecure Console Port Server SSH 's date and time
Active Sessions	Shows the active sessions
Process Status	Shows the running processes and allows the administrator to kill them
Restart Processes	Allows the administrator to start or stop some specific processes
PCMCIA	Allows the administrator to insert and eject PCMCIA cards

Table 5: Web User Management Section

Link Name	Description of Page Contents
Users	List of users allowed to access the Web server
Groups	List of possible access groups
Access Limits	List of access limits for specific URLs
Load/Save Configuration	Load/Save Configuration in /etc/websum.conf

Table 6: Information Section

Link Name	Description of Page Contents
Interface Statistics	Shows statistics for all active interfaces
DHCP client	Shows host information from DHCP
Serial Ports	Shows the status of all serial ports
Routing Table	Shows the routing table and allows the administrator to add or delete routes
ARP Cache	Shows the ARP cache
IP Statistics	Shows IP protocol statistics
ICMP Statistics	Shows ICMP protocol statistics
TCP Statistics	Shows TCP protocol statistics
UDP Statistics	Shows UDP protocol statistics
RAM Disk Usage	Shows theSecure Console Port Server SSH File System status
System Information	Shows information about the kernel, time, CPU, and memory

Chapter 2 - Installation, Configuration, Usage

Configuration using Telnet

The Secure Console Port Server SSH comes with DHCP client enabled. If you have a DHCP Server installed on your LAN, you can skip Step 2 below. If not, the DHCP request will fail and an IP address pre-configured on the Console server's Ethernet interface (192.168.160.10) will be used instead. To access the using telnet:

Step 1: Connect Hub to workstation and Secure Console Port Server SSH.

Your workstation and your Secure Console Port Server SSH must be on the same physical LAN. Connect one RJ-45 cable from the Ethernet port of the Secure Console Port Server SSH to a spare port from a hub, and another cable from another spare port of that same hub to the workstation used to manage the servers.

Step 2: If you do not have a DHCP Server in your LAN, add a route pointing to the Secure Console Port Server SSH IP.

From the workstation issue a command to add a route pointing to the network IP address of the Secure Console Port Server SSH (192.168.160.0) accessed through the workstation's Ethernet interface.

For Linux, the command would be:

```
route add -net 192.168.160.0/24 gw <IP address assigned to
the workstation’s Ethernet interface>
```

Example: if the workstation has IP address 200.246.93.150 the command would be:

```
route add -net 192.168.160.0/24 gw 200.246.93.150
```

For Windows, the command would be:

```
route add 192.168.160.0 mask 255.255.255.0 <IP address
assigned to the workstation’s Ethernet interface>
```

Example: if the workstation has IP address 200.246.93.150 the command would be:

```
route add 192.168.160.0 mask 255.255.255.0 200.246.93.150
```

Step 3: Telnet to <IP assigned by DHCP Server or 192.168.160.10 if there is no DHCP Server>.

Step 4: Enter *root* as login name and *tslinux* as password.

Step 5: Type *wiz* and press Enter.

A Configuration Wizard screen will appear on your telnet screen, asking you a series of questions.

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

INSTRUCTIONS for using the Wizard:
You can:
1) Enter the appropriate information for your system
and press ENTER. Enter '#' if you want to
deactivate that parameter or
2) Press ENTER if you are satisfied with the value
within the brackets [ ] and want to go on to the
next parameter or
3) Press ESC if you want to exit.

NOTE: For some parameters, if there is nothing within
the brackets, it will continue to ask for a value.
In that case, you must enter a valid value or # if you
do not wish to configure the value.


Press ENTER to continue...
```

After you input the requested parameters you will receive a confirmation screen:

```
Current configuration:

Hostname : CAS

DHCP: disabled

System IP : 192.168.160.10

Domain name : mycompany.com

Primary DNS Server : 197.168.160.200
```

```
Gateway : eth0

Network Mask : 255.255.255.0
```

If the parameters are correct, “y” should be typed; otherwise, type “n” and then “c” when asked to change the parameters or quit the program. After the parameters are confirmed, the next question will be whether to save the configuration to flash. Select “y” to make the new configuration permanent in non-volatile memory.

At this point you may lose your connection when saving the changes, if you disabled DHCP and assigned an IP address. *Don’t worry!* The new configuration will be valid. The Secure Console Port Server SSH is now configured as a CAS with its assigned (by DHCP or you) IP address, with no authentication, and accepting telnet to the serial ports. You can telnet the CAS IP + serial port 1 with the following command:

```
telnet <IP assigned> 7001
```

Note. Serial port 1 is configured as 9600, 8N1 by default. The server connected to this serial port has to have the same configuration for its serial port.

After you confirm the basic parameters, you will be presented with the shell prompt. From there, either select to continue configuration using the vi editor or continue using a browser. For additional configuration, see Chapter 3 - Additional Features in this guide.

Chapter 2 - Installation, Configuration, Usage

The Installation and Configuration Process

Task 1: Connect the Secure Console Port Server SSH to the Network and other Devices

Power Users

Connect a PC or terminal to the Secure Console Port Server SSH using the console cable. If you are using a PC, HyperTerminal can be used in the Windows operating system and Kermit or Minicom in the UNIX operating system. When the Secure Console Port Server SSH boots properly, a login banner will appear. Log in as *root* (default password is *linux*). A new password should be created as soon as possible. The terminal parameters should be set as follows:

- Serial Speed: 9600 bps
- Data Length: 8 bits
- Parity: None
- Stop Bits: 1 stop bit
- Flow Control: none
- ANSI emulation

You may now skip to Task 4: Edit the pslave.conf file.

Important! Any configuration change must be saved in flash once validated. To save in Flash run saveconf (see Task 7: Save the changes). To validate/activate a configuration, run *signal_ras hup* (see Task 5: Activate the changes).

Note: If your terminal does not have ANSI emulation, select vt100; then, on the-Secure Console Port Server SSH, log in as root and switch to vt100 by typing:

```
TERM=vt100;export TERM
```

Chapter 2 - Installation, Configuration, Usage

Tip. We strongly recommend to use 9600 bps console speed. In case you need to use another speed please check Appendix E - Software Upgrades and Troubleshooting.

Important! Always complete ALL the steps for your chosen configuration before testing or switching to another configuration.

New Users

If you are using a PC, you will be using HyperTerminal to perform the initial configuration of the Secure Console Port Server SSH directly through your PC's COM port connected with the Secure Console Port Server SSH console port. HyperTerminal, which comes with Windows 95, 98, Me, NT, 2K, and XP is often located under Start > Program > Accessories. HyperTerminal emulates a dumb terminal when your PC connects to the serial port (console port) of the Secure Console Port Server SSH.

After the initial configuration through the HyperTerminal connection, you will be connecting your PC (or another terminal) to the Secure Console Port Server SSH via an Ethernet connection in order to manage the Secure Console Port Server SSH. The workstation used to access the Secure Console Port Server SSH through telnet or ssh uses a LAN connection.

These events can be summarized as follows:

- PC (Hyper terminal): COM port connects via serial cable to the Secure Console Port Server SSH's console port.
- PC (Ethernet): Ethernet port connects via hub to the Secure Console Port Server SSH's Ethernet port.
- Use the HyperTerminal to configure the box.
- Use the PC Ethernet to access the box as client (telnet/ssh).

Step 1: Plug the power cable into the Secure Console Port Server SSH.

Insert the female end of the black power cable into the power socket on the Secure Console Port Server SSH and the three-prong end into a wall outlet.

DANGER! To help prevent electric shock, plug the Secure Console Port Server SSH into a properly grounded power source. The cable is equipped with a 3-prong plug to help ensure proper grounding. Do not use adapter plugs or remove the grounding prong from the cable. If you use an extension cable, use a 3-wire cable with properly grounded plugs. For the Secure Console Port Server SSH 1-Port, 4-Port, and 8-Port, the grounded power cable constraint does not apply, as these products have an external power supply, and one power cable instead of two.

Step 2: Connect the console cable.

You will be constructing a Console Cable out of the RJ-45 straight-through cable and the appropriate adapter provided in the product box. (There are four options: all adapters have an RJ-45 connector on one end, and either a DB25 or DB9 connector on the other end, male or female). Connect this cable to the port labeled “Console” on the Secure Console Port Server SSH with the RJ-45 connector end, and connect the adapter end to your PC’s available COM port. For more detailed information on cables, see Appendix B - Cabling, Hardware, and Electrical Specifications.

Note: The modem cable is not necessary for a standard installation and configuration. Use it when the configuration is complete and you want to access the box remotely through a serial port.

Step 3: Connect Hub to PC and the Secure Console Port Server SSH.

Your workstation and Secure Console Port Server SSH must be on the same physical LAN. Connect one RJ-45 cable from the Ethernet port of the Secure Console Port Server SSH to the hub, and another from the hub to the workstation used to manage the servers.

Step 4: Install and launch HyperTerminal, Kermit or Minicom if not already installed.

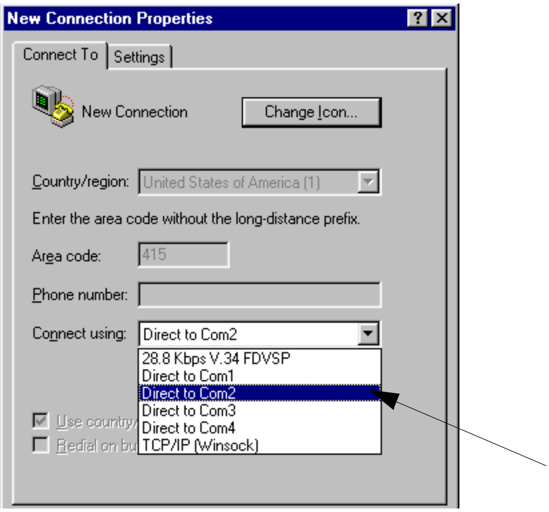
You can obtain the latest update to HyperTerminal from:

http://www.hilgraeve.com/htpe/download.html

Task 2: Configure the COM Port Connection and Log In

Step 1: Select available COM port.

In HyperTerminal (Start > Program > Accessories), select File > Properties, and click the Connect To tab. Select the available COM port number from the Connection dropdown.



New Connection Properties

Connect To | Settings

New Connection — Change Icon...

Country/region: United States of America (1)

Enter the area code without the long-distance prefix.

Area code: 415

Phone number:

Connect using: Direct to Com2

- 28.8 Kbps V.34 FDVSP
- Direct to Com1
- Direct to Com2
- Direct to Com3
- Direct to Com4
- TCP/IP (Winsock)

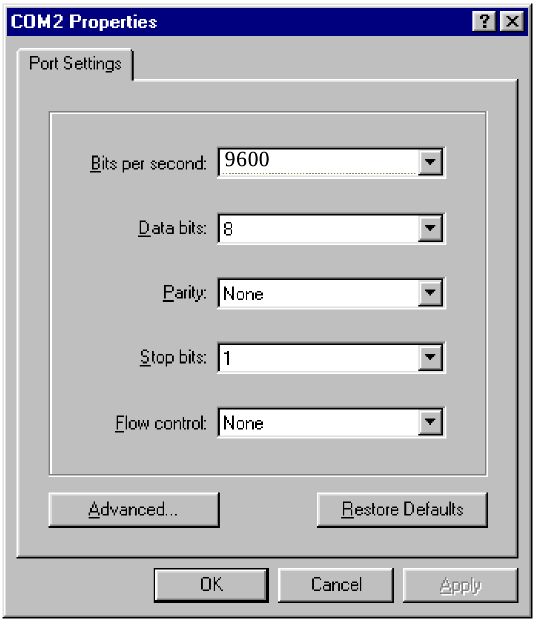
☑ Use country/

☐ Redial on bu

Figure 12: Choose a free COM port

Step 2: Configure COM port.

Click the Configure button (hidden by the dropdown menu in the above figure). Your PC, considered here to be a “dumb terminal,” should be configured to use 9600 bps, 8 data bits, no parity, 1 stop bit, and no flow control (as shown in the following figure).



COM2 Properties

Port Settings

Bits per second: 9600

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

Figure 13: Port Settings

Step 3: Power on the Secure Console Port Server SSH.

Step 4: Click OK on the Properties window.

You will see the Secure Console Port Server SSH booting on your screen. After it finishes booting, you will see a login prompt.

Task 3: Modify the System Files

When the Secure Console Port Server SSH finishes booting, a prompt will appear (a flashing underline cursor) in your HyperTerminal window. You will modify the following Linux files to let the Secure Console Port Server SSH know about its local environment:

`/etc/hostname`

`/etc/hosts`

`/etc/resolv.conf`

`/etc/network/st_routes`

`/etc/inittab` (Secure Console Port Server SSH 1-Port only)

> **Important!** If you have the Secure Console Port Server SSH 1-Port you will be modifying an additional file: /etc/inittab. See 1-Port-specific background information at the end of this chapter for instructions specific to this model.

The Linux files must be modified to identify the Secure Console Port Server SSH and other devices it will be communicating with. The operating system provides the vi editor, which is described in Appendix A - New User Background Information for the uninitiated. The Secure Console Port Server SSH runs Linux, a UNIX-like operating system, and those not familiar with it will want to refer to Appendix A.

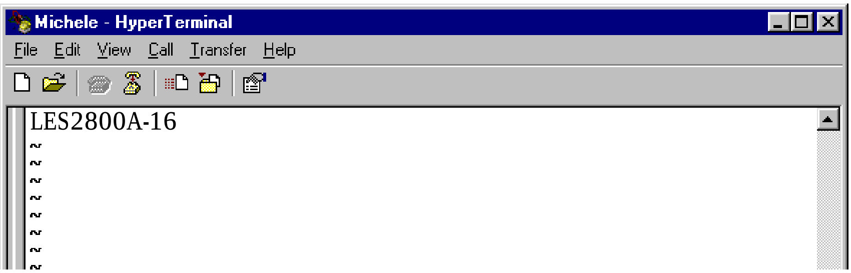
Step 1: Type *root* and press Enter.

Step 2: At the password prompt, type *tslinux*.

Press Enter.

Step 3: Modify /etc/hostname.

In HyperTerminal, type “vi /etc/hostname” (without the quotes) and press Enter. Arrow over the existing text in the file, type “r” (for replace) and type the first number of the model of your Secure Console Port Server SSH. (Or, you can replace the default naming convention with anything you’d like for your hostname.) When finished, press the Esc key, (to return to command mode), then type “:” (colon), and then “wq” and press Enter. This will save the file. (The only entry in this file should be the hostname of the Secure Console Port Server SSH.) An example is shown in the following figure. (The HyperTerminal screen is shown in this first example for clarity, however, for the other Linux files we will modify, only the command line text will be shown.)



```
LES2800A-16
~
~
~
~
~
~
~
~
```

Figure 14: The /etc/hostname file with hostname typed in

Step 4: Modify /etc/hosts.

This file should contain the IP address for the Ethernet interface and the same hostname that you entered in the /etc/hostname file. It may also contain IP addresses and host names for other hosts in the network. Modify the file using the vi as you did in Step 1.

Obtain IP address from your System Administrator

```
127.0.0.1       localhost
192.168.160.10  LES2800A-16
129.6.15.28     ntphost
```

Replace to match hostname from previous step

Figure 15: Contents of the /etc/hosts file

Step 5: Modify /etc/resolv.conf.

This file must contain the domain name and nameserver information for the network. Obtain the nameserver IP address from your Network Administrator. The default contents of this file are:

```
domain      mycompany.com

nameserver  200.200.200.2
```

Step 6: Modify /etc/network/st_routes.

The fourth file defines static routes. In the console server example in the router is a gateway router and thus its IP address is configured in this file to be the default gateway. Other static routes are also configured in this file. If you will be managing servers through a LAN, you don't need to alter this file. If you will be managing via Internet, you will be connecting through a router, and thus need to modify this file. You would get the IP address from your Network Administrator. The default contents of this file are:

```
route add default dev eth0
```

Step 7: Change password for root and new users.

The default /etc/passwd file has the user "root" with password "tslinux". You should change the password for user *root* as soon as possible. Before changing any password or adding new users you should also activate *shadow password*, if it is needed. The Secure Console Port Server SSH has support for shadow password, but it is not active by default. To activate shadow password follow the steps listed below:

Step A: Create an empty file called /etc/shadow.

```
# cd /etc
# touch shadow
```

Step B: Add a temporary user to the system. It will be removed later.

```
# adduser boo
```

Step C: Edit the file *shadow*.

For each user in passwd file, create a copy of the line that begins with "boo:" in the shadow file, then replace "boo" with the user name. The line beginning with "root" must be the first line in the file /etc/shadow.

Step D: Edit the *passwd* file.

Replace the password in all password fields with an "x". The root's line will look like this:

```
"root:x:0:0:root:/root:/bin/sh"
      ^
      ^ password field
```

Tip. Using the vi editor, put the cursor in the first byte after “root:”, then type “ct:x” plus <ESC>.

Step E: Remove the temporary user boo.

```
# deluser boo
```

Step F: Change the password for all users and add the new ones needed.

```
# passwd <username>
or
# adduser <username>
```

Step G: Edit */etc/config_files* and add a line with “*/etc/shadow.*”

Task 4: Edit the pslave.conf file

This is the main configuration file (/etc/portslave/pslave.conf) that contains most product parameters and defines the functionality of the Secure Console Port Server SSH. Only three parameters need to be modified or confirmed for a basic configuration:

- conf.eth_ip (if you disabled DHCP)
- all.authtype
- all.protocol

Tip. You can do a find for each of these parameters in vi, once you open this file by typing / <*your string*> to search the file downward for the string specified after the /.

A listing of the pslave.conf file with all possible parameters, as well as the files used to create other configurations from parameters in this file, is provided in Appendix C - The pslave Configuration File. Additional, optional modifications made to this file will depend on the configuration desired.

There are three basic types of parameters in this file:

- *conf.** parameters are global or apply to the Ethernet interface.
- *all.** parameters are used to set default parameters for all ports.
- *s#.** parameters change the default port parameters for individual ports.

An all.* parameter can be overridden by a s#.* parameter appearing later in the pslave.conf file (or vice-versa).

> **Power Users:** To find out what to input for these three parameters so that you can configure what you need, go the appropriate appendix, where you will find a complete table with an explanation for each parameter. You can use the templates from that same Appendix (pslave.conf.cas, etc.) as reference.

conf.eth_ip This is the IP address of the Ethernet interface. Use it if you don't have DHCP Server in your LAN. An example value would be:

```
200.200.200.1
```

all.authtype

This parameter controls the authentication required by the Secure Console Port Server SSH. The authentication required by the device to which the user is connecting is controlled separately. There are several authentication type options:

- *none* (no authentication)
- *local* (authentication is performed using the /etc/passwd file)
- *remote* (This is for a terminal profile only. The unit takes in a username but does not use it for authentication. Instead it passes it to the remote server where it is then used for authentication.)
- *radius* (authentication is performed using a Radius authentication server)
- *TacacsPlus* (authentication is performed using a TacacsPlus authentication server)
- *ldap* (authentication is performed against an ldap database using an ldap server. The IP address and other details of the ldap server are defined in the file /etc/ldap.conf)
- *local/radius* (authentication is performed locally first, switching to Radius if unsuccessful)
- *radius/local* (the opposite of the previous option)
- *local/TacacsPlus* (authentication is performed locally first, switching to TacacsPlus if unsuccessful)
- *TacacsPlus/local* (the opposite of the previous option)
- *RadiusDownLocal* (local authentication is tried only when the Radius server is down)
- *TacacsPlusDownLocal* (local authentication is tried only when the TacacsPlus server is down)

An example value would be:

```
radius
```

all.protocol For the console server configuration, the possible protocols are:

- *socket_server* (when telnet is used)
- *socket_ssh* (when ssh version one or two is used)
- *raw_data* (to exchange data in transparent mode – similar to socket_server mode, but without telnet negotiation, breaks to serial ports, etc.)

An example value would be:

```
socket_server
```

The Authentication feature

See Authentication in Chapter 3 - Additional Features.

Task 5: Activate the changes

Execute the following command in HyperTerminal to activate the changes:

```
signal_ras hup
```

Task 6: Test the configuration

Now you will want to make sure that the ports have been set up properly.

Step 1: Ping the Secure Console Port Server SSH from a DOS prompt.

Open a DOS window, type in the following, and then press Enter:

```
ping <IP assigned to the Secure Console Port Server SSH by
DHCP or you>
```

An example would be:

```
ping 192.168.160.10
```

If you receive a reply, your Secure Console Port Server SSH connection is OK. If there is no reply see Appendix E - Software Upgrades and Troubleshooting.

Step 2: Telnet to the server connected to the first port of the Secure Console Port Server SSH.

(This will only work if you selected socket_server as your all.protocol parameter.)

While still in the DOS window, type the following and then press Enter:

```
telnet <IP assigned to the Secure Console Port Server SSH by
DHCP or you> 7001
```

An example would be:

```
telnet 192.168.160.10 7001
```

If everything is configured correctly, a telnet session should open on the server connected to port 1. If not, check the configuration, follow the above steps again, and check Appendix E - Software Upgrades and Troubleshooting.

Task 7: Save the changes

Execute the following command in HyperTerminal to save the configuration:

```
saveconf
```

Task 8: Reboot the Secure Console Port Server SSH

After rebooting, the initial configuration is complete.

Note: restoreconf does the opposite of saveconf, copying the contents of the /proc/flash/script file to the corresponding files in the ramdisk. The files on the ramdisk are overwritten. Restoreconf is run automatically each time the Secure Console Port Server SSH is booted.

Special Configuration for the Secure Console Port Server SSH 1-Port

1-Port-specific background information

Since there are two configurable physical interfaces available in the 1-Port (RS-232 and RS-485), these models require the configuration of the parameter described below.

Configuring the 1-Port for the first time

DANGER! When reconfiguring the media from RS232 to RS485 (or RS485 to RS232), it is extremely important to remove the serial cable (DB9 connector) before issuing signal_ras hup or signal_ras start (to make valid the new configuration). Using wrong cable for that newly configured media may burn the serial interface.

When using Web or telnet/ssh session to reconfigure the media, follow these steps:
1) Remove the serial cable before the reconfiguration
2) Start the reconfiguration process
3) Save and submit the changes (e.g. signal_ras hup)
4) Insert the new serial cable compatible with the new media

When using console for the configuration, follow these steps:
1) Edit the proper configuration file
2) Quit editor, saving the changes
3) Run saveconf
4) Remove the serial cable
5) Power the unit off
6) Insert the new serial cable compatible with the new media
7) Power the unit back on

The 1-Port does not have a dedicated console port. Therefore, after configuring the serial port, perform the following steps:

Step 1: Edit the file /etc/inittab.

Comment the line that designates the console port (add a “#” to it):

```
# ttyS0::respawn:/sbin/getty -p ttyS0 ansi
```

Uncommenting the line starts the program cy_buffering (remove the '#' from the beginning):

```
::once:/sbin/cy_buffering
```

Step 2: Run saveconf.

The command saveconf, which reads the /etc/config_files file, should be run. The command saveconf copies all the files listed in the file /etc/config_files from the ramdisk to /proc/flash/script. The previous contents of the file /proc/flash/script will be lost.

Step 3: Reboot.

After rebooting the 1-Port, the initial configuration is complete.

Chapter 2 - Installation, Configuration, Usage

Accessing the Serial Ports

There are four ways to access the serial ports, depending on the protocol you configured for that serial port (all.protocol being socket_server for telnet access, socket_ssh for ssh access, etc). For details on configuration to access using telnet or ssh please see Access Method, Configuration for CAS in Chapter 3.

Opening and closing a telnet session to a serial port

To open a telnet session to a serial port, issue the command:

```
telnet <CAS hostname> <TCP port number>
```

<CAS hostname> is the hostname configured in the workstation where the telnet client will run (through /etc/hosts or DNS table). It can also be just the IP address of the Secure Console Port Server SSH (Ethernet's interface) configured by the user or learned from DHCP.

Note: saveconf is equivalent to tar -czf /proc/flash/script -T /etc/config_files in standard Linux (saveconf must be used because tar on the Secure Console Port Server SSH does not support the z flag).
Note: restoreconf does the opposite of saveconf, copying the contents of the /proc/flash/script file to the corresponding files in the ramdisk. The files on the ramdisk are overwritten. Restoreconf is run automatically each time the Secure Console Port Server SSH is booted.
<TCP port number> is the number associated to the serial port. From factory, 7001 corresponds to serial port 1, 7002 to serial port 2 and so forth.

To close the telnet session, just press the telnet hot key configured in telnet client application (usually it's "Ctrl]") and "q" to quit.

Opening and closing an SSH session to a serial port

To open a ssh session to a serial port, issue the command:

```
ssh -l <Username>:<Server> <CAS hostname>
```

<Username> is the user configured to access that serial port. It is present either in the local CAS database or in a Radius/Tacacs/LDAP/Kerberos, etc database.

<Server> can be just the TCP port number assigned for that serial port (7001, 7002, etc) or the alias for the server connected to that serial port.

<CAS hostname> is the hostname configured in the workstation where the ssh client will run (through /etc/hosts or DNS table). It can also be just the IP address of the Secure Console Port Server SSH (Ethernet's interface) configured by the user or learned from DHCP.

To exit the ssh session, press the hot key configured for that ssh client (usually "~.").

Accessing Serial Ports using “ts_menu”

To access the serial port (telnet or ssh) using *ts_menu*, login to the CAS unit and, after receiving the shell prompt, run *ts_menu*. The servers (aliases) or serial ports will be shown as option to start a connection (telnet/ssh). After typing ts_menu, you will see something similar to the following:

```
Serial Console Server Connection Menu for your Master Terminal
Server

1 ttyS1 2 ttyS2 3 ttyS3 4 ttyS4
5 ttyS5 6 ttyS6 7 ttyS7 8 ttyS8

Type 'q' to quit, a valid option[1-8], or anything else to refresh:
```

How to close the session from ts_menu (from the console of your unit)

Step 1: Enter the escape character.

The escape character is shown when you first connect to the port.
In character/text Mode, the Escape character is ^]

After entering the escape character, the following is shown:

```
Console escape. Commands are:

l go to line mode
c go to character mode
z suspend telnet
b send break
t toggle binary
e exit telnet
```

Step 2: Press “e” to exit from the session and return to the original menu.

Select the exit option and you will return to the shell prompt.

How to close the session from ts_menu (from a telnet session to your unit)

You have to be sure that a different escape character is used for exiting your telnet session; otherwise, if you were to exit from the session created through the ts_menu, you will close your entire telnet session to your unit. To do this, when you first telnet to your unit, use the “-e” option. So for example, to set Ctrl-? as the escape character, type:

```
telnet -e ^? 192.168.160.10
```

To exit from the session created through the ts_menu, just follow Step 1 from above. To exit from the entire telnet session to your unit, type the escape character you had set.

Accessing Serial Ports using the Web Interface

From the Web, there's a “Connect to Serial Port” option that has to be selected. A serial port is chosen and a Java window will open on the user's screen. For a telnet session, just log in and provide the password (whenever necessary). For ssh, enter

```
<username>:<TCP port number or alias for the server>
```

as login name and provide the password (whenever necessary). To exit the session, select “Disconnect” from the Java window. See the Step-by-Step Process section of Appendix H - Connect to Serial Ports from Web for more details.

Chapter 3 - Additional Features

Introduction

After the Configuration Wizard section in this chapter, each of the following sections is listed alphabetically and shows how to configure the option using vi, the custom Wizard (when available), browser, where appropriate, and the Command Line Interface (CLI), when available. This chapter contains the following sections:

- Configuration Wizard - Basic Wizard
- Access Method
- Authentication
- CAS Port Pool
- Clustering
- CronD
- Data Buffering
- DHCP
- Filters
- Generating Alarms
- Help
- Modbus
- NTP
- Ports Configured as Terminal ServersSerial Settings
- Session Sniffing
- SNMP
- Syslog
- Terminal Appearance
- Time Zone

Configuration Wizard - Basic Wizard

The configuration wizard application is a quicker and easier way to configure the Secure Console Port Server SSH. It is recommended that you use this application if you are not familiar with the vi editor or if you just want to do a quick installation of the Secure Console Port Server SSH.

The command *wiz* gets you started with some basic configuration. After executing this command, you can continue the configuration of the Secure Console Port Server SSH using any browser or by editing system files with the vi editor. What follows are the basic parameters to get you quickly started. The files that will be eventually modified if you decide to save to flash at the end of this application are:

1. /etc/hostname
2. /etc/hosts
3. /etc/resolv.conf
4. /etc/network/st_routes
5. /etc/network/ifcfg_eth0
6. /etc/portslave/pslave.conf

Step 1: Enter the command *wiz*.

At the command prompt type "wiz" in your terminal to bring up the wizard. You will receive an initial instruction screen.

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

INSTRUCTIONS for using the Wizard:
You can:
1) Enter the appropriate information for your system
and press ENTER or
2) Press ENTER if you are satisfied with the value
within the brackets [ ] and want to go on to the
next parameter or
```

```
3) Press ESC if you want to exit.
NOTE: For some parameters, if there is nothing within
the brackets, it will continue to ask for a value.
In that case, you must enter a valid value or # if you
do not wish to configure the value.


Press ENTER to continue...
```

Step 2: Press Enter to continue with the wizard.

You will see the current configurations and have the choice of setting them to default values, or not.

```
************************************************************
********* C O N F I G U R A T I O N   W I Z A R D **********
************************************************************

Current configuration:

Hostname: CAS
DHCP: enabled
Domain name: #
Primary DNS Server: #
Gateway IP: eth0


Set to defaults? (y/n) [n] :
```

Step 3: Press Enter or type *n* or *y*.

The default answer or value to any question is in the brackets. You can take one of three actions:

- Either just press the ENTER key to execute whatever is in between the brackets, or
- Type *n* to NOT reset the current configurations to the Black Box defaults, or
- Type *y* to reset to Black Box default configurations.

Tip. On most of the following configuration screens, the default or current value of the parameter is displayed inside brackets. Just press the ENTER key if you are satisfied with the value in the brackets. If not, enter the appropriate parameter and press ENTER.

If at any time after choosing whether to set your configurations to default or not, you want to exit the wizard or skip the rest of the configurations, press ESC. This will immediately display a summary of the current configurations for your verification before exiting the application. This will not work if you did not enter a valid choice for the parameter you are currently on.

For some parameters, if there is nothing within the brackets, it will continue to ask for a value. In that case, you must enter a valid value or # if you do not wish to configure the value.

Step 4: Enter Hostname and then press the Enter key.

This is an alias for yourSecure Console Port Server SSH that allows you to refer to the Secure Console Port Server SSH by this name rather than its IP address. Enter hostname after the prompt:

```
Hostname[CAS]:
```

Step 5: Type *y*, *n*, or press Enter to enable or disable DHCP client.

Type *y* or press Enter if there is a DHCP Server in your LAN, to have the Dynamic Host Configuration Protocol (DHCP) automatically assign an IP address for your Secure Console Port Server SSH. Type *n* to manually assign an IP address.

```
Do you want to use dhcp to automatically assign an IP for
your system (y/n) [y]:
```

Note: Typing *y* omits Steps 6 and Step 10.

Step 6: If DHCP client is disabled, enter IP Address of yourSecure Console Port Server SSH and then press the Enter key.

If the DHCP client is enabled, skip this step. This question will only appear if DHCP client is disabled. This is the IP address of theSecure Console Port Server SSH within your network. See your network administrator to obtain a valid IP address for theSecure Console Port Server SSH .

```
IP of your system[]: 192.168.160.10
```

Step 7: Enter Domain name and then press Enter.

Domain name locates or identifies your organization within the Internet.

```
Domain name[#]: mycompany.com
```

Step 8: Enter IP address of Domain Name Server and press Enter.

At the prompt, enter the IP address of the server that resolves domain names. Your domain name is alphabetical so that it is easier to remember. Every time you see the domain name, it is actually being translated into an IP address by the domain name server. See your network administrator to obtain this IP address for the domain name server.

```
Domain Name Server[#]: 192.168.160.200
```

Step 9: Enter Gateway IP address and press Enter.

The Gateway is a node on a network that serves as an entrance point into another network. See your network administrator to find out your organization's gateway address.

```
Gateway IP[eth0]: 192.168.160.1
```

Step 10: If DHCP client is disabled, enter Netmask and press Enter.

If the DHCP client is enabled, skip this step. This question will appear only if DHCP client is disabled. The Netmask is a string of 0s and 1s that mask or screen out the host part of an IP address so that only the network part of the address remains.

```
Netmask[#]: 255.255.255.0
```

Step 11: Review configuration parameters.

You will now have the parameters you just configured displayed back to you. If you entered *y* in Step 5:

```
************************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
************************************************************

Current configuration:

Hostname: CAS
DHCP: enabled
Domain name: mycompany.com
Primary DNS Server: 197.168.160.200
Gateway IP: 192.168.160.1

Are all these parameters correct (Y)es or (N)o [N]:
```

If you entered *n* in Step 5:

```
Current configuration:

Hostname: CAS
DHCP: disabled
System IP: 192.168.160.10
Domain name: mycompany.com
Primary DNS Server: 192.168.160.200
Gateway IP: 192.168.160.1
Network Mask: 255.255.255.0

Are all these parameters correct (y/n) [y]:
```

Step 12: Type *y*, or *n*, or press Enter.

Type *y* if all parameters are correct. Type *n* or just press ENTER if not all the parameters are correct and you want to go back and redo them.

Step 13: If you typed *n* in Step 11, type *c* or *q*.

As directed by the prompt, type *c* to go back to very beginning of this application to change the parameters. Type *q* to exit.

Step 14: If you typed *y* in Step 11, choose whether to activate your configurations.

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

You can now use the browser to finish your system configu-
rations, but before that, please read below.


(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)


Do you want to activate your configurations now? (y/n) [y] :
```

Step 15: Choose whether to save to flash.

Flash is a type of memory that will maintain the information saved on it even after the Secure Console Port Server SSH is turned off. Once it is turned on again, the saved information can be recovered. If *y* is entered, the screen will display an explanation of what saving to flash means:

```
Flash refers to a type of memory that can be erased and
reprogrammed in units of memory known as blocks rather than
one byte at a time, thus making updating to memory easier.

If you choose to save to flash, your configurations thus far
will still be in the memory of the Console Port Server SSH
even after you reboot it. If you don't save to flash and if
you were to reboot the system, all your new configurations
will be lost and you will have to reconfigure the Secure
Console Port Server SSH.

Do you want to save your configurations to flash? (y/n) [n]:
```

Step 16: Type 'y' if you want to save to flash. Type 'n' if you don't want to save to flash.

You can now continue Secure Console Port Server SSH configurations using the Web browser by typing in the IP address of the Secure Console Port Server SSH.

Using the Wizard through your Browser

The Web interface supports wizards for serial ports configuration. The wizard is a useful tool that simplifies configuration of serial ports. The Web interface will access the following wizard files:

- /etc/portslave/pslave.wiz.cas (CAS)
- /etc/portslave/pslave.wiz.ts (TS)
- /etc/portslave/pslave.wiz.ras (Dial-in Access)
- /etc/portslave/pslave.wiz.auto (Automation)

The step-by-step process to configuring ports for a specific profile appear in the following sections, and the exact screen flow begins with.

To summarize the process, the wizard configuration is started by first selecting the desired port(s) on the Port Selection page (Figure 17: Port Selection page), clicking Submit, and then selecting either the CAS, TS, or RAS profile buttons on the subsequent Serial Port Configuration Page. Change the appropriate parameters, and then click the Submit button on the Serial Port Configuration Page. For most applications, the parameters to be changed are:

For CAS:

- Port Speed
- First RADIUS/TacacsPlus Authentication Server
- First Accounting Server
- RADIUS/TacacsPlus secret
- Protocol (if the protocol is Socket SSH, Socket Telnet, or Socket Raw)
- Socket Port (keep the “Incremented” option on)

For TS:

- Port Speed
- First RADIUS/TacacsPlus Authentication Server
- First Accounting Server
- RADIUS/TacacsPlus secret
- Protocol (if the protocol is Login, Rlogin, SSH, or Socket Client)
- Socket Port (write the TCP port for the protocol selected; keep the “incremented” option off)

For Dial-in access:

- First RADIUS/TacacsPlus Authentication Server
- First Accounting Server
- RADIUS/TacacsPlus secret
- Remote IP Address (keep the “Incremented” option on)

Access Method

Access method is how a user accesses a server connected to one of the serial ports on the Secure Console Port Server SSH (CAS profile) or how a user connected to one of the serial ports accesses a server in the network (TS profile or Dial-In profile).

Configuration for CAS

Parameters Involved and Passed Values

The parameters involved in configuring Access Method for CAS are as follows:

all.ipno

This is the default IP address of the Secure Console Port Server SSH's serial ports. Any host can access a port using its IP address as long as a path to the address exists in the host's routing table. An example value would be 192.168.1.101+. The “+” indicates that the first port should be addressed as 192.168.1.101 and the following ports should have consecutive values.

all.socket_port

In the CAS profile, this defines an alternative labeling system for the Secure Console Port Server SSH ports. An example value would be 7001+. The “+” after the numerical value causes the serial interfaces to be numbered consecutively. In this example, serial interface 1 is assigned the port value 7001, serial interface 2 is assigned the port value 7002, etc. One example on how this could be used is in the case of all.protocol or s<n>.protocol socket_ssh and the port value (7001, 7002, etc), if supplied by the ssh client like username:port value, the ssh client will be directly connected with the serial interface.

all.protocol

The possible protocols are telnet, ssh1/ssh2 or raw data:
socket_server = telnet protocol,
socket_ssh = ssh1/ssh2 protocol,
raw_data = used to exchange data in transparent mode. Raw_data is similar to socket_server mode but without telnet negotiation breaks to serial ports.
An example value would be socket_server.

all.users

Restricts access to ports by user name (only the users listed can access the port or, using the character “!,” all but the users listed can access the port.) A single comma and spaces/tabs may be used between names. A comma may not appear between the “!” and the first user name. The users may be local, Radius or TacacsPlus. User groups (defined with the parameter conf.group) can be used in combination with user names in the parameter list. Notice that these are common users, not administrators. Example: all.users ! joe, mark, user_group. In this example, the users joe, mark, and members of user_group cannot access the port.

all.poll_interval Valid only for protocols socket_server and raw_data. When not set to zero, this parameter sets the wait for a TCP connection keep-alive timer. If no traffic passes through the Secure Console Port Server SSH for this period of time, the Secure Console Port Server SSH will send a line status message to the remote device to see if the connection is still up. If not configured, 1000 ms is assumed (the unit for this parameter is ms). If set to zero, line status messages will not be sent to the socket client.

all.tx_interval Valid for protocols socket_server and raw_data. Defines the delay (in milliseconds) before transmission to the Ethernet of data received through a serial port. If not configured, 100ms is assumed. If set to zero or a value above 1000, no buffering will take place.

all.idletimeout *Valid only for the CAS configuration* (protocols socket_server, socket_ssh, raw_data) and modbus. Specifies how long (in minutes) a connection can remain inactive before it is cut off. If set to zero (the default), the connection will not time out.

conf.group Used to group users to simplify configuration of the parameter all.users later on. This parameter can be used to define more than one group. The format is:
<group name>:<user1>{,<user2>[,<user3>]]
Example: conf.group group_name: user1, user2.

s<n>.serverfarm Alias name given to the server connected to the serial port. Server_connected.
Example: s1.serverfarm Server_connected_serial1.

vi Method

The parameters described above must be changed by directly editing the /etc/portslave/plsave.conf file.

Browser Method

To configure Access Method with your browser:

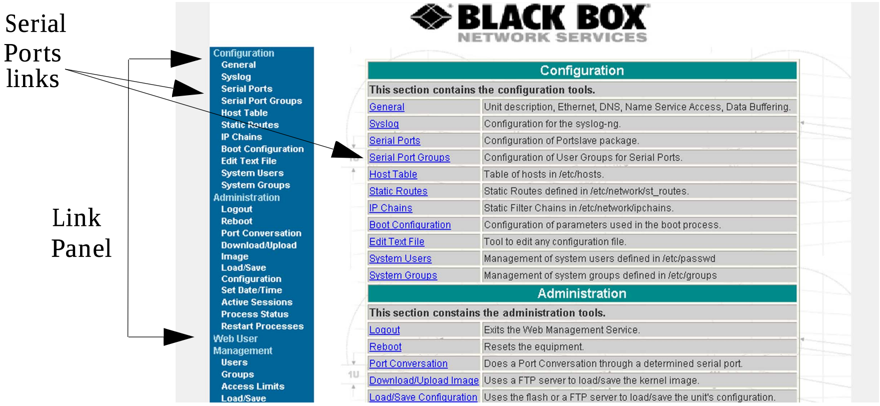
Step 1: Point your browser to the Console Server.

In the address or location field of your browser type the Console Access Server’s IP address. For example:

```
http://10.0.0.0
```

Step 2: Log in as root and type the Web root password configured by the Web server.

This will take you to the Configuration and Administration page.



Configuration

This section contains the configuration tools.	
General	Unit description, Ethernet, DNS, Name Service Access, Data Buffering.
Syslog	Configuration for the syslog-ng.
Serial Ports	Configuration of Portslave package.
Serial Port Groups	Configuration of User Groups for Serial Ports.
Host Table	Table of hosts in /etc/hosts.
Static Routes	Static Routes defined in /etc/network/st_routes.
IP Chains	Static Filter Chains in /etc/network/ipchains.
Boot Configuration	Configuration of parameters used in the boot process.
Edit Text File	Tool to edit any configuration file.
System Users	Management of system users defined in /etc/passwd
System Groups	Management of system groups defined in /etc/groups

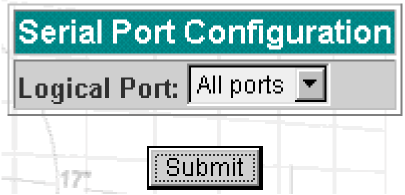
Administration

This section constains the administration tools.	
Logout	Exits the Web Management Service.
Reboot	Resets the equipment.
Port Conversation	Does a Port Conversation through a determined serial port.
Download/Upload Image	Uses a FTP server to load/save the kernel image.
Load/Save Configuration	Uses the flash or a FTP server to load/save the unit's configuration.

Figure 16: Configuration and Administration page

Step 3: Select the Serial Ports link.

Click on the Serial Ports link on the Link Panel to the left of the page or in the Configuration section of the page. This will take you to the Port Selection page.

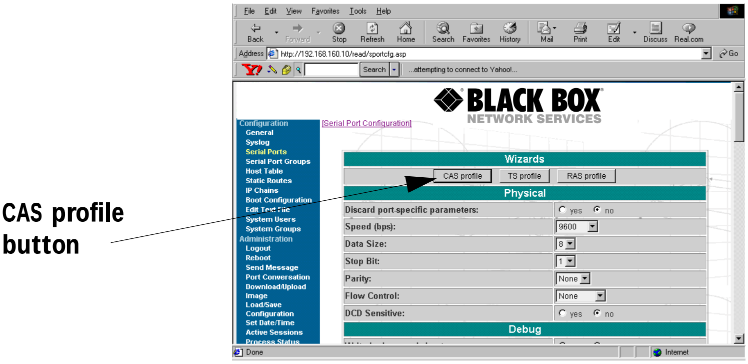


Logical Port: All ports

Figure 17: Port Selection page

Step 4: Select port(s).

On the Port Selection page, choose all ports or an individual port from the dropdown menu. This will take you to the Serial Port Configuration page.



http://192.168.160.10/read/sportcfg.asp

...attempting to connect to Yahoo!...

[Serial Port Configuration]

Wizards

CAS profile | TS profile | RAS profile

Physical

Discard port-specific parameters:	○ yes ● no
Speed (bps):	9600
Data Size:	8
Stop Bit:	1
Parity:	None
Flow Control:	None
DCD Sensitive:	○ yes ● no

Debug

CAS profile button

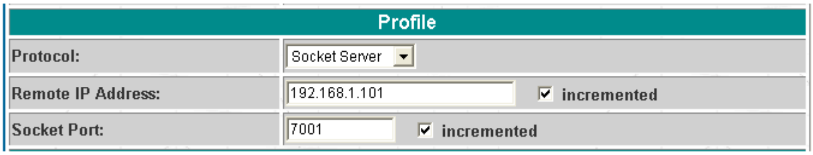
Figure 18: Serial Port Configuration page

Step 5: Click the CAS profile button.

Click the CAS profile button in the wizards section. The default CAS profile parameters are now loaded.

Step 6: Scroll down to the Profile section.

You can change the settings for *all.ipno*, *all.socket_port*, and *all.protocol* in this section.



Profile

Protocol:	Socket Server	
Remote IP Address:	192.168.1.101	☑ incremented
Socket Port:	7001	☑ incremented

Figure 19: Profile Section of Serial Port Configuration page

Step 7: Scroll to the Authentication Section.

You can configure the parameter *all.users* here under Access Restriction on Users.

Step 8: Scroll to Console Access Server Section.

You can configure the following parameters here:

- all.sttyCmd
- all.poll_interval
- all.tx_interval
- all.idletimeout

Step 9: Configure s<n>.serverfarm.

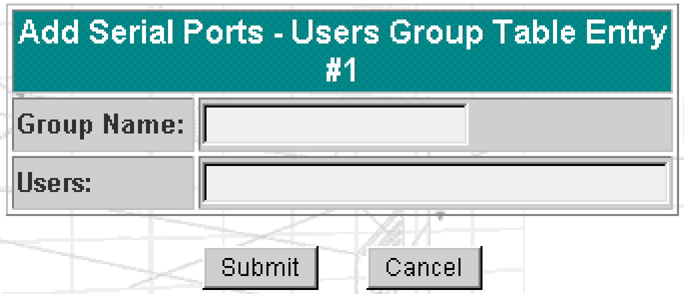
This parameter will not appear on the configuration page when “All ports” is selected. Scroll to the SSH section. Each port can be named after the server or device connected to it. This makes the process of associating what is connecting to which port easier.

Step 10: Click the Submit button.

This will take you back to the Port Selection page. At this point, the configuration file is written in the RAMdisk.

Step 11: Click on the Serial Port Groups link on the Link Panel.

Click the Add Group button that appears. A Serial Ports - Users Group Table Entry page appears.



Add Serial Ports - Users Group Table Entry #1

Group Name:

Users:

Figure 20: Serial Ports - Users Group Table Entry page

Step 12: Configure conf.group.

Fill in the Group Name and Users fields to configure the group.

Step 13: Click the Submit button.

At this point, the configuration file is written in the RAMdisk.

Step 14: Make the changes effective.

Click on the Administration > Run Configuration link, check the Serial Ports/ Ethernet/Static Routes box and click on the Activate Configuration button.

Step 15: Save it in the flash.

Go to the link Administration > Load/Save Configuration and click the Save to Flash button.

Wizard Method

Step 1: Bring up the wizard.

At the command prompt, type the following to bring up the Access Method custom wizard:

```
wiz --ac cas
```

This will bring up Screen 1:

Screen 1:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

INSTRUCTIONS for using the Wizard:
You can:
1) Enter the appropriate information for your system
and press ENTER. Enter '#' if you want to
deactivate that parameter or
2) Press ENTER if you are satisfied with the value
within the brackets [ ] and want to go on to the
next parameter or
3) Press ESC if you want to exit.

NOTE: For some parameters, if there is nothing within
```

```
the brackets, it will continue to ask for a value.
In that case, you must enter a valid value or # if you
do not wish to configure the value.


Press ENTER to continue...
```

Screen 2:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************

Current configuration:
(The ones with the '#' means it's not activated.)

all.ipno : #
all.socket_port : 7001+
all.protocol : socket_server
all.users : #
all.poll_interval : #
all.tx_interval : #
all.idletimeout : #
conf.group : #


Set to defaults? (y/n) [n] :
```

Screen 3:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

ALL.IPNO - This is the default IP address of the system's
serial ports. If configured as 192.168.1.101+, the '+'
indicates that the first port should be addressed as
192.168.1.101 and the following ports should have
consecutive values. Any host can access a port using
its IP address as long as a path to the address exists
```

```
in the host's routing table.

all.ipno[#] :


ALL.SOCKET_PORT - This defines an alternative labeling
system for the system ports. The '+' after the numerical
value causes the interfaces (or ports) to be numbered
consecutively.
(e.g. interface 1 of your system is assigned port 7001,
interface 2 has the value 7002, etc.)

all.socket_port[7001+] :
```

Screen 4:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

ALL.PROTOCOL - The possible protocols are telnet,
ssh1/ssh2, raw data, or modbus.
(e.g. socket_server -telnet protocol, socket_ssh -ssh1/ssh2
protocol, raw_data -used to exchange data in transparent
mode; similar to socket_server mode but without telnet
negotiation breaks to serial ports modbus -an application
layer messaging protocol for client/server communication
widely used for industrial automation, etc.)

all.protocol[socket_server] :

ALL.MODBUS_SMODE - Communication mode through the serial
ports. This parameter is valid only if the protocol
configured is modbus. If it is and this parameter is
```

Note: The modbus option only applies if you are using a 1-Port. The modbus_smode parameter will only be requested if you are configuring a 1-Port.

```
not configured, ASCII mode will be assumed.
(e.g. ascii -normal TX/RX mode, rtu -Remote Transmission
mode where some time constraints are observed between
characters while transmitting a frame)

all.modbus_smode[#] :


ALL.USERS - Restricts access to ports by user name. Only
the users listed can access the port, or using a '!',
all but the users listed can access the port.
A single comma and spaces/tabs may be used between names.
A comma may NOT appear between the '!' and the first user
name. The users may be local, Radius or TacacsPlus. User
groups (defined with the parameter conf.group) can be
used in combination with user names in the parameter list.
Notice that these are common users, not administrators.
(e.g. !joe, mark, grp1 -the users, Joe, Mark, and members
of grp1, cannot access the port.)

all.users[#] :
```

Screen 5:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

ALL.POLL_INTERVAL - Valid for protocols socket_server and
raw_data. When not set to 0, this parameter sets the wait
for a TCP connection keep-alive timer. If no traffic passes
through the system for this period of time, the system will
send a line status message to the remote device to see if
the connection is still up. If not configured, default is
1000ms. If set to 0, line status messages will not be sent
to the socket client.

all.poll_interval[#] :


ALL.TX_INTERVAL - Valid for protocols socket_server and
raw_data. This parameter defines the delay (in milli-
seconds) before transmission to the Ethernet of data
received through a serial port. If not configured, 100ms
is assumed. If set to 0 or a value above 1000, no buffering
will take place.

all.tx_interval[#] :
```

Screen 6:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

ALL.IDLETIMEOUT - This parameter specifies how long (in
minutes) a connection can remain inactive before it is cut
off. If set to 0 (the default), the connection will not
time out.

all.idletimeout[#] :

CONF.GROUP - Used to combine users into a group. This
simplifies the parameter, all.users. You can define more
than one group. (e.g. groupName: user1, user2)

conf.group[#] :sales: john, jane


Would you like to create another group? (y/n) [n] :
```

Screen 7:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
************************************************************

Current configuration:
(The ones with the '#' means it's not activated.)

all.ipno : #
all.socket_port : 7001+
all.protocol : socket_server
all.modbus_smode : #
all.users : #
all.poll_interval : #
all.tx_interval : #
all.idletimeout : #
conf.group : #

Are these configuration(s) all correct? (y/n) [n]:
```

If you type 'n':

```
Type 'c' to go back and CORRECT these parameters or 'q' to
QUIT :
```

Typing 'c' repeats the application, typing 'q' exits the entire wiz application.

If you type 'y':

```
Discard previous port-specific parameters? (y/n) [n] :
```

Note: Answering yes to this question will discard only the parameter(s) which you are currently configuring if they were configured for a specific port in a previous session. For instance, if you are currently configuring parameter, all.x, and there was a specific port, s2.x, configured; then, answering yes to this question will discard s2.x.

```
Type 'c' to CONTINUE to set these parameters for specific
ports or 'q' to QUIT :
```

Typing 'c' leads to Screen 8, typing 'q' leads to Screen 9.

Screen 8:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


You have 8 available ports on this system.

Type 'q' to quit, a valid port number[1-8], or anything
else to refresh :
```

Note: The number of available ports depends on the system you are on. Typing in a valid port number repeats this program except this time it's configuring for the port number you have chosen. For “wiz --ac cas,” an additional parameter is asked: serverfarm. Typing 'q' leads to Screen 9.

Screen 9:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************

(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)


Do you want to activate your configurations now? (y/n) [y] :
```

Screen 10:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Flash refers to a type of memory that can be erased and
reprogrammed in units of memory known as blocks rather than
one byte at a time; thus, making updating to memory easier.


If you choose to save to flash, your configurations thus
far will still be in the memory of the system even after you
reboot it. If you don't save to flash and if you were to
reboot the system, all your new configurations will be lost
and you will have to reconfigure the system.


Do you want to save your configurations to flash? (y/n) [n] :
```

CLI Method

To configure certain parameters for a specific serial port:

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure the ipno:

```
config configure line <serial port number> ipno <string>
```

To configure the socket_port:

```
config configure line <serial port number> socket <number>
```

To configure the protocol. <string> is the type of protocol desired:

```
config configure line <serial port number> protocol <string>
```

To configure modbus_smode:

```
config configure line <serial port number> modbus <string>
```

To configure users:

```
config configure line <serial port number> users <string>
```

To configure the poll_interval:

```
config configure line <serial port number> pollinterval
<number>
```

To configure tx_interval:

```
config configure line <serial port number> txinterval <num-
ber>
```

To configure idletimeout:

```
config configure line <serial port number> idletimeout <number>
```

To configure conf.group:

```
config configure conf group <string>
```

Tip. You can configure all the parameters for a serial port in one line.

```
config configure line <serial port number> tty <string>
ipno <string> socket <number> protocol <string>
modbus <string> users <string> pollinterval <number>
txinterval <number> idletimeout <number>
```

Step 2: Activate and Save.

To activate your new configurations and save them to flash, type:

```
config write
```

(This is essentially typing *signal_ras hup* and *saveconf* from the normal terminal prompt.)

Chapter 3 - Additional Features

Configuration for TS

Parameters and Passed Values

For TS configuration, you will need to configure the following parameters:

all.host The IP address of the host to which the terminals will connect.

all.protocol For the terminal server configuration, the possible protocols are login (which requests username and password), rlogin (receives username from the Secure Console Port Server SSH and requests a password), telnet, ssh, ssh2, or socket_client. If the protocol is configured as telnet or socket_client, the parameter socket_port needs to be configured.

all.socket_port This parameter is valid only if all.protocol is configured as socket_client or telnet. The socket_port is the TCP port number of the application that will accept connections requested by this serial port.

all.telnet_client_mode When the protocol is TELNET, this parameter configured as BINARY (1) causes an attempt to negotiate the TELNET BINARY option on both input and output with the Telnet server. So it puts the telnet client in binary mode. The acceptable values are "0" or “1”, where “0” is text mode (default) and “1” is a binary mode.

all.userauto
(unique to TS) Username used when connected to a UNIX server from the user’s serial terminal.

vi Method

The parameters described above must be changed by directly editing the /etc/portslave/pslave.conf file.

Browser Method

Step 1: Follow the steps 1 to 4 in the section titled Configuration for CAS, “Browser Method” on page 81.

Step 2: Click the TS Profile button in the Wizard section.

Configure the following parameters:

Profile section:	Protocol (telnet, ssh, rlogin or socket client) Socket port (23 for telnet, 22 for ssh, 513 for rlogin)
Terminal Server section:	Host (the name or the IP address of the host) Automatic User

Step 3: Click the Submit button.

At this point, the configuration file is written in the RAMdisk.

Step 4: Make changes effective.

Click on the Administration > Run Configuration link, check the Serial Ports/ Ethernet/Static Routes box and click on the Activate Configuration button.

Step 5: Save it in the flash.

Go to the link Administration > Load/Save Configuration and click the Save to Flash button.

Wizard Method

Step 1: Bring up the wizard.

At the command prompt, type the following to bring up the Access Method custom wizard:

```
wiz --ac ts
```

This will bring up Screen 1:

Screen 1:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************

INSTRUCTIONS for using the Wizard:
You can:
1) Enter the appropriate information for your system
and press ENTER. Enter '#' if you want to
deactivate that parameter or
2) Press ENTER if you are satisfied with the value
within the brackets [ ] and want to go on to the
next parameter or
3) Press ESC if you want to exit.

NOTE: For some parameters, if there is nothing within
the brackets, it will continue to ask for a value.
In that case, you must enter a valid value or # if you
do not wish to configure the value.

Press ENTER to continue...
```

Screen 2:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************

Current configuration:
(The ones with the '#' means it's not activated.)

all.protocol : rlogin
all.socket_port : 23
all.telnet_client_mode : 0
all.userauto : #

Set to defaults? (y/n) [n] :
```

Screen 3:

```
***********************************************************
********* C O N F I G U R A T I O NW I Z A R D *********
***********************************************************

ALL.PROTOCOL - Users can access the servers through the
serial port using ssh, ssh2, telnet, login, rlogin,
or socket_client.
(e.g. login -requests username and password, rlogin -
receives username from the system and requests a password,
etc.)

all.protocol[rlogin] :


ALL.SOCKET_PORT - This defines the port(s) to be used by
the protocols telnet and socket_client. For these two
protocols a default value of 23 is used when no value
is configured.

all.socket_port[23] :
```

Screen 4:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

ALL.TELNET_CLIENT_MODE - This parameter only applies if
the current protocol configured is telnet. Configuring as
binary (1) causes an attempt to negotiate the TELNET
BINARY option on both input and output with the Telnet
server. Thus, it puts the telnet client in binary mode.
The default is 0 which represents text mode.

all.telnet_client_mode[0] :
```

```
ALL.USERAUTO - Username used when connected to a Unix
server from the user's serial terminal.

all.userauto[#] :
```

Note: all.host is configured under the wiz - - tso.

Screen 5:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Current configuration:
(The ones with the '#' means it's not activated.)

all.protocol : rlogin
all.socket_port : 23
all.telnet_client_mode : 0
all.userauto : #
Are these configuration(s) all correct? (y/n) [n]:
```

If you type 'n'

```
Type 'c' to go back and CORRECT these parameters or 'q' to
QUIT :
```

Typing 'c' repeats the application, typing 'q' exits the entire wiz application

If you type 'y'

```
Discard previous port-specific parameters? (y/n) [n] :
```

Note: Answering yes to this question will discard only the parameter(s) which you are currently configuring if they were configured for a specific port in a previous session. For instance, if you are currently configuring parameter, all.x, and there was a specific port, s2.x, configured; then, answering yes to this question will discard s2.x.

```
Type 'c' to CONTINUE to set these parameters for specific
ports or 'q' to QUIT :
```

Typing 'c' leads to Screen 6, typing 'q' leads to Screen 7.

Screen 6:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


You have 8 available ports on this system.

Type 'q' to quit, a valid port number[1-8], or anything
else to refresh :
```

Note: The number of available ports depends on the system you are on. Typing in a valid port number repeats this program except this time it's configuring for the port number you have chosen. Typing 'q' leads to Screen 7.

Screen 7:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)


Do you want to activate your configurations now? (y/n) [y] :
```

Screen 8:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


Flash refers to a type of memory that can be erased and
reprogrammed in units of memory known as blocks rather than
one byte at a time; thus, making updating to memory easier.

If you choose to save to flash, your configurations thus
far will still be in the memory of the system even after you
reboot it. If you don't save to flash and if you were to
reboot the system, all your new configurations will be lost
and you will have to reconfigure the system.

Do you want to save your configurations to flash? (y/n) [n] :
```

CLI Method

To configure certain parameters for a specific serial port:

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure the protocol (<string> is the type of protocol desired):

```
config configure line <serial port number> protocol <string>
```

To configure the socket_port:

```
config configure line <serial port number> socket <number>
```

To configure the telnet_client_mode:

```
config configure line <serial port number> telnetclientmode
<number>
```

To configure userauto:

```
config configure line <serial port number> userauto <string>
```

Tip. You can configure all the parameters for a serial port in one line.

```
config configure line <serial port number> tty <string>
protocol <string> socket <number> telnetclientmode
<number> userauto <string>
```

Step 2: Activate and Save.

To activate your new configurations and save them to flash, type:

```
config write
```

(This is essentially typing *signal_ras hup* and *saveconf* from the normal terminal prompt.)

Chapter 3 - Additional Features

Configuration for Dial-in Access

Parameters and Passed Values

The parameters that need to be configured are shown in the following list. *Note: The character "\" at the end of a line means that the string continues on the next line.*

conf.pppd Location of the ppp daemon with Radius. Default value: /usr/local/sbin/pppd.

all.ipno This is the default IP address of the 's serial ports. Any host can access a port using its IP address as long as a path to the address exists in the host's routing table. An example value would be 192.168.1.101+. The "+" indicates that the first port should be addressed as 192.168.1.101 and the following ports should have consecutive values.

all.initchat Modem initialization string. Example value:

```
TIMEOUT 10 "" \d\l\dATZ \OK\r\n-ATZ-OK\r\n “” \“” ATMO OK\R\N “”\
TIMEOUT 3600 RING “” \
STATUS Incoming %p:I.HANDSHAKE “” ATA\
TIMEOUT 60 CONNECT@ “” \
STATUS Connected %p:I.HANDSHAKE
```

all.autoppp Options to auto-detect a ppp session. The cb-script parameter defines the file used for callback and enables negotiation with the callback server. Callback is available in combination with Radius Server authentication. When a registered user calls the Secure Console Port Server SSH, it will disconnect the user, then call the user back. The following three parameters must be configured in the Radius Server.

- attribute Service_type(6): Callback Framed;
- attribute Framed_Protocol(7): PPP;
- attribute Callback_Number(19): the dial number (example: 50903300).

Example value:

```
%j novj \
proxyarp modem asyncmap 000A0000 \
noipx noccp login auth require-pap refusechap\
mtu %t mru %t \
cb-script /etc/portslave/cb_script \
plugin /usr/lib/libpsr.so
```

all.pppopt PPP options when user has already been authenticated.

Example value:

```
%i:%j novj \
proxyarp modem asyncmap 000A0000 \
noipx noccp mtu %t mru %t netmask%m \
idle %I maxconnect %T \
plugin /usr/lib/libpsr.so
```

all.protocol For the Dial-in configuration, the available protocols are PPP, SLIP and CSLIP.

> **Tip**. Documentation about PPP options can be found on the Linux pppd man page.

vi Method

The parameters described above must be changed by directly editing the /etc/portslave/pslave.conf file.

Browser Method

For the serial ports you would have all the parameters described above but conf.*.
To configure Access Method with your browser:

Step 1: Follow the steps 1 to 4 in the section titled Configuration for CAS, “Browser Method” on page 81.

Step 2: Click the Dial in Profile button in the Wizard section.

Step 3: Scroll down to the Profile section.

You can change the settings for *all.ipno* and *all.protocol* in this section.

Step 4: Scroll to the modem Section.

You can configure the parameter all.initchat here.

Step 5: Scroll to the PPP Section.

You can configure the parameter *all.autoppp and all.pppopt* here.

Step 6: Click the Submit button.

At this point, the configuration file is written in the RAMdisk.

Step 7: Make the changes effective.

Click on the Administration > Run Configuration link, check the Serial Ports/ Ethernet/Static Routes box and click on the Activate Configuration button.

Step 8: Save it in the flash.

Go to the link Administration > Load/Save Configuration and click the Save to Flash button.

CLI Method

To configure certain parameters for a specific serial port:

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure the protocol. <string> is the type of protocol desired:

```
config configure line <serial port number> protocol <string>
```

To configure ipno:

```
config configure line <serial port number> ipno <string>
```

Tip. You can configure all the parameters for a serial port in one line.

```
config configure line <serial port number> tty <string>
protocol <string> ipno <string>
```

Step 2: Activate and Save.

To activate your new configurations and save them to flash, type:

```
config write
```

(This is essentially typing *signal_ras hup* and *saveconf* from the normal terminal prompt.)

Chapter 3 - Additional Features

Authentication

Authentication is the process of identifying an individual, usually based on a username and password. In security systems, authentication is distinct from authorization, which is the process of giving individuals access to system objects based on their identity. Authentication merely ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual. With the Secure Console Port Server SSH, authentication can be performed locally, or with a remote Radius, Tacacs, or ldap database,.

Parameters Involved and Passed Values

The authentication feature utilizes the following parameters:

all.authtype Type of authentication used. There are several authentication type options:

- *none* (no authentication)
- *local* (authentication is performed using the /etc/passwd file)
- *remote* (This is for a terminal profile only. The unit takes in a username but does not use it for authentication. Instead it passes it to the remote server where it is then used for authentication.)
- *radius* (authentication is performed using a Radius authentication server)
- *TacacsPlus* (authentication is performed using a TacacsPlus authentication server)
- *ldap* (authentication is performed against an ldap database using an ldap server. The IP address and other details of the ldap server are defined in the file /etc/ldap.conf)

- *local/radius* (authentication is performed locally first, switching to Radius if unsuccessful)
- *radius/local* (the opposite of the previous option)
- *local/TacacsPlus* (authentication is performed locally first, switching to TacacsPlus if unsuccessful)
- *TacacsPlus/local* (the opposite of the previous option)
- *RadiusDownLocal* (local authentication is tried only when the Radius server is down)
- *TacacsPlusDownLocal* (local authentication is tried only when the TacacsPlus server is down)

Note that this parameter controls the authentication required by the Secure Console Port Server SSH. The authentication required by the device to which the user is connecting is controlled separately.

all.authhost1
all.authhost2
This address indicates the location of the Radius/TacacsPlus authentication server and is only necessary if this option is chosen in the previous parameter. A second Radius/TacacsPlus authentication server can be configured with the parameter all.authhost2.

all.accthost1
all.accthost2
This address indicates the location of the Radius/TacacsPlus accounting server, which can be used to track how long users are connected after being authorized by the authentication server. Its use is optional. If this parameter is not used, accounting will not be performed. If the same server is used for authentication and accounting, both parameters must be filled with the same address. A second Radius/TacacsPlus accounting server can be configured with the parameter all.accthost2.

all.radtimeout This is the timeout (in seconds) for a Radius authentication query to be answered.

all.radretries Defines the number of times each Radius/ TacacsPlus server is tried before another is contacted. The first server (authhost1) is tried “radretries” times, and then the second (authhost2), if configured, is contacted “radretries” times. If the second also fails to respond, Radius/ TacacsPlus authentication fails.

all.secret This is the shared secret (password) necessary for communication between the Secure Console Port Server SSH and the Radius/TacacsPlus servers.

Configuration for CAS, TS, and Dial-in Access

vi Method

The parameters described above must be changed by directly editing the /etc/portslave/pslave.conf file.

Browser Method

To configure Authentication with your browser:

Step 1: Follow the steps 1 to 4 in the section titled Configuration for CAS, “Browser Method” on page 81**.**

Step 2: Scroll to the Authentication section.

Scroll down to the Authentication section and configure the parameters in this section.

Step 3: Click the Submit button.

At this point, the configuration file is written in the RAMdisk.

Step 4: Make changes effective.

Click on the Administration > Run Configuration link, check the Serial Ports/ Ethernet/Static Routes box and click on the Activate Configuration button.

Step 5: Save it in the flash.

Go to the link Administration > Load/Save Configuration and click the Save to Flash button.

Wizard Method

Step 1: Bring up the wizard.

At the command prompt, type the following to bring up the Authentication custom wizard:

```
wiz --auth
```

Screen 1 will appear.

Screen 1:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

INSTRUCTIONS for using the Wizard:
You can:
1) Enter the appropriate information for your system
and press ENTER. Enter '#' if you want to
deactivate that parameter or
2) Press ENTER if you are satisfied with the value
within the brackets [ ] and want to go on to the
next parameter or
3) Press ESC if you want to exit.

NOTE: For some parameters, if there is nothing within
the brackets, it will continue to ask for a value.
In that case, you must enter a valid value or # if you
do not wish to configure the value.

Press ENTER to continue...
```

Screen 2:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Current configuration:
(The ones with the '#' means it's not activated.)

all.authtype : none
all.authhost1 : 192.168.160.3
all.accthost1 : 192.168.160.3
all.authhost2 : 192.168.160.4
all.accthost2 : 192.168.160.4
all.radtimeout : 3
all.radretries : 5
all.secret : secret

Set to defaults? (y/n) [n] :
```

Screen 3:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


ALL.AUTHTYPE - This parameter controls the authentication
required by the system. Users' access to the server
through the serial port is granted through the check of
username and password locally or remotely.
(e.g. none, local, TacacsPlus (note the
capital 'T' in TacacsPlus), radius, ldap, etc.


all.authtype[none] :
```

Note: If authtype is configured as *none*, *local*, or *ldap*, the application will skip immediately to the summary screen because the rest of the parameters pertain only if the system is configured to use a Radius or TacacsPlus server. Configurations for ldap are done in /etc/ldap.conf.

```
ALL.AUTHHOST1 - This IP address indicates where the
Radius or TacacsPlus authentication server is located.

all.authhost1[200.200.200.2] :
```

Screen 4:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


ALL.ACCTHOST1 - This IP address indicates where the Radius
or TacacsPlus accounting server is located. The accounting
server can be used to track how long users are connected
after being authorized by the authentication server.
all.accthost1[200.200.200.3] :



ALL.AUTHHOST2 - This IP address indicates where the SECOND
Radius or TacacsPlus authentication server is located.

all.authhost2[200.200.200.2] :
```

Screen 5:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

ALL.ACCTHOST2 - This IP address indicates where the SECOND
Radius or TacacsPlus accounting server is located.

all.accthost2[200.200.200.3] :

ALL.RADTIMEOUT- This is the timeout (in seconds) for a
Radius or TacacsPlus authentication query to be answered.

all.radtimeout[3] :
```

Screen 6:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

ALL.RADRETRIES - This defines the number of times each
Radius or TacacsPlus server is tried before another is
contacted.

all.radretries[5] :

ALL.SECRET - This is the shared secret necessary for
communication between the system and the Radius or
TacacsPlus servers.

all.secret[secret] :
```

Screen 7:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
Current configuration:
(The ones with the '#' means it's not activated.)

all.authtype : none
all.authhost1 : 200.200.200.2
all.accthost1 : 200.200.200.3
all.authhost2 : 200.200.200.2
all.accthost2 : 200.200.200.3
all.radtimeout : 3
all.radretries : 5
all.secret : rad-secret

Are these configuration(s) all correct? (y/n) [n] :
```

If you type 'n'

```
Type 'c' to go back and CORRECT these parameters or 'q' to
QUIT :
```

Typing 'c' repeats application, typing 'q' exits the entire wiz application

If you type 'y'

```
Discard previous port-specific parameters? (y/n) [n] :
```

Note: Answering yes to this question will discard only the parameter(s) which you are currently configuring if they were configured for a specific port in a previous session. For instance, if you are currently configuring parameter, all.x, and there was a specific port, s2.x, configured; then, answering yes to this question will discard s2.x.

```
Type 'c' to CONTINUE to set these parameters for specific
ports or 'q' to QUIT :
```

Typing 'c' leads to Screen 8, typing 'q' leads to Screen 9.

Screen 8:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
You have 8 available ports on this system.

Type 'q' to quit, a valid port number[1-8], or anything
else to refresh :
```

Note: The number of available ports depends on the system you are on. Typing in a valid port number repeats this program except this time it's configuring for the port number you have chosen. Typing 'q' leads to Screen 9.

Screen 9:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************

(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)

Do you want to activate your configurations now? (y/n) [y] :
```

Screen 10:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Flash refers to a type of memory that can be erased and
reprogrammed in units of memory known as blocks rather than
one byte at a time; thus, making updating to memory easier.

If you choose to save to flash, your configurations thus
far will still be in the memory of the system even after you
reboot it. If you don't save to flash and if you were to
reboot the system, all your new configurations will be lost
and you will have to reconfigure the system.

Do you want to save your configurations to flash? (y/n) [n] :
```

CLI Method

To configure certain parameters for a specific serial port.

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure authtype:

```
config configure line <serial port number> authtype <string>
```

To configure authhost1:

```
config configure line <serial port number> authhost1
<string>
```

To configure accthost1:

```
config configure line <serial port number> accthost1
<string>
```

To configure authhost2:

```
config configure line <serial port number> authhost2
<string>
```

To configure accthost2:

```
config configure line <serial port number> accthost2
<string>
```

To configure radtimeout:

```
config configure line <serial port number> timeout <number>
```

To configure radretries:

```
config configure line <serial port number> retries <number>
```

To configure secret:

```
config configure line <serial port number> secret <string>
```

Tip. You can configure all the parameters for a serial port in one line.

```
config configure line <serial port number> tty <string>
authtype <string> authhost1 <string> accthost1 <string>
authhost2 <string> accthost2 <string> timeout <number>
retries <number> secret <string>
```

Step 2: Activate and Save.

To activate your new configurations and save them to flash, type:

```
config write
```

CAS Port Pool

This feature is available for the Secure Console Port Server SSH Version 1.3.7 onward. CAS Port Pooling allows you to access a free serial port from a pool in addition to the original feature where you could access a specific serial port. When you access a serial port through the pool the features sniff session and multiple sessions are not available. This feature is available for serial ports configured as CAS profile only.

You can define more than one pool of serial ports. Each serial port can only belong to ONE pool.The pool is uniquely identified by a four parameter scheme:

- protocol,
- pool_ipno,
- pool_serverfarm, and
- pool_socket_port

The three new parameters: pool_ipno, pool_serverfarm, and pool_socket_port
have the same meaning as ipno, serverfarm, and socket_port respectively. Ports belonging to the same pool MUST be configured with the same value in these fields.

It is strongly recommended that you configure the same values in all parameters related to authentication for all serial ports belonging to a pool. Some of the authentication parameters are users, admin_users, and authtype.

You can access the serial ports from a pool with the same commands you use today to access a specific serial port. You just need to use pool_ipno, pool_serverfarm, or pool_socket_port instead ipno, serverfarm, or socket_port respectively in the ssh/telnet command.

When a connection request arrives using one of pool_ipno, pool_serverfarm, or pool_socket_port the Secure Console Port Server SSH will look for the first free serial port from the pool and that port will be assigned to connection. If there is no serial port free in the pool the connection is just dropped.

Chapter 3 - Additional Features

How to Configure it

Following is an example of serial port pool configuration:

```
#
# Serial port pool: pool-1
#

s1.tty ttyS1
s1.protocol socket_server
s1.socket_port 7001 // TCP port # for specific allocation
s1.pool_socket_port 3000 // TCP port # for the pool
s1.ipno 10.0.0.1 // IP address for specific allocation
s1.pool_ipno 10.1.0.1 // IP address for the pool
s1.serverfarm serial-1 // alias for specific allocation
s1.pool_serverfarm pool-1 // alias for the pool

s2.tty ttyS2
s2.protocol socket_server
s2.socket_port 7002 // TCP port # for specific allocation
s2.pool_socket_port 3000 // TCP port # for the pool
s2.ipno 10.0.0.2 // IP address for specific allocation
s2.pool_ipno 10.1.0.1 // IP address for the pool
s2.serverfarm serial-2 // alias for specific allocation
s2.pool_serverfarm pool-1 // alias for the pool

#
# Serial port pool: pool-2
#

s3.tty ttyS3
s3.protocol socket_ssh
s3.socket_port 7003 // TCP port # for specific allocation
s3.pool_socket_port 4000 // TCP port # for the pool
s3.ipno 10.0.0.3 // IP address for specific allocation
s3.pool_ipno 10.2.0.1 // IP address for the pool
s3.serverfarm serial-3 // alias for specific allocation
s3.pool_serverfarm pool-2 // alias for the pool

s4.tty ttyS4
s4.protocol socket_ssh
s4.socket_port 7004 // TCP port # for specific allocation
```

```
s4.pool_socket_port 4000 // TCP port # for the pool
s4.ipno 10.0.0.4 // IP address for specific allocation
s4.pool_ipno 10.2.0.1 // IP address for the pool
s4.serverfarm serial-4 // alias for specific allocation
s4.pool_serverfarm pool-2 // alias for the pool
```

In the example above, there are two pools:

- *pool-1* (identified by Protocol socket_server, TCP port #3000, IP 10.1.0.1, and alias pool-1)
- *pool-2* (identified by Protocol socket_ssh, TCP port #4000, IP 10.2.0.1, and alias pool-2)

The serial ports ttyS1 and ttyS2 belong to the pool-1. The serial ports ttyS3 and ttyS4 belong to the pool-2.

You can access specifically serial port ttyS1 by using TCP port 7001, IP address 10.0.0.1 or alias serial-1. If the ttyS1 is being used by somebody else the connection will be dropped if the user is not a admin_user. Alternately, you can access ttyS1 through pool (if it's free) using TCP port 3000, IP 10.1.0.1 or alias pool-1. If it is not free ttyS2 will be automatically allocated. Additionally, if ttyS2 is not free, the connection will be dropped.

Chapter 3 - Additional Features

Clustering

Clustering is available for the Secure Console Port Server SSH 1.3.0 and up (except for the 1-Port). It allows the stringing of Terminal Servers so that one Master Secure Console Port Server SSH can be used to access all Secure Console Port Server SSHs on a LAN. The MasterSecure Console Port Server SSH can manage up to 1024 serial ports, so that the following can be clustered:

- 1 Master 16-Port + 31 Slave16-Ports
- 1 Master 32-Port+ 15 Slave 32-Ports, or
- 1 Master 48-Port + 9 slave 48-Ports + 1 slave 32-Port

An example with one Master Secure Console Port Server SSHand two Slave Secure Console Port Server SSHs is shown in the following figure.

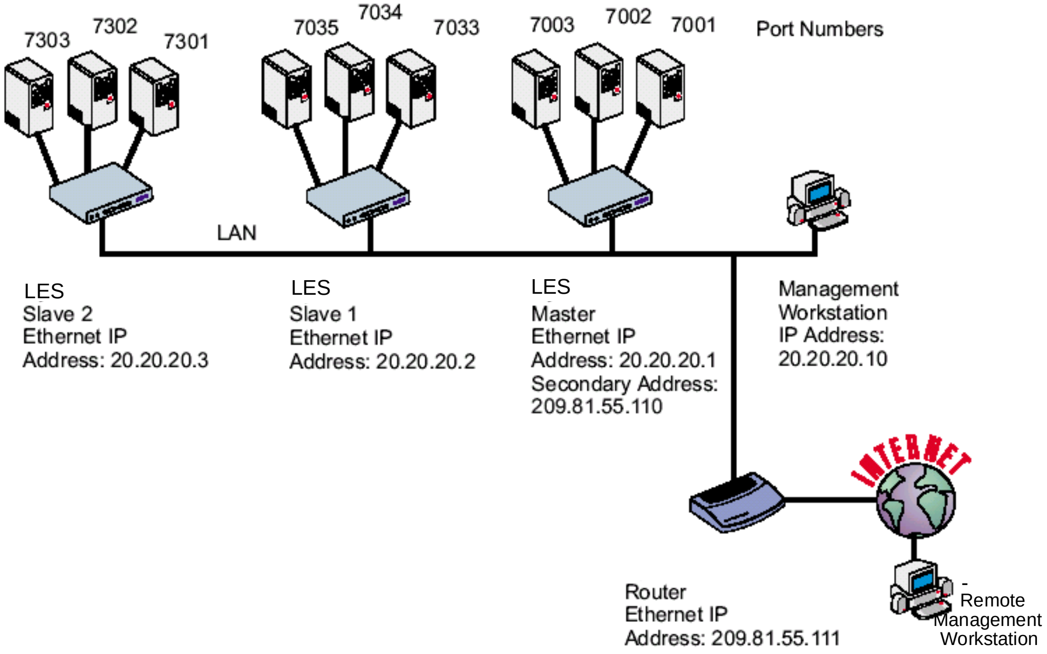


Figure 21: An example of the clustering feature

Parameters Involved and Passed Values

The Master Secure Console Port Server SSH must contain references to the Slave ports. The configuration described earlier for Console Access Servers should be followed with the following exceptions for the Master and Slaves:

Table 7: Master Black Box Configuration (where it differs from the CAS standard)

Parameter	Description	Value for this example
conf.eth_ip	Ethernet Interface IP address.	20.20.20.1
conf.eth_ip_alias	Secondary IP address for the Ethernet Interface (needed for clustering feature).	
conf.eth_mask_alias	Mask for secondary IP address above.	255.255.255.0
all.socket_port	This value applies to both the local ports and ports on Slave Secure Console Port Server SSH.	7001+
all.protocol	Depends on the application.	Socket_ssh or socket_server
all.authtype	Depends on the application.	Radius or local or none
s33.tty	This parameter must be created in the Master Secure Console Port Server SSH file for every Slave port. Its format is: IP_of_Slave:[slave_socket_port] for non-Master ports. In this case, the slave_socket_port value is not necessary because s33.socket_port is automatically set to 7033 by all.socket_port above.	20.20.20.2:7033
s33.serverfarm	An alias for this port.	Server_on_slave1_serial_s1

Table 7: Master Black Box Configuration (where it differs from the CAS standard)

Parameter	Description	Value for this example
s33.ipno	This parameter must be created in the Master Secure Console Port Server SSH file for every Slave port, unless configured using all.ipno.	0.0.0.0
s34.tty	See s33.tty.	20.20.20.2:7034
s34.serverfarm	An alias for this port.	Server_on_slave1_ serial_s2
s34.ipno	See s33.ipno.	0.0.0.0
s35.tty	See s33.tty.	20.20.20.2:7035
s35.serverfarm	An alias for this port.	Server_on_slave1_ serial_s3
s35.ipno	See s33.ipno.	0.0.0.0
etc. for s36-s64		
S65.tty	The format of this parameter is IP_of_Slave:[slave_socket_port] for non-Master ports. The value 7301 was chosen arbitrarily for this example.	20.20.20.3:7301
S65.serverfarm	An alias for this port.	Server_on_slave2_ serial_s1
S65.ipno	See s33.ipno.	0.0.0.0
S66.tty	See s65.tty	20.20.20.3:7302
S66.serverfarm	An alias for this port.	Server_on_slave2_ serial_s2
S66.ipno	See s33.ipno.	0.0.0.0
S67.tty	See s65.tty.	20.20.20.3:7303

Table 7: Master Black Box Configuration (where it differs from the CAS standard)

Parameter	Description	Value for this example
S67.serverfarm	An alias for this port.	Server_on_slave2_ serial_s3
S67.ipno	See s33.ipno.	0.0.0.0
etc. for s68-s96		

The Slave Secure Console Port Server SSHs do not need to know they are being accessed through the Master Secure Console Port Server SSH. (You are creating virtual terminals: virtual serial ports.) Their port numbers, however, must agree with those assigned by the Master.

Table 8: Secure Console Port Server SSH configuration for Slave 1 (where it differs from the CAS standard)

Parameter	Value for this example
all.protocol	socket_server
all.authtype	none
conf.eth_ip	20.20.20.2
all.socket_port	7033+
all.authtype	none

Table 9: Secure Console Port Server SSH configuration for Slave 2 (where it differs from the CAS standard)

Parameter	Value for this example
all.protocol	socket_server

Table 9: Secure Console Port Server SSH configuration for Slave 2 (where it differs from the CAS standard)

Parameter	Value for this example
all.authtype	none
conf.eth_ip	20.20.20.3
all.authtype	none
all.socket_port	7301+

To access ports from the remote management workstation, use telnet with the secondary IP address:

```
telnet 209.81.55.110 7001
```

to access the first port of the Master Secure Console Port Server SSH.

```
telnet 209.81.55.110 7033
```

to access the first port of Slave 1.

```
telnet 209.81.55.110 7065
```

to access the first port of Slave 2.

Ssh can also be used from the remote management workstation:

```
ssh -l <username>:Server_on_slave2_serial_s3 209.81.55.110
```

to access the third port of Slave 2, or

```
ssh -l <username>:7069 209.81.55.110
```

to access the fifth port of Slave 2.

Centralized Management - the Include File

The Secure Console Port Server SSH allows centralized management through the use of a Master pslave.conf file. Administrators should consider this approach to configure multiple Secure Console Port Server SSH. Using this feature, each unit has a simplified pslave.conf file where a Master include file is cited. This common configuration file contains information for all units, properly divided in separate sections, and would be stored on one central server.

This file, in our example shown in Figure 22: Example of Centralized Management, is /etc/portslave/TScommon.conf. It must be downloaded to each Secure Console Port Server SSH.

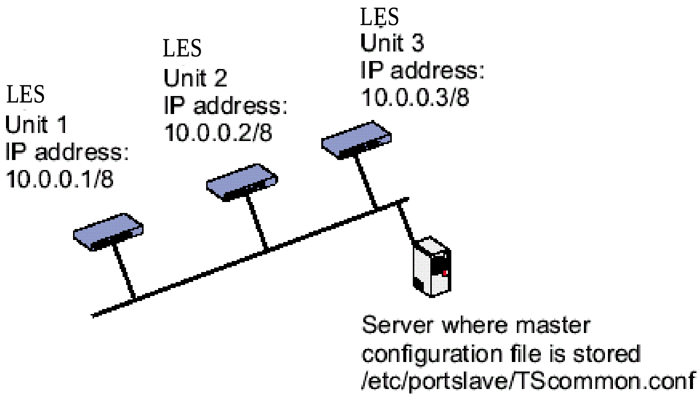


Figure 22: Example of Centralized Management

The abbreviated pslave.conf and /etc/hostname files in each unit, for the example are:

For the /etc/hostname file in *unit 1*:

```
unit1
```

For the plsave.conf file in *unit 1*:

```
conf.eth_ip 10.0.0.1

conf.eth_mask 255.0.0.0

conf.include /etc/portslave/Scommon.conf
```

For the /etc/hostname file in *unit 2*:

```
unit2
```

For the plsave.conf file in *unit 2*:

```
conf.eth_ip 10.0.0.2

conf.eth_mask 255.0.0.0
conf.include /etc/portslave/TScommon.conf
```

For the /etc/hostname file in *unit 3*:

```
unit3
```

For the plsave.conf file in *unit 3*:

```
conf.eth_ip 10.0.0.3

conf.eth_mask 255.0.0.0

conf.include /etc/portslave/TScommon.conf
```

The common include file for the example is:

```
conf.host_config unit1

<parameters for unit1 following the rules for pslave.conf>

conf.host_config unit2

<parameters for unit2 following the rules for pslave.conf>

conf.host_config unit3

<parameters for unit3 following the rules for pslave.conf>

conf.host_config.end
```

When this file is included, unit1 would read only the information between *conf.host_config unit1* and *conf.host_config unit2*. Unit2 would use only the information between *conf.host_config unit2* and *conf.host_config unit3* and unit3 would use information after *conf.host_config unit3* and before *conf.host_config.end*.

Steps for using Centralized Configuration

Step 1: Create and save the */etc/portslave/pslave.conf* **and** */etc/hostname* **files in each Secure Console Port Server SSH.**

Step 2: Execute the command *signal_ras hup* **on each unit.**

Step 3: Create, save, and download the common configuration.

Create and save the common configuration file on the server, then download it (probably using scp) to each unit. Make sure to put it in the directory set in the pslave.conf file (/etc/portslave in the example).

Step 4: Execute the command signal_ras hup on each unit again.

Step 5: Test each unit.

If everything works, add the line /etc/portslave/TScommon.conf to the /etc/config_files file.

Step 6: Save the file and close it.

Step 7: Execute the *saveconf* command.

Note: The included file /etc/portslave/TScommon.conf cannot contain another include file (i.e., the parameter conf.include must not be defined).

Also, <max ports of Secure Console Port Server SSH> + N(+) is done same way as serial port.

Chapter 3 - Additional Features

CronD

CronD is a service provided by the Secure Console Port Server SSH system that allows automatic, periodically-run custom-made scripts. It replaces the need for the same commands to be run manually.

Parameters Involved and Passed Values

The following parameters are created in the /etc/crontab_files file:

status Active or inactive. If this item is not active, the script will not be executed.

user The process will be run with the privileges of this user, who must be a valid local user.

source Pathname of the crontab file that specifies frequency of execution, the name of shell script, etc. It should be set using the traditional crontab file format.

Example:
The name of the shell script with the commands to be executed is */etc/teste_cron.sh.*
The name of the crontab file is */etc/crontab_tst* and it contains one line:

```
0-59 * * * * /etc/test_cron.sh
```

Insert the follow line in the */etc/crontab*_files:

```
active root /etc/crontab_tst
```

Result: CronD will execute the shell script teste_cron.sh with root privileges each minute.

Note: In /etc/crontab, you can only have one active entry per user. For instance, from the example above, you cannot add another active entry for root because it already has an entry. If you want to add more scripts, you can just add them to the source file (/etc/crontab_tst).

Configuration for CAS, TS, and Dial-in Access

Important! After creating the shell script and *crontab* file and modifying the *crontab_files* file, make sure the file named */etc/config_files* contains the names of all files that should be saved to flash. Run the command *saveconf* after this confirmation.

vi Method

The files Crontab and shell script are created and the file */etc/crontab_files* is modified as indicated.

To use cronD:

Step 1: Create the files for every process that it will execute:

Step 2: Create a line in the file /etc/crontab_files for each process to be run.

Step 3: Update the system.

The next step is to update the system with the modified data. Make sure the file named /etc/config_files contains the names of all files that should be saved to flash.

Step 4: Run ***saveconf*****.**

The command *saveconf*, which reads the /etc/config_files file, should then be run. *saveconf* copies all the files listed in the file /etc/config_files from the ramdisk to /proc/flash/script.

Step 5: Reboot the Secure Console Port Server SSH.

Browser Method

To configure CronD with your browser:

Step 1: Point your browser to the Console Server.

In the address or location field of your browser type the Console Access Server's IP address. For example:

```
http://10.0.0.0
```

Step 2: Log in as root and type the Web root password configured by the Web server.

This will take you to the Configuration and Administration page.

Step 3: Click on the Edit Text File link.

Click on this link on the Link Panel. You can then pull up the appropriate file and edit it.

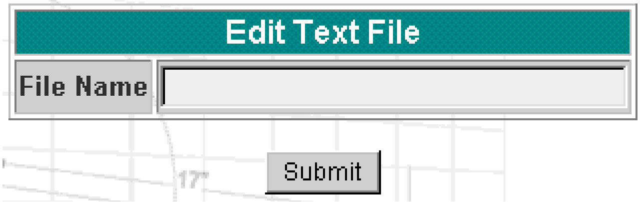


Figure 23: Edit Text File page

Data Buffering

Introduction

Data buffering can be done in local files or in remote files through NFS. When using remote files, the limitation is imposed by the remote Server (disk/partition space) and the data is kept in linear (sequential) files in the remote Server. When using local files, the limitation is imposed by the size of the available ramdisk. You may wish to have data buffering done in file, syslog or both. For syslog, *all.syslog_buffering* and *conf.DB_facility* are the parameters to be dealt with, and syslog-ng.conf file should be set accordingly. (Please see Syslog for the syslog-ng configuration file.) For the file, *all.data_buffering* is the parameter to be dealt with.

Conf.nfs_data_buffering is a remote network file system where databuffering will be written, instead of using the default directory /var/run/DB.When commented, it indicates local data buffering. The directory tree to which the file will be written must be NFS-mounted and the local path name is /mnt/DB_nfs. The remote host must have NFS installed and the administrator must create, export, and allow reading/writing to this directory. The size of this file is not limited by the value of the parameter s1.data_buffering,though the value cannot be zero since a zero value turns off data buffering.

The conf.nfs_data_buffering parameter format is:

```
<server name or IP address>:<remote pathname>
```

If data buffering is turned on for port 1, for example, the data will be stored in the file ttyS1.data (or &lt;serverfarm1&gt;.data if s1.serverfarmwas configured) in local directory /var/run/DB or in remote path name and server indicated by the conf.nfs_data_buffering.

Ramdisks

Data buffering files are created in the directory */var/run/DB*. If the parameter s<nn>.serverfarm is configured for the port <nn>, this name will be used. For example, if the serverfarm is called bunny, the data buffering file will be named bunny.data.

The shell script */bin/build_DB_ramdisk* creates a 48 Mbyte ramdisk for the Secure Console Port Server SSH. Use this script as a model to create customized ramdisks for your environment. Any user-created scripts should be listed in the file /etc/user_scripts because rc.sysinit executes all shell scripts found there. This avoids changing rc.sysinit itself.

Chapter 3 - Additional Features

Linear vs. Circular Buffering

For local data buffering, this parameter allow users to buffer data in either a circular or linear fashion. Circular format (cir) is a revolving buffer file that is overwritten whenever the limit of the buffer size (set by all.data_buffering) is reached. In linear format (lin), data transmission between the remote device and the serial port ceases once the 4k bytes Rx buffer in the kernel is reached. Then if a session is established to the serial port, the data in the buffer is shown (dont_show_DBmenu must be 2), cleared, and data transmission is resumed. Linear buffering is impossible if flow control is set to none. Default is cir.

Parameters Involved and Passed Values

Data Buffering uses the following parameters:

all.data_buffering — A non zero value activates data buffering (local or remote, according to what was configured in the parameter conf.nfs_data_buffering). If local data buffering, a file is created on the Secure Console Port Server SSH; if remote, a file is created through NFS in a remote server. All data received from the port is captured in this file. If local data buffering, this parameter means the maximum file size (in bytes). If remote, this parameter is just a flag to activate (greater than zero) or deactivate data buffering. When local data buffering is used, each time the maximum is reached the oldest 10% of stored data is discarded, releasing space for new data (FIFO system) - circular file. When remote data

all.data_buffering (cont.) — buffering is used, there's no maximum file size other than the one imposed by the remote server - linear file. This file can be viewed using the normal UNIX tools (cat, vi, more, etc.). *Size is in bytes not kilobytes.*

conf.nfs_data_buffering

This is the Remote Network File System where data captured from the serial port will be written instead of being written to the local directory */var/run/ DB*. The directory tree to which the file will be written must be NFS-mounted, so the remote host must have NFS installed and the administrator must create, export and allow reading/ writing to this directory. The size of this file is not limited by the value of the parameter all.data_buffering, though the value cannot be zero since a zero value turns off data buffering. The size of the file is dependent on the NFS server only (hard drive, partition size, etc.).

all.DB_mode

When configured as cir for circular format, the buffer is like a revolving file that is overwritten whenever the limit of the buffer size (as configured in all.data_buffering or s<n>.data_buffering) is reached. When configured as lin for linear format, once 4k bytes of the Rx buffer in the kernel is reached, a flow control stop (RTS off or XOFF-depending on how all.flow or s<n>.flow is set) is issued to prevent the serial port from receiving further data from the remote. Then when a session is established to the serial port, a flow control start (RTS on or XON) will be issued and data reception will then resume. If all.flow or s<n>.flow is set to none, linear buffering isn’t possible. Default is cir.

all.syslog_buffering

When nonzero, the contents of the data buffer are sent to the syslog-ng every time a quantity of data equal to this parameter is collected. The syslog level for data buffering is hard coded to level 5 (notice) and facility is local plus conf.DB_facility. The file */etc/syslog-ng/syslog-ng.conf* should be set accordingly for the syslog-ng to take some action.

all.syslog_sess

This parameter determines whether syslog is generated when a user is connected to the port or not. Originally, syslog is always generated whether the user is connected to the port or not. Now, users have the option to NOT have syslog generate messages when they connect to a port. This feature does not affect the local data_buffering file. When set to 0 (default), syslog is always generated. When set to 1, syslog is only generated when the user is NOT connected to the port sending the data. When the user does connect to the port that is sending data, syslog messages won't be generated.

all.dont_show_DBmenu

When zero, a menu with data buffering options is shown when a nonempty data buffering file is found. When 1, the data buffering menu is not shown. When 2, the data buffering menu is not shown but the data buffering file is shown if not empty. When 3, the data buffering menu is shown, but without the erase and show and erase options.

all.DB_timestamp

Records the time stamp in the data buffering file (1) or not (0). If it is configured as 1, the software will accumulate input characters until it receives a CR and LF from the serial port or the accumulated data reaches 256 characters. Either way, the accumulated data will be recorded in the data buffering file along with the current time. The parameter all.data_buffering has to be with a non-zero value for this parameter to be meaningful.

Configuration for CAS

vi Method

Files to be modified:

- pslave.conf
- syslog-ng.conf

Browser Method

To configure Data Buffering with your browser:

Step 1: Point your browser to the Console Server.

In the address or location field of your browser type the Console Access Server's IP address. For example:

```
http://10.0.0.0
```

Step 2: Log in as root and type the Web root password configured by the Web server.

This will take you to the Configuration and Administration page.

Step 3: Select the Serial Ports link.

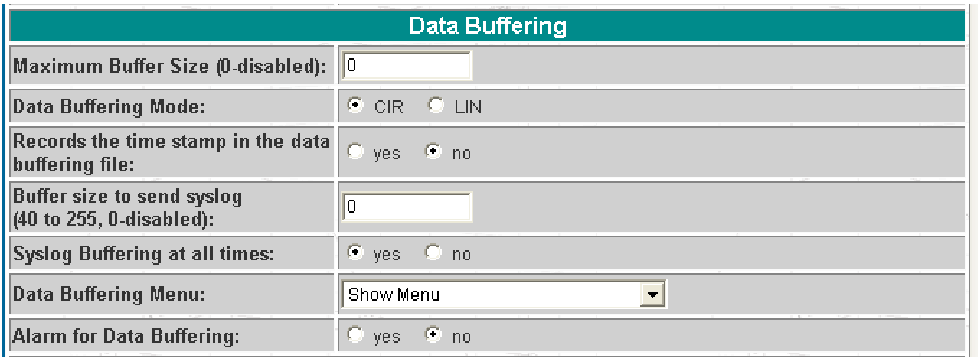
Click on the Serial Ports link on the Link Panel to the left of the page or in the Configuration section of the page. This will take you to the Port Selection page.

Step 4: Select port(s).

On the Port Selection page, choose all ports or an individual port to configure, from the dropdown menu. Click the Submit button. This will take you to the Serial Port Configuration page.

Step 5: Scroll down to the Data Buffering section.

You can change the settings in this section.



Data Buffering

Maximum Buffer Size (0-disabled):	0
Data Buffering Mode:	● CIR ○ LIN
Records the time stamp in the data buffering file:	○ yes ● no
Buffer size to send syslog (40 to 255, 0-disabled):	0
Syslog Buffering at all times:	● yes ○ no
Data Buffering Menu:	Show Menu
Alarm for Data Buffering:	○ yes ● no

Figure 24: Data Buffering section of the Serial Port Configuration page

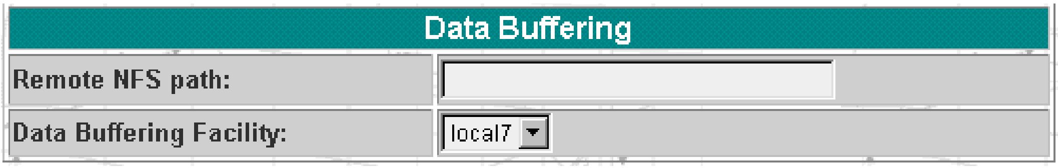
Step 6: Click the Submit button.

Step 7: Select the General link.

Click on the General link on the Link Panel to the left of the page.

Step 8: Scroll down to the Data Buffering section.

Choose whether NFS will be used or not, and choose the Data Buffering Facility level here.



Data Buffering

Remote NFS path:

Data Buffering Facility: local7

Figure 25: Data Buffering section of the General page

Step 9: Click the Submit button.

Step 10: Make the changes effective.

Click on the Administration > Run Configuration link, check the Serial Ports/Ethernet/Static Routes box and click on the Activate Configuration button.

Step 11: Click on the link Administration > Load/Save Configuration.

Step 12: Click the Save Configuration to Flash button.

Wizard Method

Step 1: Bring up the wizard.

At the command prompt, type the following to bring up the Data Buffer custom wizard:

```
wiz --db
```

Screen 1:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************

INSTRUCTIONS for using the Wizard:
You can:
1) Enter the appropriate information for your system
and press ENTER. Enter '#' if you want to
deactivate that parameter or
2) Press ENTER if you are satisfied with the value
within the brackets [ ] and want to go on to the
next parameter or
3) Press ESC if you want to exit.

NOTE: For some parameters, if there is nothing within
the brackets, it will continue to ask for a value.
In that case, you must enter a valid value or # if you
do not wish to configure the value.

Press ENTER to continue...
```

Screen 2:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Current configuration:
(The ones with the '#' means it's not activated.)

conf.nfs_data_buffering : #
all.data_buffering : 0
all.DB_mode : cir
all.dont_show_DBmenu : 0
all.DB_timestamp : 0
all.syslog_buffering : 0
all.syslog_sess : 0

Set to defaults? (y/n) [n] :
```

Screen 3:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

CONF.NFS_DATA_BUFFERING - This parameter applies only if
users choose to remotely buffer data. This is the remote
directory name where data buffering will be written to
instead of the default directory '/var/run'. If deactiva-
vated, data buffering will be done locally.

conf.nfs_data_buffering[#] :

ALL.DATA_BUFFERING - For local data buffering, this para-
meter represents the maximum file size in bytes allowed to
be captured before it is discarded for new space. If re-
mote this parameter is just a flag to either activate (any
value greater than 0) or deactivate data buffering.

all.data_buffering[0] :
```

Screen 4:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

ALL.DB_MODE - For local data buffering, this parameter allow
users to buffer data in either a circular or linear fashion.
Circular format (cir) is a revolving buffer file that is
overwritten whenever the limit of the buffer size (set by
all.data_buffering) is reached. In linear format (lin), data
transmission between the remote device and the serial port
ceases once the 4k bytes Rx buffer in the kernel is reached.
Then if a session is established to the serial port, the data
in the buffer is shown (dont_show_DBmenu must be 2), cleared,
and data transmission is resumed. Linear buffering is impos-
sible if flow control is set to none. Default is cir.

all.DB_mode[cir] :
```

```
ALL.DONT_SHOW_DBMENU - When 0, a menu with data
buffering options is shown when a non-empty data
buffering file is found. When 1, the data buffering
menu is not shown. When 2, the data buffering menu is
not shown but the data buffering file is shown if not
empty. When 3, the data buffering menu is shown, but
without the 'erase and show' and 'erase' options.

all.dont_show_DBmenu[0] :
```

Screen 5:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
ALL.DB_TIMESTAMP - Records the time stamp in the data
buffering file (1) or not (0). In case it is configured as
1, the software will accumulate input characters until it
receives a CR and LF from the serial port, or the accumu-
lated data reaches 256 characters. Either way, the accumu-
lated data will be recorded in the data buffering file
along with the current time. The parameter, all.data_buf-
fering, has to be nonzero in order for this parameter to
work.

all.DB_timestamp[0] :


ALL.SYSLOG_BUFFERING - This parameter is another option to
data buffering. Users can also have syslog perform this
function along with data buffering into files. When
nonzero, the contents of the data buffer are sent to the
syslog-ng every time a quantity of data equal to this
parameter is collected. The syslog level for data buffering
is hard coded to level 5 (notice) and facility
conf.DB_facility. The file /etc/syslog-ng/syslog-ng.conf
should be set accordingly for the syslog-ng to take some
action.


(Please see the 'Syslog-ng Configuration to use with
Syslog Buffering Feature' section under Generating Alarms
in Chapter 3 of the system's manual for the syslog-ng
configuration file.)

all.syslog_buffering[0] :
```

Screen 6:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************
ALL.SYSLOG_SESS - In order for this parameter to function,
make sure syslog buffering is activate. When set as 0,
syslog messages are always generated whether or not there
is a connection to the port that is sending data to your
unit. When set to 1, syslog messages are NOT generated when
there IS a connection to the port that is sending data. It
is only generated when there isn't a session to the port
that is sending data to your unit.

all.syslog_sess[0] :
```

Screen 7:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************

Current configuration:
(The ones with the '#' means it's not activated.)

conf.nfs_data_buffering : #
all.data_buffering : 0
all.DB_mode : cir
all.dont_show_DBmenu : 0
all.DB_timestamp : 0
all.syslog_buffering : 0
all.syslog_sess : 0

Are these configuration(s) all correct? (y/n) [n] :
```

If you type 'n'

```
Type 'c' to go back and CORRECT these parameters or 'q' to
QUIT :
```

Typing 'c' repeats the application, typing 'q' exits the entire wiz application

If you type 'y'

```
Discard previous port-specific parameters? (y/n) [n] :
```

> **Note:** Answering yes to this question will discard only the parameter(s) which you are currently configuring if they were configured for a specific port in a previous session. For instance, if you are currently configuring parameter, all.x, and there was a specific port, s2.x, configured; then, answering yes to this question will discard s2.x.

```
Type 'c' to CONTINUE to set these parameters for specific
ports or 'q' to QUIT :
```

Typing 'c' leads to Screen 8, typing 'q' leads to Screen 9.

Screen 8:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
You have 8 available ports on this system.

Type 'q' to quit, a valid port number[1-8], or anything
else to refresh :
```

Note: The number of available ports depends on the system you are on. Typing in a valid port number repeats this program except this time it's configuring for the port number you have chosen. Typing 'q' leads to Screen 9.

Screen 9:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)


Do you want to activate your configurations now? (y/n) [y] :
```

Screen 10:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Flash refers to a type of memory that can be erased and
reprogrammed in units of memory known as blocks rather than
one byte at a time; thus, making updating to memory easier.

If you choose to save to flash, your configurations thus
far will still be in the memory of the system even after you
reboot it. If you don't save to flash and if you were to
reboot the system, all your new configurations will be lost
and you will have to reconfigure the system.
```

```
Do you want to save your configurations to flash? (y/n) [n] :
```

CLI Method

To configure certain parameters for a specific serial port.

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure nfs_data_buffering:

```
config configure conf nfsdb <string>
```

To configure data_buffering:

```
config configure line <serial port number> databuffering
<number>
```

To configure DB_mode:

```
config configure line <serial port number> dbmode <string>
```

To configure dont_show_DBmenu:

```
config configure line <serial port number> dbmenu <number>
```

To configure DB_timestamp:

```
config configure line <serial port number> dbtimestamp
<number>
```

To configure syslog_buffering:

```
config configure line <serial port number> syslogdb <number>
```

> **Tip**. You can configure all the parameters for a serial port in one line:
>
> ```
> config configure line <serial port number> tty <string>
> conf nfsdb <string> db <number> dbmode <string> dbmenu
> <number> dbtimestamp <number> syslogdb <number>
> ```

Step 2: Activate and Save.

To activate your new configurations and save them to flash, type:

```
config write
```

(This is essentially typing *signal_ras hup* and *saveconf* from the normal terminal prompt.)

DHCP

The DHCP (Dynamic Host Configuration Protocol) Client is available for firmware versions 1.2.x and above. DHCP is a protocol that allows network administrators to assign IP addresses automatically to network devices. Without DHCP (or a similar protocol like BOOTP), each device would have to be manually configured. DHCP automatically sends a new IP address to a connected device when it is moved to another location on the network. DHCP uses the concept of a fixed time period during which the assigned IP address is valid for the device it was assigned for. This "lease" time can vary for each device. A short lease time can be used when there are more devices than available IP numbers. For more information, see RFC 2131.

Parameter Involved and Passed Values

The DHCP client on the Ethernet Interface can be configured in two different ways, depending on the action the Secure Console Port Server SSH should take in case the DHCP Server does not answer the IP address request:

1. No action is taken and no IP address is assigned to the Ethernet Interface (most common configuration):
    - Set the global parameter conf.dhcp_client to 1.

- Comment all other parameters related to the Ethernet Interface (conf.eth_ip, etc.).
- Add the necessary options to the file /etc/network/dhcpcd_cmd (some options are described below).

2. The Secure Console Port Server SSH restores the last IP address previously provided in another boot and assigns this IP address to the Ethernet Interface. For the very first time the unit is powered ON, the IP address restored is 192.168.160.10 in case of failure in the DHCP. The unit goes out from the factory with DHCP enabled (conf.dhcp_client 2):

    - Set the global parameter conf.dhcp_client to 2.
    - Comment all other parameters related to the Ethernet Interface (conf.eth_ip, etc.).
    - Add the following lines to the file /etc/config_files:

      ```
      /etc/network/dhcpcd_cmd
      ```

      (from factory file already present in /etc/config_files)

      ```
      /etc/dhcpcd-eth0.save
      ```

      (From the factory, the file is already present in /etc/config_files.)

    - Add the option "-x" to the factory default content of the file /etc/network/dhcpcd_cmd:

      ```
      /sbin/dhcpcd -l 3600 -x -c /sbin/handle_dhcp
      ```

      From the factory, /etc/network/dhcpcd_cmd already has such content.

    - Add all other necessary options to the file /etc/network/dhcpcd_cmd (some options are described below). In both cases if the IP address of the Secure Console Port Server SSH or the default gateway are changed, the Secure Console Port Server SSH will adjust the routing table accordingly.

Two files are related to DHCP:

/bin/handle_dhcp — The script which is run by the DHCP client each time an IP address negotiation takes place.

/etc/network/dhcpcd_cmd — Contains a command that activates the DHCP client (used by the cy_ras program). Its factory contents are:

```
/bin/dhcpcd -c /bin/handle_dhcp
```

The options available that can be used on this command line are:

-D This option forces dhcpcd to set the domain name of the host to the domain name parameter sent by the DHCP Server. The default option is to NOT set the domain name of the host to the domain name parameter sent by the DHCP Server.

-H This option forces dhcpcd to set the host name of the host to the hostname parameter sent by the DHCP Server. The default option is to NOT set the host name of the host to the hostname parameter sent by the DHCP Server.

-R This option prevents dhcpcd from replacing the existing /etc/resolv.conf file.

> **Note**. Do not modify the -c /bin/handle_dhcp option.

Configuration for CAS, TS, and Dial-in Access

vi Method

Steps 1 and 2 under Parameters and Passed Values should be followed. You'll need to edit /etc/portslave/pslave.conf, comment some lines, etc.

Browser Method

To configure DHCP via your Web browser:

Step 1: Point your browser to the Console Server.

In the address or location field of your browser type the Console Access Server's IP address. For example:

```
http://10.0.0.0
```

Step 2: Log in as root and type the Web root password configured by the Web server.

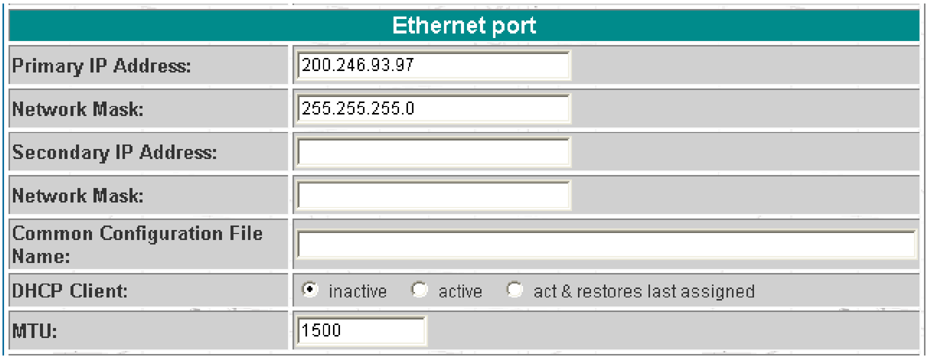
This will take you to the Configuration and Administration page.

Step 3: Click the General link on the Link Panel.

This takes you to the General page.

Step 4: Scroll down to the Ethernet port section.

You can choose the DHCP Client option in this section. Select the radio button and click the Submit button at the bottom of the page.



Ethernet port	
Primary IP Address:	200.246.93.97
Network Mask:	255.255.255.0
Secondary IP Address:	
Network Mask:	
Common Configuration File Name:	
DHCP Client:	● inactive ○ active ○ act & restores last assigned
MTU:	1500

Figure 26: DHCP client section

Step 5: Make the changes effective.

Click on the Administration > Run Configuration link, check the Serial Ports/Ethernet/Static Routes box and click on the Activate Configuration button.

Step 6: Click on the link Administration > Load/Save Configuration.

Step 7: Click the Save Configuration to Flash button.

The configuration will be saved in flash.

Chapter 3 - Additional Features

Filters

This feature is only available for firmware versions 1.2.x and above.

Description

Secure Console Port Server SSH uses the Linux utility *ipchains* to filter IP packets entering, leaving and passing through its interfaces.

An ipchains tutorial is beyond the scope of this manual. For more information on ipchains, see the ipchains man page (not included with the Secure Console Port Server SSH).

The syntax of the ipchains command is:

```
ipchains -command chain rule-specification [options]

ipchains -E old-chain-name new-chain-name
```

where:
chain is one of the following:

input	Filters for packets coming into the Secure Console Port Server SSH itself.
output	Filters for locally-generated packets.
forward	Filters for packets being routed through the Secure Console Port Server SSH.
user_created_chain	A previously defined (or in the process of being defined) chain created by the command “-N.”

command:
Only one command can be specified on the command line unless otherwise specified below. For all the long versions of the command and option names, you need to use only enough letters to ensure that ipchains can differentiate it from all other options.

Configuration for CAS, TS, and Dial-in Access

Browser Method

To configure filters in IP chains via your Web browser:

Step 1: Point your browser to the Secure Console Port Server SSH.

In the address or location field of your browser type:

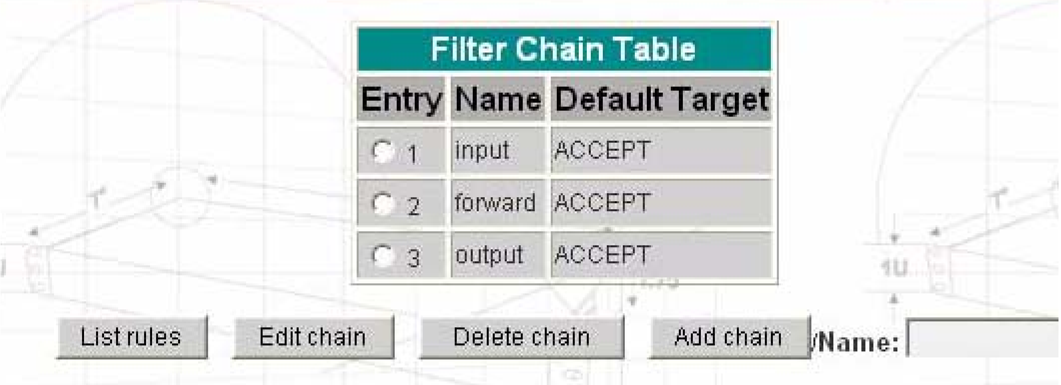
```
<Console Access Server’s IP address>
```

Step 2: Log in as root and type the Web root password configured by the Web server.

This will take you to the Configuration and Administration page.

Step 3: Click IPChains filter link.

Click on this link on the Link Panel. The following page will appear:



Filter Chain Table

Entry	Name	Default Target
○ 1	input	ACCEPT
○ 2	forward	ACCEPT
○ 3	output	ACCEPT

Figure 27: IP Chain filtering

Step 4: To create a new filter chain:

Type in the name of the filter chain in the Name box to the far right of the page, and then click the Add chain button. To enter the default target, click the appropriate Select button and then the Submit button. The new filter chain will be added to the Filter Chain Table.

Step 5: To edit or delete a filter chain:

To change the default target or to delete the filter chain, click the radio button of the filter chain and then click the Edit chain button or the Delete chain button.

Step 6: To edit the rules of the filter chain:

Click the radio button of the filter chain and then click the List rules button.
If the filter chain doesn't have rules, you need to add them. Skip to Step 9.

Step 7: To delete a rule:

Click the radio button of the rule and then click the Delete rule button.

Step 8: To edit a rule:

Click the radio button of the rule and then click the Edit rule button.

Generating Alarms

This feature helps the administrator to manage the servers. It filters the messages received by the serial port (the server's console) based on the contents of the messages. It then performs an action, such as sending an email or pager message. To configure this feature, you need to configure filters and actions in the syslog-ng.conf file. (You can read more about syslog-ng in the Syslog section.)

Port Slave Parameters Involved with Generating Alarms

conf.DB_facility This value (0-7) is the Local facility sent to the syslog-ng with data when syslog_buffering and/or alarm is active.

all.alarm When nonzero, all data received from the port is captured and sent to syslog-ng with INFO level and LOCAL[0+conf.DB_facility] facility.

Configuration for CAS, TS, and Dial-in Access

vi Method

Files to be modified:

- pslave.conf
- syslog-ng.conf

Browser Method

To configure PortSlave parameters involved with syslog-ng and the syslog-ng configuration file with your browser:

Step 1: Point your browser to the Console Server.

In the address or location field of your browser type the Console Access Server's IP address. For example:

```
http://10.0.0.0
```

Step 2: Log in as root and type the Web root password configured by the Web server.

This will take you to the Configuration and Administration page.

Step 3: Select the General link.

Click on the General link on the Link Panel to the left of the page in the Configuration section. This will take you to the General page.

Step 4: Scroll down to the Data Buffering section.

You can change the Data Buffering Facility value (conf.DB_facility). Click the Submit button.

Step 5: Select the Serial Ports link.

Click on the Serial Ports link on the Link Panel to the left of the page in the Configuration section. This will take you to the Port Selection page.

Step 6: Select port(s).

On the Port Selection page, choose all ports or an individual port to configure from the dropdown menu. Click the Submit button. This will take you to the Serial Port Configuration page.

Step 7: Scroll down to the Data Buffering section.

You can change the “Alarm for Data Buffering” (.alarm) value. Click the Submit button.

Step 8: Select the Syslog link.

Click on the Syslog link on the Link Panel to the left of the page in the Configuration section. This will take you to the Edit the Syslog-ng Configuration File page.

Step 9: Make the changes effective.

Click on the Administration > Run Configuration link, check the Serial Ports/ Ethernet/Static Routes box and click on the Activate Configuration button.

Step 10: Click on the link Administration > Load/Save Configuration.

Step 11: Click the Save Configuration to Flash button.

The configuration was saved in flash.

Wizard Method

The Alarm Generation custom wizard configures the ALL.ALARM parameter.

Step 1: Bring up the wizard.

At the command prompt, type the following to bring up the Alarm Generation custom wizard:

```
wiz --al
```

Screen 1 (below) will appear.

Screen 1:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

ALL.ALARM - When non zero, all data received from the port
are captured and sent to syslog-ng with INFO level and
LOCAL[0+conf.DB_facility] facility. The syslog-ng.conf
file should be set accordingly, for the syslog-ng to take
some action.


(Please see the 'Syslog-ng Configuration to use with Alarm
Feature' section under Generating Alarms in Chapter 3 of
the system's manual for the syslog-ng configuration file.)

all.alarm[0] :
```

Screen 2:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Current configuration:
(The ones with the '#' means it's not activated.)

all.alarm : 0


Set to defaults? (y/n) [n] :
```

Screen 3:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


ALL.ALARM - When non zero, all data received from the port
are captured and sent to syslog-ng with DAEMON facility
and ALERT level. The syslog-ng.conf file should be set
accordingly, for the syslog-ng to take some action.
(Please see the 'Syslog-ng Configuration to use with Alarm
Feature' section under Generating Alarms in Chapter 3 of
the system's manual for the syslog-ng configuration file.)

all.alarm[0] :
```

Note: conf.DB_facility is configured under the syslog parameters (wiz - - sl).

Screen 4:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
Current configuration:
(The ones with the '#' means it's not activated.)

all.alarm : 0

Are these configuration(s) all correct? (y/n) [n] :
```

If you type 'n'

```
Type 'c' to go back and CORRECT these parameters or 'q' to
QUIT :
```

Typing 'c' repeats the application, typing 'q' exits the entire wiz application

If you type 'y'

```
Discard previous port-specific parameters? (y/n) [n] :
```

Note: Answering yes to this question will discard only the parameter(s) which you are currently configuring if they were configured for a specific port in a previous session. For instance, if you are currently configuring parameter, all.x, and there was a specific port, s2.x, configured; then, answering yes to this question will discard s2.x.

```
Type 'c' to CONTINUE to set these parameters for specific
ports or 'q' to QUIT :
```

Typing 'c' leads to Screen 5, typing 'q' leads to Screen 6.

Screen 5:

```
************************************************************
********* C O N F I G U R A T I O N    W I Z A R D *********
************************************************************


You have 8 available ports on this system.


Type 'q' to quit, a valid port number[1-8], or anything
else to refresh :
```

Note: The number of available ports depends on the system you are on. Typing in a valid port number repeats this program except this time it's configuring for the port number you have chosen. Typing 'q' leads to Screen 6.

Screen 6:

```
************************************************************
********* C O N F I G U R A T I O N    W I Z A R D *********
************************************************************

(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)


Do you want to activate your configurations now? (y/n) [y] :
```

Screen 7:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Flash refers to a type of memory that can be erased and
reprogrammed in units of memory known as blocks rather than
one byte at a time; thus, making updating to memory easier.

If you choose to save to flash, your configurations thus
far will still be in the memory of the system even after you
reboot it. If you don't save to flash and if you were to
reboot the system, all your new configurations will be lost
and you will have to reconfigure the system.

Do you want to save your configurations to flash? (y/n) [n] :
```

CLI Method

To configure certain parameters for a specific serial port:

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure conf.DB_facility:

```
config configure conf dbfacility <number>
```

To configure alarm:

```
config configure line <serial port number> alarm <number>
```

Tip. You can configure all the parameters for a serial port in one line.

```
config configure line <serial port number> tty <string>
alarm <number>
```

Step 2: Activate and Save.

To activate your new configurations and save them to flash, type:

```
config write
```

(This is essentially typing *signal_ras hup* and *saveconf* from the normal terminal prompt.)

Syslog-ng Configuration to use with Alarm Feature

This configuration example is used for the alarm feature.

Step 1: Configure the pslave.conf file parameter.

In the pslave.conf file the parameters of the alarm feature are configured as:

```
all.alarm 1

conf.DB_facility  2
```

Step 2: Add lines to syslog-ng.conf.

The syslog-ng.conf file needs these lines:

```
# local syslog clients

source sysl { unix-stream("/dev/log"); };

# To filter ALARM message with the string "kernel panic" :

filter f_kpanic {facility(local2) and level(info) and
match("ALARM") and match("kernel panic"); };

# To filter ALARM message with the string "root login" :

filter f_root { facility(local2) and level(info) and
match("ALARM") and match("root login"); };

# To send e-mail to z@none.com (SMTP's IP address 10.0.0.2)

# from the e-mail address a@none.com with subject "ALARM".

# The message will carry the current date, the hostname
# of this unit and the message that was received from the
source.

destination d_mail1 {

    pipe("/dev/cyc_alarm"

    template("sendmail -t z@none.com -f a@none.com -s
\"ALARM\"  -m \"$FULLDATE $HOST $MSG\" -h 10.0.0.2"));

};

# Example to send a pager to phone number 123 (Pager server
at 10.0.0.1) with message

# carrying the current date, the hostname of this Secure Con-
sole Port Server SSH and the message that was received from
the source :

destination d_pager {

pipe("/dev/cyc_alarm"

template("sendsms -d 123 -m \"$FULLDATE $HOST $MSG\"
10.0.0.1"););
};

# Example to send a Link Down trap to server at 10.0.0.1 with
message carrying the current

# date, the hostname of this unit and the message that
received from the source :

destination d_trap {

pipe("/dev/cyc_alarm"

template("snmptrap -v1 10.0.0.1 public \"\" \"\" 2 0 \"\" \

.1.3.6.1.2.1.2.2.1.2.1 s \"$FULLDATE $HOST $MSG\" "););

};

# To send e-mail and snmptrap if message received from local
syslog client has the string "kernel panic" :
```

```
log { source(sysl); filter(f_kpanic); destination(d_mail1);
destination(d_trap); };

# To send e-mail and pager if message received from local
syslog client has the string

# "root login":

log { source(sysl); filter(f_root); destination(d_mail1);
destination(d_pager); };
```

Alarm, Sendmail, Sendsms and Snmptrap

Alarm

This feature is available only for the Console Server Application. The Secure Console Port Server SSH sends messages using pager, e-mail, or snmptrap if the serial port receives messages with specific string. To configure this feature:

Step 1: Activate alarm in Portslave configuration file.

Parameter all.alarm - 0 inactive or <> 0 active.

Step 2: Configure filters in the syslog-ng configuration file.

```
filter f_alarm { facility(local[0+conf.DB_facility]) and
level(info) and match("ALARM") and match("<your string>"); }
;
```

Example: to filter the ALARM message with the string “kernel panic"
(conf.DB_facility is configured with value 1):

```
filter f_kpanic {facility(local1) and level(info) and
match("ALARM") and match ("kernel panic"); };
```

Example: to filter the ALARM message with the string “root login" :

```
filter f_root { facility(local1) and level(info) and
match("ALARM") and match("root login"); };
```

Step 3: Configure actions in the syslog-ng configuration file.

(See more details in syslog-ng examples.)

Example: alarm is active and if the serial port receives the string “kernel panic," one message will be sent to the pager.

```
log (source(sysl); filter(f_kpanic); destination(d_pager);
};
```

To send e-mail:

```
destination d_mail { pipe("/dev/cyc_alarm" template("send-
mail <pars>"));};
```

To send a pager message:

```
destination d_pager {pipe("/dev/cyc_alarm" template("sendsms
<pars>"));};
```

To send snmptrap:

```
destination d_trap {pipe("/dev/cyc_alarm" template("snmptrap
<pars>")); };
```

Step 4: Connect filters and actions in the syslog-ng configuration file.

Example: alarm is active and if the serial port receives the string “kernel panic," one message will be sent to the pager.

```
log (source(sysl); filter(f_kpanic); destination(d_trap);
destination(d_pager); };
```

Sendmail

Sendmail sends a message to a SMTP server. It is not intended as a user interface routine; it is used only to send pre-formatted messages. Sendmail reads all parameters in the command line. If the SMTP server does not answer the SMTP protocol requests sent by sendmail, the message is dropped.

Synopsis:

```
sendmail -t <name>[,<name>] [-c <name> [,<name>]] [-b <name>
[,<name>]] [-r <name>] -f <name> -s <text> -m <text> -h <SMTP
server> [-p <smtp-port>]
```

```
where:
```

-t <name>[,<name>]	“To: ” Required. Multi-part allowed (multiple names are separated by commas). Names are expanded as explained below.
[-c <name> [,<name>]]	“Cc: ” Optional. Multi-part allowed (multiple names are separated by commas).
[-b <name> [,<name>]]	“Bcc: ” Optional. Multi-part allowed (multiple names are separated by commas).
[-r <name>]	“Reply-To: ” Optional. Use the Reply-To: field to make sure the destination user can send a reply to a regular mailbox.
-f <name>	“From: ” Required.
-s <text>	"Subject: " Required.
-m <text>	"body" The message body.
-h <SMTP server>	Required. IP address or name of the SMTP server.
[-p <SMTP port>	Optional. The port number used in the connection with the server. Default: 25.
<name>	Any email address.
<text>	A text field. As this kind of field can contain blank spaces, please use the quotation marks to enclose the text.

For example, to send e-mail to z@none.com (SMTP's IP address 10.0.0.2) from the e-mail address a@none.com with subject “sendmail test.”

```
sendmail -t z@none.com -f a@none.com -s "sendmail test" -m "Send-
mail test. \n Is it OK??? " -h 10.0.0.2
```

Sendsms

The sendsms is the Linux command line client for the SMSLink project. It accepts command line parameters that define the message to be sent, and transmits them to the SMS server process running on the designated server. The sendsms was developed specifically for easy calling from shell scripts or similar situations.

Synopsis:

```
sendsms [-r] [-g] [-v] -d dest (-m message or -f msgfile)

[-u user] [-p port] server
```

where:

-r Reporting. Additional info will be included in the message printed on stderr (namely, the device name used by the server to send the SMS out, and the message ID attributed to the SMS by the module's SIM card). If any of these items is missing or can't be parsed, a value of “??” will be returned.

-g Turns debugging on. Will output the entire dialog with the server on stderr (and more).

-h Displays a short help message and exits.

-v Displays version information and exits.

-d dest Required. The GSM network address (i.e. phone number) of the mobile phone the message is to be sent to. Supported format is: [int. prefix - country code] area code - phone number. The international prefix can be either “+” or “00” (or any other value supported by the GSM network provider the server is subscribed to). Some separation characters can be used to beautify the number, but they are purely cosmetic and will be stripped by the server. Those characters are [./-]. The pause character (',') is not supported. Regarding the international country code, don't forget that its necessity is to be considered respective to the SMS gateway location (the host this client program is connecting to), not the location where the client is run from.

-d dest (cont.)

If there are any doubts, please contact the SMS server administrator for your network. Please always include the area code (even when sending to a destination in the same “area”, i.e., on the same network). The number without the area code, though syntactically correct and accepted by the network, may never get delivered.

-m message

Required (Use one and only one of “-m” or “-f”). The text of the message to be sent. Unless made up of a single word, it will have to be quoted for obvious reasons. Maximum length is 160 characters. A longer message will be truncated (you will be warned about it), but the message will still be sent. At the present time, only 7-bit ASCII is supported for the message text.

-f msgfile

Required (use one and only one of “-m” or “-f”). The name of a text file where the message to send is to be read from. This file can contain multiple lines of text (they will be concatenated), but its total length can't exceed 160 characters. A longer text will be truncated (you will be warned about it), but the message will still be sent. The special file '-' means that input will be read from stdin. At the present time, only 7-bit ASCII is supported for the message text.

-u user

Optional. The server module requires the user to identify her/ himself for logging purposes. No authentication is performed on this information, however. If this parameter is omitted, sendsms will send the UNIX username of the current user. This parameter allows you to override this default behavior (might be useful in the case of automated sending).

-p port

Optional. Communication port on the target server. If provided here, this value will be used to connect to the server. If omitted, the client will query the local system for the port number associated with the “well known service” sms (as defined in /etc/services). If that doesn't return an answer, the compiled-in default value 6701 will be used.

server Required. The host name or IP address of the computer where the SMS gateway server process is running. By default, this server will be listening on TCP port 6701.

Upon success (when the server module reports that the message was successfully sent), sendsms returns 0. When a problem occurs, a non zero value is returned. Different return values indicate different problems. A return value of 1 indicates a general failure of the client program.

COPYRIGHT: SMSLink is (c) Les Ateliers du Heron, 1998 by Philippe Andersson.

Example to send a pager message to phone number 123 (Pager server at 10.0.0.1) with message:

```
sendsms -d 123 -m "Hi. This is a test message send from Secure Console Port Server SSH using sendsms" 10.0.0.1
```

Snmptrap

Snmptrap is an SNMP application that uses the TRAP-PDU Request to send information to a network manager. One or more fully qualified object identifiers can be given as arguments on the command line. A type and a value must accompany each object identifier. Each variable name is given in the format specified. If any of the required version 1 parameters—enterprise-oid, agent and uptime—are specified as empty, it defaults to “.1.3.6.1.4.1.3.1.1”, hostname, and host-uptime respectively.

Synopsis

```
snmptrap -v 1 [-Ci] [common arguments] enterprise-oid agent
generic-trap specific-trap uptime [objectID type value]...
```

```
snmptrap -v [2c|3] [-Ci] [common arguments] uptime trap-oid
[objectID type value]...
```

where:

-Ci Optional. It sends INFORM-PDU.

common arguments	Required. They are: "-c <community name> <SNMP server IP address>"
enterprise-oid	Required, but it can be empty ('').
agent	Required, but it can be empty (''). The agent name.
generic-trap	The generic trap number: 2 (link down), 3 (link up), 4 (authentication failure), ...
specific-trap	Required. The specific trap number.
uptime	Required.
[objectID type value]	Optional. objectID is the object oid. You want to inform its value to server.

If the network entity has an error processing the request packet, an error packet will be returned and a message will be shown, helping to pinpoint in what way the request was malformed. If there were other variables in the request, the request will be resent without the bad variable.

For example, to send a Link Down trap to server at 10.0.0.1 with interfaces.iftable.ifentry.ifde-scr:

```
snmptrap -v 1 10.0.0.1 public "" 2 0 "" .1.3.6.1.2.1.2.2.1.2.1 s
"Secure Console Port Server SSH: serial port number 1 is down"
```

-Ci	Optional. It sends INFORM-PDU.
common arguments	Required. They are: SNMP server IP address and community.
enterprise-oid	Required, but it can be empty ('').

Help

Help Wizard Information

```
Synopsis: wiz [--OPTIONS] [--port <port number>]
```

Note: To directly configure a feature for a specific serial port, use the ”- -port <port number>” option after “wiz - -[option].”

Note: Make sure there are two hyphens before any of the options listed on the following table.

Table 10: General Options for the Help Wizard

Option	Description
ac <cas or ts>	Configuration of access method parameters
al	Configuration of alarm parameter
all <cas or ts>	Configuration of all parameters
auth	Configuration of authentication parameters
db	Configuration of data buffering parameters
help	Print this help message
pm	Configuration of power management parameters.

Table 10: General Options for the Help Wizard

Option	Description
sl	Configuration of syslog parameters
snf	Configuration of sniffing parameters
sset <cas or ts>	Configuration of serial setting parameters
tl	Configuration of terminal login display parameters
tso	Configuration of other parameters specific to the TS profile

Step 1: Bring up the wizard.

At the command prompt, type the following to bring up the Help custom wizard (you can also type wiz -h):

```
wiz --help
```

Help Command Line Interface Information

Note: To enter into CLI mode, type config at the terminal prompt. You will then get a CLI prompt similar to config@hostname>>. Once in CLI mode, you eliminate the need to type config at the beginning of your CLI commands. To exit from this mode, type exit or quit.

Synopsis 1 - Configuration of Port Specific Parameters

```
config configure line [serial port number] [options]
```

or in CLI mode:

```
configure line [serial port number] [options]
```

The following table shows Help CLI Options and the actual parameter modified for Synopsis 1.

Table 11: Help CLI Options - Synopsis 1

Option	Actual Parameter Modified
accthost1 <string>	accthost1
accthost2 <string>	accthost2
adminusers <string>	admin_users
alarm <number>	alarm
authhost1 <string>	authhost1
authhost2 <string>	authhost2
authtype <string>	authtype
auto_input <string>	auto_answer_input
auto_output <string>	auto_answer_output
break <string>	break_sequence
datasize <number>	datasize
databuffering <number>	data_buffering
dbmenu <number>	dont_show_DBmenu
dbmode <string>	DB_mode
dbtimestamp <number>	DB_timestamp
dcd <number>	dcd
dtr_reset <number>	DTR_reset
escape <string>	escape_char
flow <string>	flow
host <string>	host
idletimeout <number>	idletimeout

Table 11: Help CLI Options - Synopsis 1

Option	Actual Parameter Modified
ipno <string>	ipno
issue <string>	issue
lf <number>	lf_suppress
modbus <string>	modbus_smode
multiplesess <string>	multiple_sessions
parity <string>	parity
pmkey <string>	pmkey
pmnumofoutlets <number>	pmNumOfOutlets
pmoutlet <string>	pmoutlet
pmtype <string>	pmtype
pmusers <string>	pmusers
pollinterval <number>	poll_interval
prompt <string>	prompt
protocol <string>	protocol
retries <number>	timeout
secret <string>	secret
sniffmode <string>	sniff_mode
socket <number>	socket_port
speed <number>	speed
stopbits <number>	stopbits
sttycmd <string>	sttyCmd
syslogdb <number>	syslog_buffering

Table 11: Help CLI Options - Synopsis 1

Option	Actual Parameter Modified
syslogsess <number>	syslog_sess
telnetclientmode <number>	telnet_client_mode
term <string>	term
timeout <number>	timeout
tty <string>	tty
txinterval <number>	tx_interval
userauto <string>	userauto
users <string>	users

(Refer to Appendix C for more info on the parameters.)

Synopsis 2 - Configuration of Network-related Parameters

```
config configure ether [options]
```

or in CLI mode:

```
configure ether [options]
```

Table 12: Help CLI Options - Synopsis 2

Option	Description	Actual Parameters Modified
ip <string>	Configuration of the IP of the Ethernet interface.	*conf.eth_ip*
mask <string>	Configuration of the mask for the Ethernet network.	*conf.eth_mask*
mtu <number>	Configuration of the Maximum Transmission Unit size.	*conf.eth_mtu*

(Refer to Appendix C for more info on the parameters.)

Synopsis 3 - Configuration of other Conf. Parameters

```
config configure conf [options]
```

or in CLI mode:

```
configure conf [options]
```

Table 13: Help CLI Options - Synopsis 3

Option	Actual Parameter Modified
dbfacility <number>	conf.DB_facility
facility <number>	conf.facility
group <string>	conf.group
locallogins <number>	conf.locallogins
nfsdb <string>	conf.nfs_data_buffering

(Refer to Appendix C for more info on the parameters.)

Note: To include spaces within the string you are configuring, encapsulate the string within single or double quotes. For instance, to configure s2.sttyCmd -igncr -onlcr, type (do not put a space after a comma):

```
config configure line 2 sttycmd "-igncr -onlcr"
```

Tip. You can specify the range or list of serial ports if you wish to configure the same parameters for several ports. For instance, to configure parameters for ports 2 through 4, you can type this command: config configure line 2-4 [options]. Or to configure parameters for just ports 4, 6, and 9, you can type:

```
config configure line 4,6,9 [options]
```

(Do not put a space after the commas when listing the serial ports.)

Requesting Help for the CLI

There are two methods for requesting help for the CLI:

- To obtain general help on the format of CLI, type *config help | more* at the terminal prompt.
- Help may be requested at any point in a command by entering a “?.” If nothing matches, the help list will be empty and you must backup until entering a “?” shows the available options.

For example:

- To find out possible commands that can come after *config*, type:

```
config ?
```

- To find out what parameters are configurable through CLI, type:

```
config configure line <serial port number> ?
```

Chapter 3 - Additional Features

Modbus

MODBUS is an application layer messaging protocol for client/server communication which is widely used in the industrial automation. It is a confirmed service protocol and offers many services specified by function codes, like reading and writing registers on PLCs.

A protocol converter for the MODBUS protocol over the TCP/IP communication stack (Modbus/TCP) is implemented in the Secure Console Port Server SSH and converts Modbus/TCP ADUs from the Ethernet interface to plain MODBUS message frames over a serial RS-232 or RS-485 interface, and vice versa, supporting both serial modes (ASCII and RTU).

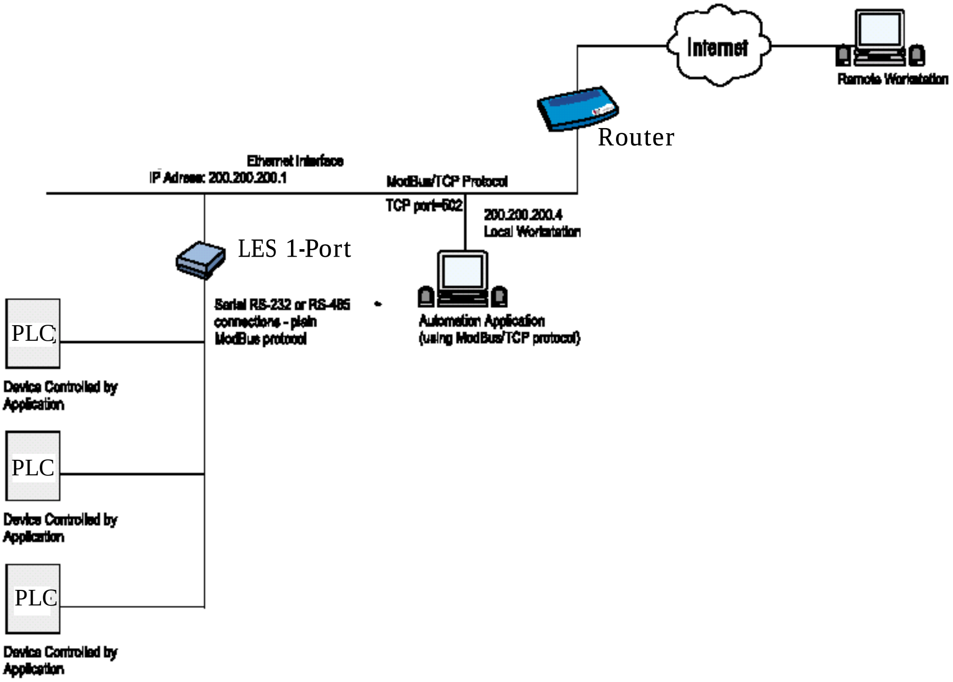


Figure 28: Modbus application

In this example, the Automation Application running in the Workstation (local or remote) controls the PLCs connected to the serial port (RS-485) of the Secure Console Port Server SSH 1-Port using MODBUS/TCP protocol. The connection is opened using the Secure Console Port Server SSH 1-Port Ethernet IP address and TCP port = 502. The Secure Console Port Server SSH 1-Port accepts the incoming connection and converts MODBUS/ TCP ADUs (packets) to plain MODBUS frames and sends them over the serial port. On the other hand, the MODBUS frames received from the serial port are converted to MODBUS/ TCP ADUs and sent through the TCP connection to the Automation Application.

The configuration described earlier for Console Access Servers (see Figure 1: Console Access Server diagram) should be followed with the following exceptions for this example:

Table 14: Modbus pslave.conf port-specific parameters (only where they differ from the standard CAS profile)Modbus parameters

Parameter	Description	Value for this Example
all.protocol	For the console server profile, the possible protocols are socket_server (when telnet is used), socket_ssh (when ssh version one or two is used), raw_data (to exchange data in transparent mode – similar to socket_server mode, but without telnet negotiation, breaks to serial ports, etc.), or *modbus* (an application layer messaging protocol for client/server communication widely used for industrial automation).	modbus
all.modbus_ smode	Communication mode through the serial ports. This parameter is meaningful only when modbus protocol is configured. The valid options are ascii (normal TX/ RX mode) and rtu (some time constraints are observed between characters while transmitting a frame). If not configured, ASCII mode will be assumed.	ascii

Chapter 3 - Additional Features

NTP

The ntpclient is a *Network Time Protocol* (RFC-1305) client for UNIX- and Linux-based computers. In order for the Secure Console Port Server SSH to work as a NTP client, the IP address of the NTP server must be set in the file /etc/ntpclient.conf.

The script shell /bin/ntpclient.sh reads the configuration file (/etc/ntpclient.conf) and build the line command to call /bin/ntpclient program.

Parameters Involved and Passed Values

The file /etc/ntpclient.conf has the value of two parameters:

NTPSERVER	The IP address of the NTP server.
INTERVAL	Check time every interval seconds (default 300).

The data and time will be update from the NPT server according to the parameter options. The ntpclient program has this syntax:

```
ntpclient [options]
```

Options:

-c count	Stop after count time measurements (default 0 means go forever).
-d	Print diagnostics.
-h hostname	NTP server host (mandatory).
-i interval	Check time every interval seconds.
-l	Attempt to lock local clock to server using adjtimex(2).
-p port	Local NTP client UDP port.
-r	Replay analysis code based on stdin.
-s	Clock set (if count is not defined this sets count to 1).

Configuration for CAS, TS, and Dial-in Access

vi Method

Files to be changed:

```
/etc/ntpclient.conf
```

Browser Method

To configure NTP with your browser:

Step 1: Point your browser to the Console Server.

In the address or location field of your browser type the Console Access Server's IP address. For example:

```
http://10.0.0.0
```

Step 2: Log in as root and type the Web root password configured by the Web server.

This will take you to the Configuration and Administration page.

Step 3: Click on the Edit Text File link.

Click on this link on the Link Panel or on the Configuration section of the Configuration and Administration page. (See .) You can then pull up the appropriate file and edit it.

Step 4: in /etc/mgetty/login.configin /etc/ppp/pap-secretsCreate the user for login in the Radius server.

If the login option was used, create the user either locally (by running adduser) or create the user in the Radius server for Radius authentication. When the login option is used, /etc/pam.conf may also need to be changed. (By default, /etc/pam.conf has the ppp and login services configured for local authentication. You will have to change them if you want Radius authentication. More information can be found in "Appendix D - Linux-PAM".)

Step 5: as /etc/ppp/options.ttyS33 (the modem port)s in /etc/ppp/options.ttyS33Save /etc/ppp/options.ttyS33 in flash.

Chapter 3 - Additional Features

Ports Configured as Terminal Servers

Important! 1-Port owners: please skip to the special section on the 1-Port later in the installation chapter called Configuring the 1-Port for the first time, then perform "Task 5: Activate the changes" on page 64 through Task 8: Reboot the Secure Console Port Server SSH listed in Chapter 2 - Installation and Configuration to finish the configuration. Make into links.

There are TS-specific parameters that are required to be configured when using the serial ports with the TS profile. The configuration of these TS-specific parameters are described in this section. Additional configuration for TS is described in Access Method and Serial Settings in Chapter 3, and in Appendix C – The pslave Configuration File.

TS Setup Wizard

The Wizard can be used to configure TS-specific parameters. (TSO stands for "TS Other"--other parameters specific to the TS profile):

Step 1: At the command line interface type the following:

```
wiz --tso
```

Screen 1:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

INSTRUCTIONS for using the Wizard:
You can:
1) Enter the appropriate information for your system
and press ENTER. Enter '#' if you want to
deactivate that parameter or
2) Press ENTER if you are satisfied with the value
within the brackets [ ] and want to go on to the
```

```
next parameter or
3) Press ESC if you want to exit.

NOTE: For some parameters, if there is nothing within
the brackets, it will continue to ask for a value.
In that case, you must enter a valid value or # if you
do not wish to configure the value.


Press ENTER to continue...
```

Screen 2:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Current configuration:
(The ones with the '#' means it's not activated.)

all.host : 192.168.160.8
all.term : vt100
conf.locallogins : 0


Set to defaults? (y/n) [n] :
```

Screen 3:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
ALL.HOST - The IP address of the host to which the
terminals will connect.

all.host[192.168.160.8] :

ALL.TERM - This parameter defines the terminal type assumed
when performing rlogin or telnet to other hosts.

all.term[vt100] :
```

Screen 4:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
CONF.LOCALLOGINS - This parameter is only necessary when
authentication is being performed for a port. When set to
1, it is possible to log into the system directly by
placing a '!' before users' login name, then using their
normal password. This is useful if the Radius authentica-
tion server is down.

conf.locallogins[0] :
```

Screen 5:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
Current configuration:
(The ones with the '#' means it's not activated.)

all.host : 192.168.160.8
all.term : vt100
conf.locallogins : 0


Are these configuration(s) all correct? (y/n) [n] :
```

If you type 'n'

```
Type 'c' to go back and CORRECT these parameters
or 'q' to QUIT :
```

Typing 'c' repeats the application, typing 'q' exits the entire wiz application

If you type 'y'

```
Discard previous port-specific parameters? (y/n) [n] :
```

```
Type 'c' to CONTINUE to set these parameters for
specific ports or 'q' to QUIT :
```

Typing 'c' leads to Screen 6, typing 'q' leads to Screen 7.

Screen 6:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


You have 8 available ports on this system.

Type 'q' to quit, a valid port number[1-8], or anything
else to refresh :
```

Tip. The number of available ports depends on the system you are on. Typing in a valid port number repeats this program except this time it's configuring for the port number you have chosen. Typing 'q' leads to Screen 7.

Screen 7:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************

(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)

Do you want to activate your configurations now? (y/n) [n]:
```

Screen 8:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Flash refers to a type of memory that can be erased and
reprogrammed in units of memory known as blocks rather than
one byte at a time; thus, making updating to memory easier.

If you choose to save to flash, your configurations thus
far will still be in the memory of the system even after you
reboot it. If you don't save to flash and if you were to
reboot the system, all your new configurations will be lost
and you will have to reconfigure the system.


Do you want to save your configurations to flash? (y/n) [n] :
```

CLI Method

To configure certain parameters for a specific serial port:

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure host:

```
config configure line <serial port number> host <string>
```

To configure term:

```
config configure line <serial port number> term <string>
```

To configure conf.locallogins:

```
config configure conf locallogins <number>
```

Step 2: Activate and Save.

Tip. You can configure all the parameters for a serial port in one line.

```
config configure line <serial port number> tty <string>
host <string> term <string> locallogins <number>
```

To activate your new configurations and save them to flash, type:

```
config write
```

(This is essentially typing *signal_ras hup* and *saveconf* from the normal terminal prompt.)

Chapter 3 - Additional Features

Serial Settings

This feature controls the speed, data size, parity, and stop bits of all ports. It also sets the flow control to hardware, software, or none; the DCD signal; and tty settings after a socket connection to that serial port is established.

Parameters Involved and Passed Values

Terminal Settings involve the following parameters (the first four are physical parameters):

all.speed

The speed for all ports. Default value: *9600*.

all.datasize

The data size for all ports. Default value: *8*.

all.stopbits

The number of stop bits for all ports. Default value: *1*.

all.parity

The parity for all ports. Default value: *none*.

all.flow

This sets the flow control to hardware, software, or none. Default value: *none*.

all.dcd

DCD signal (sets the tty parameter CLOCAL). Valid values are 0 or 1. If all.dcd=0, a connection request will be accepted regardless of the DCD signal and the connection will not be closed if the DCD signal is set to DOWN. If all.dcd=1 a connection request will be accepted only if the DCD signal is UP and the connection will be closed if the DCD signal is set to DOWN. Default value: *0*.

all.sttyCmd (for CAS only)

The TTY is programmed to work as configured and this user-specific configuration is applied over that serial port. Parameters must be separated by a space. The following example sets :
-*igncr*
This tells the terminal not to ignore the carriage-return on input,
-*onlcr*
Do not map newline character to a carriage return or newline character sequence on output,
opost
Post-process output,
-*icrnl*
Do not map carriage-return to a newline character on input.

```
all.sttyCmd -igncr -onlcr opost -icrnl
```

DTR_reset (for CAS only)

This parameter specifies the behavior of the DTR signal in the serial port configured with buffering or sniff session. If set to zero the DTR signal will be ON if there is a connection to the serial port, otherwise OFF. If set from 1 to 99 the DTR signal will be always ON. A value greater or equal 100 specifies for how long (in milliseconds) the DTR signal will be turned off before it is turned back on again when a connection to the serial port is closed. Example value: 3.

Configuration for CAS

Browser Method

Step 1: Point your browser to the Console Server.

In the address or location field of your browser type the Console Access Server's IP address. For example:

```
http://10.0.0.0
```

Step 2: Log in as root and type the Web root password configured by the Web server.

This will take you to the Configuration and Administration page.

Step 3: Select the Serial Ports link.

Click on the Serial Ports link on the Link Panel to the left of the page or in the Configuration section of the page. This will take you to the Port Selection page.

Step 4: Select port(s).

On the Port Selection page, choose all ports or an individual port to configure, from the dropdown menu. Click the Submit button. This will take you to the Serial Port Configuration page.

Step 5: Click the “CAS Profile” button.

Step 6: Scroll down to the Physical section.

You can change the settings for Speed, Data Size, Stop Bit, Parity, Flow Control, and DCD-sensitivity here.

Step 7: Click on the Submit button.

Step 8: Make the changes effective.

Click on the Administration > Run Configuration link, check the Serial Ports/ Ethernet/Static Routes box and click on the Activate Configuration button.

Step 9: Click on the link Administration > Load/Save Configuration.

Step 10: Click the Save Configuration to Flash button.

The configuration was saved in flash.

Wizard Method

Step 1: Bring up the wizard.

At the command prompt, type the following to bring up the CAS Terminal Settings custom wizard:

```
wiz --sset cas
```

Screen 1 will appear.

Screen 1:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

INSTRUCTIONS for using the Wizard:
You can:
1) Enter the appropriate information for your system
and press ENTER. Enter '#' if you want to
deactivate that parameter or
2) Press ENTER if you are satisfied with the value
within the brackets [ ] and want to go on to the
next parameter or
3) Press ESC if you want to exit.

NOTE: For some parameters, if there is nothing within
the brackets, it will continue to ask for a value.
In that case, you must enter a valid value or # if you
do not wish to configure the value.


Press ENTER to continue...
```

Screen 2:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Current configuration:
(The ones with the '#' means it's not activated.)

all.speed : 9600
all.datasize : 8
all.stopbits : 1
all.parity : none
all.flow : none
all.dcd : 0
all.DTR_reset : 100
all.sttyCmd : #
Set to defaults? (y/n) [n] :
```

Screen 3:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
ALL.SPEED - The data speed in bits per second (bps) of
all ports.

all.speed[9600] :

ALL.DATASIZE - The data size in bits per character of
all ports.

all.datasize[8] :
```

Screen 4:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


ALL.STOPBITS - The number of stop bits for all ports.

all.stopbits[1] :


ALL.PARITY - The parity for all ports.
(e.g. none, odd, even)

all.parity[none] :
```

Screen 5:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


ALL.FLOW - This sets the flow control to hardware,
software, or none. (e.g. hard, soft, none)

all.flow[none] :


ALL.DCD - DCD signal (sets the tty parameter CLOCAL).
Valid values are 0 or 1. In a socket session, if
all.dcd=0, a connection request (telnet or ssh) will be
accepted regardless of the DCD signal and the connection
will not be closed if the DCD signal is set to DOWN. In a
socket connection, if all.dcd=1 a connection request will
be accepted only if the DCD signal is UP and the connection
(telnet or ssh) will be closed if the DCD signal is set to
DOWN.

all.dcd[0] :
```

Screen 6:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


ALL.DTR_RESET - This parameter specifies the behavior of the
DTR signal in the serial port. If set to 0 the DTR signal
will be ON if there is a connection to the serial port, oth-
erwise it will be OFF. If set from 1 to 99 the DTR signal
will be always ON. A value greater or equal to 100 specifies
for how long (in milliseconds) the DTR signal will be turned
off before it is turned back on again when a connection to
the serial port is closed.

all.DTR_reset[100] :
```

```
ALL.STTYCMD - Tty settings after a socket connection to
that serial port is established. The tty is programmed to
work as a CAS profile and this user specific configuration
is applied over that serial port. Parameters must be
separated by space.(e.g. all.sttyCmd -igncr -onlcr opost -
icrnl)-igncr tells the terminal not to ignore the carriage-
return on input, -onlcr means do not map newline character to
a carriage return/newline character sequence on output,
opost represents post-process output, -icrnl means do not map
carriage-return to a newline character on input.

all.sttyCmd[#] :
```

Screen 7:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
Current configuration:
(The ones with the '#' means it's not activated.)

all.speed : 9600
all.datasize : 8
all.stopbits : 1
all.parity : none
all.flow : none
all.dcd : 0
all.DTR_reset : 100
all.sttyCmd : #


Are these configuration(s) all correct? (y/n) [n] :
```

If you type 'n'

```
Type 'c' to go back and CORRECT these parameters
or 'q' to QUIT :
```

Typing 'c' repeats the application, typing 'q' exits the entire wiz application

If you type 'y'

```
Discard previous port-specific parameters? (y/n) [n] :
```

Note: Answering yes to this question will discard only the parameter(s) which you are currently configuring if they were configured for a specific port in a previous session. For instance, if you are currently configuring parameter, all.x, and there was a specific port, s2.x, configured; then, answering yes to this question will discard s2.x.

```
Type 'c' to CONTINUE to set these parameters for
specific ports or 'q' to QUIT :
```

Typing 'c' leads to Screen 8, typing 'q' leads to Screen 9.

Screen 8:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
You have 8 available ports on this system.

Type 'q' to quit, a valid port number[1-8], or anything
else to refresh :
```

Note: The number of available ports depends on the system you are on. Typing in a valid port number repeats this program except this time it's configuring for the port number you have chosen. Typing 'q' leads to Screen 9.

Screen 9:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************

(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)

Do you want to activate your configurations now? (y/n) [n]:
```

Screen 10:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Flash refers to a type of memory that can be erased and
reprogrammed in units of memory known as blocks rather than
one byte at a time; thus, making updating to memory easier.

If you choose to save to flash, your configurations thus
far will still be in the memory of the system even after you
reboot it. If you don't save to flash and if you were to
reboot the system, all your new configurations will be lost
and you will have to reconfigure the system.

Do you want to save your configurations to flash? (y/n) [n] :
```

CLI Method

To configure line parameters for a specific serial port.

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure speed:

```
config configure line <serial port number> speed <number>
```

To configure datasize:

```
config configure line <serial port number> datasize <number>
```

To configure stopbits:

```
config configure line <serial port number> stopbits <number>
```

To configure parity:

```
config configure line <serial port number> parity <string>
```

To configure flow:

```
config configure line <serial port number> flow <string>
```

To configure dcd:

```
config configure line <serial port number> dcd <number>
```

To configure DTR_reset:

```
config configure line <serial port number> dtr_reset
<number>
```

To configure sttyCmd:

```
config configure line <serial port number> sttycmd <string>
```

Tip. You can configure all the parameters for a serial port in one line:

```
config configure line <serial port number> tty <string> speed <number> datasize <number>  stopbits <number> parity <string> flow <string> dcd <number> dtr_reset <number> sttycmd <string>
```

Step 2: Activate and Save.

To activate your new configurations and save them to flash, type:

```
config write
```

(This is essentially typing *signal_ras hup* and *saveconf* from the normal terminal prompt.)

Configuration for TS

Browser Method

See the browser method for the CAS, earlier in this section. The only difference for TS is that "TS Profile" button should be clicked in Step 5.

Wizard Method

Step 1: Bring up the wizard.

At the command prompt, type the following to bring up the TS Terminal Settings custom wizard:

```
wiz --sset ts
```

Note: Screens 1- 5 are the same as those of the previous wizard for sset cas, thus, they are omitted here. The only difference between this feature and the CAS wizard is the parameter sttyCmd and DTR_reset. In the TS configuration, neither of these parameters is requested.

Screen 6:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
Current configuration:
(The ones with the '#' means it's not activated.)

all.speed : 9600
all.datasize : 8
all.stopbits : 1
all.parity : none
all.flow : none
all.dcd : 0

Are these configuration(s) all correct? (y/n) [n] :
```

If you type 'n':

```
Type 'c' to go back and CORRECT these parameters
or 'q' to QUIT :
```

Typing 'c' repeats the application, typing 'q' exits the entire wiz application.

If you type 'y':

```
Type 'c' to CONTINUE to set these parameters for specific
ports or 'q' to QUIT :
```

Typing 'c' leads to Screen 7 typing 'q' leads to Screen 8.

Screen 7:

```
************************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
************************************************************
You have 8 available ports on this system.


Type 'q' to quit, a valid port number[1-8], or anything
else to refresh :
```

Note: The number of available ports depends on the system you are on. Typing in a valid port number repeats this program except this time it's configuring for the port number you have chosen. Typing 'q' leads to Screen 8.

Screen 8:

```
************************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
************************************************************

(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)


Do you want to activate your configurations now? (y/n) [y] :
```

Screen 9:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Flash refers to a type of memory that can be erased and
reprogrammed in units of memory known as blocks rather than
one byte at a time; thus, making updating to memory easier.

If you choose to save to flash, your configurations thus
far will still be in the memory of the system even after you
reboot it. If you don't save to flash and if you were to
reboot the system, all your new configurations will be lost
and you will have to reconfigure the system.


Do you want to save your configurations to flash? (y/n) [n] :
```

CLI Method

To configure line parameters for a specific serial port:

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure speed:

```
config configure line <serial port number> speed <number>
```

To configure datasize:

```
config configure line <serial port number> datasize <number>
```

To configure stopbits:

```
config configure line <serial port number> stopbits <number>
```

To configure parity:

```
configure line <serial port number> parity <string>
```

To configure flow:

```
config configure line <serial port number> flow <string>
```

To configure dcd:

```
config configure line <serial port number> dcd <number>
```

Tip. You can configure all the parameters for a serial port in one line:

```
config configure line <serial port number> tty <string>
speed <number> datasize <number>  stopbits <number>
parity <string> flow <string> dcd <number>
```

Step 2: Activate and Save.

To activate your new configurations and save them to flash, type:

```
config write
```

(This is essentially typing *signal_ras hup* and *saveconf* from the normal terminal prompt.)

Configuration for Dial-in Access

Browser Method

See the browser method for the CAS, earlier in this section. The only difference for Dial-in is that the “Dial-in Profile” button should be clicked in Step 5.

CLI Method

To configure line parameters for a specific serial port:

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure speed:

```
config configure line <serial port number> speed <number>
```

To configure datasize:

```
config configure line <serial port number> datasize <number>
```

To configure stopbits:

```
config configure line <serial port number> stopbits <number>
```

Chapter 3 - Additional Features

Session Sniffing

Versions 1.3.2 and earlier

The Secure Console Port Server SSH allows a maximum of two connections to each serial port, as follows:

- One common session: user can execute read and write commands to the tty port. Session can be established by a regular user or by an administrator.
- One sniffer session: user can execute only read commands, in order to monitor what is going on in the other (main) session. Session can only be established by an administrator, defined by the parameter all.admin_users or sN.admin_users in the file pslave.conf (exception: authentication none - anyone can open a sniffer).

The first connection always opens a common session. After the second connection has been established and the user is authenticated, the Secure Console Port Server SSH shows the following menu to the administrator user:

```
*
* * * ttySN is being used by (<user_name>) !!!
*
1 - Assume the main session
2 - Initiate a sniff session
3 - Quit
Enter your option:
```

If the second user is not an administrator, his connection is automatically refused. This description is valid for all of the available protocols (socket_server, socket_ssh or raw_data).

Versions 1.3.3 and later

You can open more than one common and sniff session at the same port. For this purpose, the following configuration items are available in the file pslave.conf:

- all.multiple_sessions: If it is configured as *no*, only two users can connect to the same port simultaneously. If it is configured as *yes*, more than two simultaneous users can connect to the same serial port. A “Sniffer menu” will be presented to the user and they can choose either to open a sniff session; to open a read and/or write session; to cancel a connection; or to send a message to other users connected to the same serial port. If it is configured as “RW_sessions,” only read and/or write sessions will be opened, and the sniffer menu won’t be presented. If it is configured as “sniff_session” only, a sniff session will be opened, and the sniffer menu won’t be presented. Default value: no.

- sN.multiple_sessions: Valid only for port N. If it is not defined, it will assume the value of all.multiple_sessions.

- all.escape_char: Valid for all the serial ports; this parameter will be used to present the menus below to the user. Only characters from ‘^a’ to ‘^z’ (i.e., CTRL-A to CTRL-Z) will be accepted. The default value is ‘^z’ (CTRL-Z).

- sN.escape_char: Valid only for port N; this parameter will be used to present the menus below to the user. Only characters from ‘^a’ to ‘^z’ (i.e. CTRL-A to CTRL-Z) will be accepted. If it is not defined, it will assume the value of all.escape_char.

When multiple sessions are allowed for one port, the behavior of the Secure Console Port Server SSH will be as follows:

1. The first user to connect to the port will open a common session.

2. From the second connection on, only admin users will be allowed to connect to that port. The Secure Console Port Server SSH will send the following menu to these administrators (defined by the parameter all.admin_users or sN.admin_users in the file pslave.conf):

```
*

* * * ttySN is being used by (<first_user_name>) !!!

*

1 - Initiate a regular session
```

```
2 - Initiate a sniff session

3 - Send messages to another user

4 - Kill session(s)

5 - Quit

Enter your option:
```

If the user selects *1 - Initiate a regular session*, s/he will share that serial port with the users that were previously connected. S/he will read everything that is received by the serial port, and will also be able to write to it.

If the user selects *2 - Initiate a sniff session*, s/he will start reading everything that is sent and/or received by the serial port, according to the parameter all.sniff_mode or sN.sniff_mode (that can be in, out or i/o).

When the user selects *3 - Send messages to another user*, the Secure Console Port Server SSH will send the user's messages to all the sessions, but not to the tty port. Everyone connected to that port will see all the "conversation" that's going on, as if they were physically in front of the console in the same room. These messages will be formatted as:

```
[Message from user/PID] <<message text goes here>> by the
```

To inform theSecure Console Port Server SSH that the message is to be sent to the serial port or not, the user will have to use the menu.

If the administrator chooses the option *4 - Kill session(s)*, the Secure Console Port Server SSH will show him/her a list of the pairs PID/user_name, and s/he will be able to select one session typing its PID, or "all" to kill all the sessions. If the administrator kills all the regular sessions, his session initiates as a regular session automatically.

Option 5 - Quit will close the current session and the TCP connection.

Only for the administrator users:

Typing *all.escape_char* or *sN.escape_char* from the sniff session or "send message mode" will make the Secure Console Port Server SSH show the previous menu. The first regular sessions will not be allowed to return to the menu. If you kill all regular sessions using the option 4, your session initiates as a regular session automatically.

Parameters Involved and Passed Values

Sniffing involves the following parameters:

all.admin_users This parameter determines which users can receive the sniff menu. When users want access per port to be controlled by administrators, this parameter is obligatory and authtype must not be none. User groups (defined with the parameter conf.group) can be used in combination with user names in the parameter list. Example values: peter, john, user_group.

all.sniff_mode This parameter determines what other users connected to the very same port (see parameter admin_users below) can see of the session of the first connected user (main session): *in* shows data written to the port, *out* shows data received from the port, and *i/o* shows both streams. The second and later sessions are called sniff sessions and this feature is activated whenever the protocol parameter is set to socket_ssh or socket_server. Example value: out.

all.escape_char This parameter determines which character must be typed to make the session enter *menu mode*. The possible values are <CTRL-a> to <CTRL-z>. Represent the CTRL with carat: ^. This parameter is only valid when the port protocol is socket_server or socket_ssh. Default value is ^z.

all.multiple_sessions If it is configured as *no*, only two users can connect to the same port simultaneously. If it is configured as *yes*, more than two simultaneous users can connect to the same serial port. A “Sniffer menu” will be presented to the user and they can choose either to open a sniff session; to open a read and/or write session; to cancel a connection; or to send a message to other users connected to the same serial port. If it is configured as “RW_sessions,” only read and/or write sessions will be opened, and the sniffer menu won’t be presented. If it is configured as “sniff_session” only, a sniff session will be opened, and the sniffer menu won’t be presented. Default value: no.

Chapter 3 - Additional Features

Configuration for CAS

vi Method

Only the file /etc/portslave/pslave.conf has to be changed.

Browser Method

To configure Session Sniffing with your browser:

Step 1: Point your browser to the Console Server.

In the address or location field of your browser type the Console Access Server's IP address. For example:

```
http://10.0.0.0
```

Step 2: Log in as root and type the Web root password configured by the Web server.

This will take you to the Configuration and Administration page.

Step 3: Select the Serial Ports link.

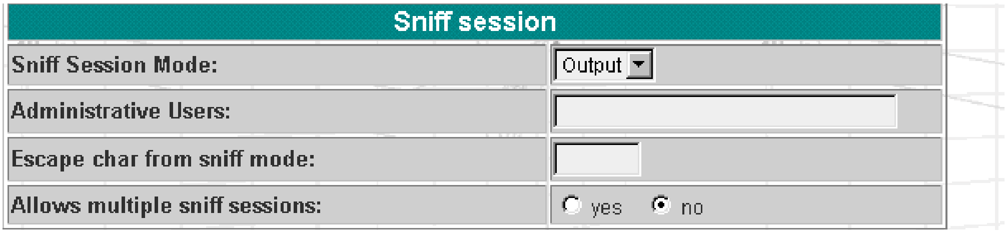
Click on the Serial Ports link on the Link Panel to the left of the page or in the Configuration section of the page. This will take you to the Port Selection page.

Step 4: Select port(s).

On the Port Selection page, choose all ports or an individual port to configure, from the dropdown menu. Click the Submit button. This will take you to the Serial Port Configuration page.

Step 5: Scroll down to the Sniff Session section.

You can configure the appropriate values here.



Sniff session

Sniff Session Mode:	Output ▾
Administrative Users:	
Escape char from sniff mode:	
Allows multiple sniff sessions:	○ yes ● no

Figure 29: Sniff Session section of the Serial Port Configuration page

Step 6: Click on the Submit button.

Step 7: Make the changes effective.

Click on the Administration > Run Configuration link, check the Serial Ports/ Ethernet/Static Routes box and click on the Activate Configuration button.

Step 8: Click on the link Administration > Load/Save Configuration.

Step 9: Click the Save Configuration to Flash button.

The configuration was saved in flash.

Wizard Method

Step 1: Bring up the wizard.

At the command prompt, type the following to bring up the Sniffing custom wizard:

```
wiz --snf
```

Screen 1:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

INSTRUCTIONS for using the Wizard:
You can:
1) Enter the appropriate information for your system
and press ENTER. Enter '#' if you want to
deactivate that parameter or
2) Press ENTER if you are satisfied with the value
within the brackets [ ] and want to go on to the
next parameter or
3) Press ESC if you want to exit.

NOTE: For some parameters, if there is nothing within
the brackets, it will continue to ask for a value.
In that case, you must enter a valid value or # if you
do not wish to configure the value.


Press ENTER to continue...
```

Screen 2:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Current configuration:
(The ones with the '#' means it's not activated.)

all.admin_users : #
all.sniff_mode : out
all.escape_char : ^z
all.multiple_sessions : no

Set to defaults? (y/n) [n] :
```

Screen 3:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


ALL.ADMIN_USERS - This parameter determines which users
can open a sniff session, which is where other users
connected to the very same port can see everything that
the first user is doing. The other users connected to the
very same port can also cancel the first user's session
(and take over). If the parameter, all.multiple_sessions,
is configured as 'no', then only two users can connect to
the same port simultaneously. If it is configured as 'yes',
more simultaneous users can sniff the session or have
read/write permissions.
(Please see details in Session Sniffing in Chapter 3 of
the system's manual.)

all.admin_users[#] :

ALL.SNIFF_MODE - This parameter determines what other
users connected to the very same port can see of the
session of the first connected user (main session). The
second session is called a sniff session and this
feature is activated whenever the protocol is set to
socket_ssh or socket_server.
(e.g. in -shows data written to the port, out -shows data
received from the port, i/o -shows both streams.)

all.sniff_mode[out] :
```

Screen 4:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

ALL.ESCAPE_CHAR - This parameter determines which
character must be typed to make the session enter into
"menu mode." The possible values are <CRTL-a> to <CRTL-z>,
and this is only valid when the port protocol is
socket_server or socket_ssh. Represent the CRTL
character with '^'. Default value is ^z.

all.escape_char[^z] :


ALL.MULTIPLE_SESSIONS - Allows users to open multiple
common and sniff sessions on the same port. The options
are “yes,” “no,” “RW_session,” or “sniff_session.”
Default is set to “no.”

all.multiple_sessions[no] :
```

Screen 5:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


Current configuration:
(The ones with the '#' means it's not activated.)

all.admin_users : #
all.sniff_mode : out
all.escape_char : ^z
all.multiple_sessions : no


Are these configuration(s) all correct? (y/n) [n] :
```

If you type 'N'

```
Type 'c' to go back and CORRECT these parameters
or 'q' to QUIT :
```

Typing 'c' repeats the application, typing 'q' exits the entire wiz application

If you type 'Y'

```
Discard previous port-specific parameters? (y/n) [n] :
```

Note: Answering yes to this question will discard only the parameter(s) which you are currently configuring if they were configured for a specific port in a previous session. For instance, if you are currently configuring parameter, all.x, and there was a specific port, s2.x, configured; then, answering yes to this question will discard s2.x.

```
Type 'c' to CONTINUE to set these parameters for
specific ports or 'q' to QUIT :
```

Typing 'c' leads to Screen 6, typing 'q' leads to Screen 7.

Screen 6:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
You have 8 available ports on this system.


Type 'q' to quit, a valid port number[1-8], or anything
else to refresh :
```

NOTE: The number of available ports depends on the system you are on. Typing in a valid port number repeats this program except this time it's configuring for the port number you have chosen. Typing 'q' leads to Screen 7.

Screen 7:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************

(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)

Do you want to activate your configurations now? (y/n) [y] :
```

Screen 8:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Flash refers to a type of memory that can be erased and
reprogrammed in units of memory known as blocks rather than
one byte at a time; thus, making updating to memory easier.

If you choose to save to flash, your configurations thus
far will still be in the memory of the system even after you
reboot it. If you don't save to flash and if you were to
reboot the system, all your new configurations will be lost
and you will have to reconfigure the system.

Do you want to save your configurations to flash? (y/n) [n] :
```

CLI Method

To configure certain parameters for a specific serial port:

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure admin_users:

```
config configure line <serial port number> adminusers
<string>
```

To configure sniff_mode:

```
config configure line <serial port number> sniffmode
<string>
```

To configure escape_char:

```
config configure line <serial port number> escape <string>
```

To configure multiple_sessions:

```
config configure line <serial port number> multiplesess
<string>
```

Tip. You can configure all the parameters for a serial port in one line.

```
config configure line <serial port number> tty <string>
adminusers <string> sniffmode <string> escape <string>
multiplesess <string>
```

Step 2: Activate and Save.

To activate your new configurations and save them to flash, type:

```
config write
```

Chapter 3 - Additional Features

SNMP

Short for Simple Network Management Protocol: a set of protocols for managing complex networks. The first versions of SNMP were developed in the early 80s. SNMP works by sending messages, called protocol data units (PDUs), to different parts of a network. SNMP-compliant devices, called agents, store data about themselves in Management Information Bases (MIBs) and return this data to the SNMP requesters.

The Secure Console Port Server SSH uses the net-snmp package (http://www.net-snmp.org).

Important! Check the SNMP configuration before gathering information about Secure Console Port Server SSH by SNMP. There are different types of attacks an unauthorized user can implement to retrieve sensitive information contained in the MIB. By default, the SNMP configuration in Secure Console Port Server SSH cannot permit the public community to read SNMP information.

The net-snmp supports snmp version 1, 2 and 3. To use SNMP version 3 (username/password), perform the following steps:

Step 1: Create a file /etc/snmp/snmpd.local.conf with the following line:

```
createUser <username> MD5 <password> DES
```

Step 2: Include the following line in /etc/snmp/snmpd.conf, if the user has permission to read only:

```
rouser <username>
```

Step 3: Include the following line in /etc/config_files:

```
/etc/snmp/snmpd.local.conf
```

You can configure the */etc/snmp/snmpd.conf* file as indicated later in this section.

1. Snmp version 1

   - RFC1155 - SMI for the official MIB tree

- RFC1213 - MIB-II

2. Snmp version 2
   - RFC2578 - Structure of Management Information Version 2 (SMIv2)
   - RFC2579 - Textual Conventions for SMIv2
   - RFC2580 - Conformance Statements for SMIv2
3. Snmp version 3
   - RFC2570 - Introduction to Version 3 of the Internet-standard Network Management Framework
   - RFC2571 - An Architecture for Describing SNMP Management Frameworks
   - RFC2572 - Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)
   - RFC2573 - SNMP Applications
   - RFC2574 - User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)
   - RFC2575 - View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)
   - RFC2576 - Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework
4. Private UCD SNMP mib extensions (enterprises.2021)
   - Information about memory utilization (/proc/meminfo)
   - Information about system status (vmstat)
   - Information about net-snmp packet
5. Private Black Box Vendor MIB (enterprises.2925)
   - Black Box LES 2800A-xx Remote Management Object Tree (blackbox.4). This MIB permits you to get informations about the product, to read/write some configuration items and to do some administration commands. (For more details see the blackbox.mib file.)

Chapter 3 - Additional Features

Configuration for CAS, TS, and Dial-in Access

vi Method

Files to be changed:

```
/etc/snmp/snmpd.conf
```

This file has information about configuring for SNMP.

Browser Method

To configure SNMP with your browser:

Step 1: Point your browser to the Console Server.

In the address or location field of your browser type the Console Access Server's IP address. For example:

```
http://10.0.0.0
```

Step 2: Log in as root and type the Web root password configured by the Web server.

This will take you to the Configuration and Administration page.

Step 3: Click on the SNMP link.

Select the SNMP link. The SNMP configuration file will appear in text mode.

Step 4: Edit the configuration file and click on the Submit button

Step 5: Make changes effective.

Click on the Administration > Run Configuration link. Check the SNMP box and click on the Activate Configuration button.

Step 6: Click on the Administration > Load/Save Configuration and click on the Save to Flash button.

This will save the file in the flash.

Syslog

The syslog-ng daemon provides a modern treatment to system messages. Its basic function is to read and log messages to the system console, log files, other machines (remote syslog servers) and/or users as specified by its configuration file. In addition, syslog-ng is able to filter messages based on their content and to perform an action (e.g. to send an e-mail or pager message). In order to access these functions, the *syslog-ng.conf* file needs some specific configuration.

The configuration file (default: syslog-ng.conf) is read at startup and is reread after reception of a hangup (HUP) signal. When reloading the configuration file, all destination files are closed and reopened as appropriate. The syslog-ng reads from sources (files, TCP/UDP connections, syslogd clients), filters the messages and takes an action (writes in files, sends snmptrap, pager, e-mail or syslogs to remote servers).

There are five tasks required for configuring syslog-ng:

Task 1: Define Global Options.

Task 2: Define Sources.

Task 3: Define Filters.

Task 4: Define Actions (Destinations).

Task 5: Connect all of the above.

The five tasks are explained in the following section “Syslog-ng and its Configuration” on page 223.

Port Slave Parameters Involved with syslog-ng

conf.facility	This value (0-7) is the Local facility sent to the syslog-ng from PortSlave.
conf.DB_facility	This value (0-7) is the Local facility sent to the syslog-ng with data when syslog_buffering and/or alarm is active. When nonzero, the contents of the data buffer are sent to the syslogng every time a quantity of data equal to this parameter is collected. The syslog level for data buffering is hard coded to level five (notice) and facility local[0+ conf.DB_facility]. The file /etc/syslog-ng/syslog-ng.conf should be set accordingly for the syslog-ng to take some action. Example value: 0.
all.syslog_buffering	When nonzero, the contents of the data buffer are sent to the syslog-ng every time a quantity of data equal to this parameter is collected. The syslog message is sent to syslog-ng with NOTICE level and LOCAL[0+conf.DB_facility] facility.

Configuration for CAS, TS, and Dial-in Access

vi Method

To change the PortSlave parameters: edit the */etc/portslave/pslave.conf* file.
To change the syslog-ng configuration: edit the */etc/syslog-ng/syslog-ng.conf* file.

Browser Method

To configure the PortSlave parameters, see the Data Buffering section. To configure syslog via your Web browser:

Step 1: Point your browser to the Console Server.

In the address or location field of your browser type the Console Access Server’s IP address. For example:

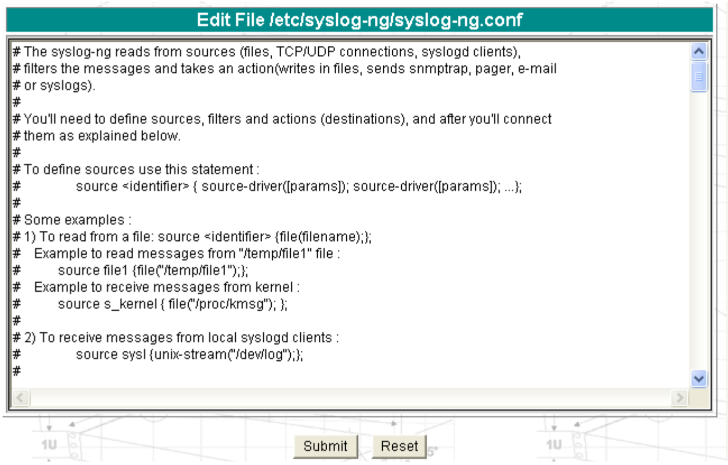
```
http://10.0.0.0
```

Step 2: Log in as root and type the Web root password configured by the Web server.

This will take you to the Configuration and Administration page.

Step 3: Click Syslog on the Configuration section.

Select the Syslog link. The following page will appear, giving information for configuring syslog:



```
# The syslog-ng reads from sources (files, TCP/UDP connections, syslogd clients),
# filters the messages and takes an action(writes in files, sends snmptrap, pager, e-mail
# or syslogs).
#
# You'll need to define sources, filters and actions (destinations), and after you'll connect
# them as explained below.
#
# To define sources use this statement :
#           source <identifier> { source-driver([params]); source-driver([params]); ...};
#
# Some examples :
# 1) To read from a file: source <identifier> {file(filename);};
#    Example to read messages from "/temp/file1" file :
#         source file1 {file("/temp/file1");};
#    Example to receive messages from kernel :
#         source s_kernel { file("/proc/kmsg"); };
#
# 2) To receive messages from local syslogd clients :
#           source sysl {unix-stream("/dev/log");};
#
```

Figure 30: Syslog page 1

Step 4: Edit the configuration file and click on the Submit button

Step 5: Make changes effective.

Click on the Administration > Run Configuration link. Check the Syslog-ng box and click on the Activate Configuration button.

Step 6: Click on the Administration > Load/Save Configuration and click on the Save to Flash button.

This will save the file in the flash.

Wizard Method

Step 1: Bring up the wizard.

At the command prompt, type the following to bring up the PortSlave parameters involved with the Syslog custom wizard:

```
wiz --sl
```

Screen 1 will appear.

Screen 1:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

INSTRUCTIONS for using the Wizard:
You can:
1) Enter the appropriate information for your system
and press ENTER. Enter '#' if you want to
deactivate that parameter or
2) Press ENTER if you are satisfied with the value
within the brackets [ ] and want to go on to the
next parameter or
3) Press ESC if you want to exit.

NOTE: For some parameters, if there is nothing within
the brackets, it will continue to ask for a value.
In that case, you must enter a valid value or # if you
do not wish to configure the value.


Press ENTER to continue...
```

Screen 2:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Current configuration:
(The ones with the '#' means it's not activated.)

conf.facility : 7
conf.DB_facility : 0


Set to defaults? (y/n) [n] :
```

Screen 3:

```
**********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
**********************************************************

CONF.FACILITY - This value (0-7) is the Local facility sent
to the syslog. The file /etc/syslog-ng/syslog-ng.conf
contains a mapping between the facility number and the
action.
(Please see the 'Syslog-ng Configuration to use with Syslog
Buffering Feature' section under Generating Alarms in
Chapter 3 the system's manual for the syslog-ng
configuration file.)

conf.facility[7] :


CONF.DB_FACILITY - This value (0-7) is the Local facility
sent to the syslog with the data when syslog_buffering is
active. The file /etc/syslog-ng/syslog-ng.conf contains a
mapping between the facility number and the action.
(Please see the 'Syslog-ng Configuration to use with Syslog
Buffering Feature' section under Generating Alarms in
Chapter 3 the system's manual for the syslog-ng
configuration file.)

conf.DB_facility[0] :
```

Note: all.syslog_buffering is configured under the wiz - - db.

Screen 4:

```
************************************************************
********* C O N F I G U R A T I O N    W I Z A R D *********
************************************************************


Current configuration:
(The ones with the '#' means it's not activated.)

conf.facility : 7
conf.DB_facility : 0

Are these configuration(s) all correct? (y/n) [n] :
```

If you type 'n'

```
Type 'c' to go back and CORRECT these parameters
or 'q' to QUIT :
```

Typing 'c' repeats the application, typing 'q' exits the entire wiz application

If you type 'y' it leads to Screen 5.

Screen 5:

```
************************************************************
********* C O N F I G U R A T I O N    W I Z A R D *********
************************************************************

(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)


Do you want to activate your configurations now? (y/n) [y] :
```

Screen 6:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Flash refers to a type of memory that can be erased and
reprogrammed in units of memory known as blocks rather than
one byte at a time; thus, making updating to memory easier.


If you choose to save to flash, your configurations thus
far will still be in the memory of the system even after you
reboot it. If you don't save to flash and if you were to
reboot the system, all your new configurations will be lost
and you will have to reconfigure the system.

Do you want to save your configurations to flash? (y/n) [n] :
```

CLI Method

To configure certain parameters for a specific serial port:

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure conf.facility:

```
config configure conf facility <number>
```

To configure DB_facility:

```
config configure conf dbfacility <number>
```

Tip. You can configure all the conf parameters in one line.

```
config configure conf facility <number> dbfacility
<number>
```

Step 2: Activate and Save.

To activate your new configurations and save them to flash, type:

```
config write
```

(This is essentially typing *signal_ras hup* and *saveconf* from the normal terminal prompt.)

The Syslog Functions

This section shows the characteristics of the syslog-ng that is implemented for all members of the Secure Console Port Server SSH. It is divided into three parts:

1. Syslog-ng and its Configuration
2. Syslog-ng Configuration to use with Syslog Buffering Feature
3. Syslog-ng Configuration to use with Multiple Remote Syslog Servers

Syslog-ng and its Configuration

The five tasks previously mentioned are detailed below.

Task 1: Specify Global Options.

You can specify several global options to syslog-ng in the options statement:

```
options { opt1(params); opt2(params); ... };
```

where *optn* can be any of the following:

time_reopen(n) The time to wait before a dead connection is reestablished.

time_reap(n) The time to wait before an idle destination file is closed.

sync_freq(n) The number of lines buffered before written to file. (The file is synced when this number of messages has been written to it.)

mark_freq(n) The number of seconds between two MARKS lines.

log_fifo_size(n) The number of lines fitting to the output queue.

chain_hostname (yes/no) or long_hostname (yes/no) Enable/disable the chained hostname format.

use_time_recvd (yes/no) Use the time a message is received instead of the one specified in the message.

use_dns (yes/no) Enable or disable DNS usage. syslog-ng blocks on DNS queries, so enabling DNS may lead to a Denial of Service attach.

gc_idle_threshold(n) Sets the threshold value for the garbage collector, when syslog-ng is idle. GC phase starts when the number of allocated objects reach this number. Default: 100.

gc_busy_threshold(n) Sets the threshold value for the garbage collector. When syslog-ng is busy, GC phase starts.

create_dirs(yes/no) Enable the creation of new directories.

owner(name) Set the owner of the created file to the one specified. Default: root.

group(name) Set the group of the created file to the one specified. Default: root.

perm(mask) Set the permission mask of the created file to the one specified. Default: 0600.

Task 2: Define sources.

To define sources use this statement:

```
source <identifier> { source-driver([params]); source
driver([params]); ...};
```

where:

identifier — Has to uniquely identify this given source.

source-driver — Is a method of getting a given message.

params — Each source-driver may take parameters. Some of them are required, some of them are optional.

The following source-drivers are available:

a) internal() — Messages are generated internally in syslog-ng.

b) unix-stream (filename [options])

and

unix-dgram (filename [options])

They open the given AF_UNIX socket, and start listening for messages.
Options: owner(name), group(name), perm(mask) are equal global options

keep-alive(yes/no) - Selects whether to keep connections opened when syslog-ng is restarted. Can be used only with unix_stream. Default: yes
max-connections(n) - Limits the number of simultaneously opened connections. Can be used only with unix_stream. Default: 10.

c) tcp([options])

and

udp([options])

These drivers let you receive messages from the network, and as the name of the drivers show, you can use both TCP and UDP. None of tcp() and udp() drivers require positional parameters. By default they bind to 0.0.0.0:514, which means that syslog-ng will listen on all available interfaces.

Options:

ip(<ip address>) - The IP address to bind to. Default: 0.0.0.0.

port(<number>) - UDP/TCP port used to listen messages. Default: 514.

max-connections(n) - Limits the number of simultaneously opened connections. Default: 10.

d) file(filename)

Opens the specified file and reads messages.

e) pipe(filename)

Opens a named pipe with the specified name, and listens for messages. (You'll need to create the pipe using mkfifo command).

Some Examples of Defining Sources

1) To read from a file:

```
source <identifier> {file(filename);};
```

Example to read messages from “/temp/file1” file:

```
source file1 {file(‘/temp/file1’);};
```

Example to receive messages from the kernel:

```
source s_kernel { file(‘/proc/kmsg’); };
```

2) To receive messages from local syslogd clients:

```
source sysl {unix-stream(‘/dev/log’);};
```

3) To receive messages from remote syslogd clients:

```
source s_udp { udp(ip(<cliente ip>) port(<udp port>)); };
```

Example to listen to messages from all machines on UDP port 514:

```
source s_udp { udp(ip(0.0.0.0) port(514));};
```

Example to listen to messages from one client (IP address=10.0.0.1) on UDP port 999:

```
source s_udp_10 { udp(ip(10.0.0.1) port(999)); };
```

Task 3: Define filters.

To define filters use this statement:

```
filter <identifier> { expression; };
```

where:

identifier	Has to uniquely identify this given filter.
expression	Boolean expression using internal functions, which has to evaluate to true for the message to pass.

The following internal functions are available:

a) facility(<facility code>)	Selects messages based on their facility code.
b) *level(<level code>) or priority(<level code>)*	Selects messages based on their priority.
c) *program(<string>)*	Tries to match the <string> to the program name field of the log message.
d) *host(<string>)*	Tries to match the <string> to the hostname field of the log message.
e) *match(<string>)*	Tries to match the <string> to the message itself.

Some Examples of Defining Filters

1) To filter by facility:

```
filter f_facilty { facility(<facility name>); };
```

Examples:

```
filter f_daemon { facility(daemon); };

filter f_kern { facility(kern); };

filter f_debug { not facility(auth, authpriv, news, mail); };
```

2) To filter by level:

```
filter f_level { level(<level name>);};
```

Examples:

```
filter f_messages { level(info .. warn)};

filter f_emergency { level(emerg); };

filter f_alert { level(alert); };
```

3) To filter by matching one string in the received message:

```
filter f_match { match(‘string’); };
```

Example to filter by matching the string “named”:

```
filter f_named { match(‘named’); };
```

4) To filter ALARM messages (note that the following three examples should be one line):

```
filter f_alarm { facility(local[0+<conf.DB_facility>]) and
level(info) and match('ALARM') and match('<your string>'); } ;
```

Example to filter ALARM message with the string “kernel panic”:

```
filter f_kpanic { facility(local[0+<conf.DB_facility>]) and
level(info) and match('ALARM') and match('kernel panic'); };
```

Example to filter ALARM message with the string “root login”:

```
filter f_root { facility(local[0+<conf.DB_facility>]) and
level(info) and match('ALARM') and match('root login'); };
```

5) To eliminate sshd debug messages:

```
filter f_sshd_debug { not program('sshd') or not level(debug); };
```

6) To filter the syslog_buffering:

```
filter f_syslog_buf { facility(local[0+<conf.DB_facility>]) and
level(notice); };
```

Task 4: Define Actions.

To define actions use this statement (note that the statement should be one line):

```
destination <identifier> { destination-driver([params]);
destination-driver([param]); ..};
```

where:

identifier	Has to uniquely identify this given destination.
destination driver	Is a method of outputting a given message.
params	Each destination-driver may take parameters. Some of them required, some of them are optional.

The following destination drivers are available:

a) file(filename [options])

This is one of the most important destination drivers in syslog-ng. It allows you to output log messages to the named file. The destination filename may include macros (by prefixing the macro name with a '$' sign) which gets expanded when the message is written. Since the state of each created file must be tracked by syslog-ng, it consumes some memory for each file. If no new messages are written to a file within 60 seconds (controlled by the time_reap global option), it's closed, and its state is freed.

Available macros in filename expansion:
HOST - The name of the source host where the message originated from.
FACILITY - The name of the facility the message is tagged as coming from.
PRIORITY or LEVEL - The priority of the message.
PROGRAM - The name of the program the message was sent by.
YEAR, MONTH, DAY, HOUR, MIN, SEC - The year, month, day, hour, min, sec of the message was sent.
TAG - Equals FACILITY/LEVEL.
FULLHOST - The name of the source host and the source-driver:
<source-driver>@<hostname>
MSG or MESSAGE - The message received.
FULLDATE - The date of the message was sent.
Available options:
log_fifo_size(number) - The number of entries in the output file.
sync_freq(number) - The file is synced when this number of messages has been written to it.
owner(name), group(name), perm(mask) - Equals global options.
template("string") - Syslog-ng writes the "string" in the file. You can use the MACROS in the string.
encrypt(yes/no) - Encrypts the resulting file.
compress(yes/no) - Compresses the resulting file using zlib.

b) *pipe(filename [options])*
This driver sends messages to a named pipe. Available options:
owner(name), group(name), perm(mask) - Equals global options.
template("string") - Syslog-ng writes the "string" in the file. You can use the MACROS in the string.

c) *unix-stream(filename) and unix-dgram(filename)*
This driver sends messages to a UNIX socket in either SOCKET_STREAM or SOCK_DGRAM mode.

d) *udp("<ip address>" port(number);) and tcp("<ip address>" port(number);)*
This driver sends messages to another host (ip address/port) using either UDP or TCP protocol.

e) *usertty(<username>)*
This driver writes messages to the terminal of a logged-in username.

f) *program(<program name and arguments>)*
This driver fork()'s executes the given program with the arguments and sends messages down to the stdin of the child.

Some Examples of Defining Actions

1) To send e-mail:

```
destination <ident> { pipe(‘/dev/cyc_alarm’ template(‘sendmail
<pars>’));};
```

where ident: uniquely identifies this destination. Parameters:

-t <name>[,<name>]	To address
[-c <name>[,<name>]]	CC address
[-b <name>[,<name>]]	Bcc address
[-r <name>[,<name>]]	Reply-to address
-f <name>	From address
-s \"<text>\"	Subject
-m \”<text message>\”	Message
-h <IP address or name>	SMTP server
[-p <port>]	Port used. default:25

To mount the message, use this macro:

$FULLDATE	The complete date when the message was sent.
$FACILITY	The facility of the message.
$PRIORITY or $LEVEL	The priority of the message.
$PROGRAM	The message was sent by this program (BUFFERING or SOCK).

$HOST	The name of the source host.
$FULLHOST	The name of the source host and the source driver. Format: <source>@<hostname>
$MSG or $MESSAGE	The message received.

Example to send e-mail to z@none.com (SMTP's IP address 10.0.0.2) from the e-mail address a@none.com with subject “Secure Console Port Server SSH-ALARM”. The message will carry the current date, the host-name of this Secure Console Port Server SSH and the message that was received from the source.

```
destination d_mail1 {

      pipe('/dev/cyc_alarm'

       template('sendmail -t z@none.com -f a@none.com -s \'Secure Console
Port Server SSH-ALARM\' \

             -m \'$FULLDATE $HOST $MSG\' -h 10.0.0.2'));

      };
```

2) To send to pager server (sms server):

```
destination <ident> {pipe(‘/dev/cyc_alarm’ template(‘sendsms
<pars>’));};

where ident: uniquely identify this destination

pars: -d <mobile phone number>

-m \’<message - max.size 160 characters>\’

-u <username to login on sms server>

-p <port sms - default : 6701>

<server IP address or name>
```

Example to send a pager to phone number 123 (Pager server at 10.0.0.1) with message carrying the current date, the hostname of this Secure Console Port Server SSH and the message that was received from the source:

```
destination d_pager {

pipe(‘/dev/cyc_alarm’

template(‘sendsms -d 123 -m \’$FULLDATE $HOST $MSG\’ 10.0.0.1’));

};
```

3) To send snmptrap:

```
destination <ident> {pipe(‘/dev/cyc_alarm’ template(‘snmptrap
<pars>’)); };

where ident : uniquely identify this destination

pars : -v 1

<snmptrapd IP address>

public : community

\"\" : enterprise-oid

\"\" : agent/hostname

<trap number> : 2-Link Down, 3-Link Up, 4-Authentication Failure
0 : specific trap

\"\" : host-uptime

.1.3.6.1.2.1.2.2.1.2.1 :interfaces.iftable.ifentry.ifdescr.1

s : the type of the next field (it is a string)

\"<message - max. size 250 characters>\"
```

Example to send a Link Down trap to server at 10.0.0.1 with message carrying the current date, the hostname of this Secure Console Port Server SSH and the message that was received from the source:

```
destination d_trap {

pipe("/dev/cyc_alarm"
template("snmptrap -v1 10.0.0.1 public \"\" \"\" 2 0 \"\" \
.1.3.6.1.2.1.2.2.1.2.1 s \"$FULLDATE $HOST $MSG\" "));

};
```

4) To write in file :

```
destination d_file { file(<filename>);};

Example send message to console :

destination d_console { file("/dev/ttyS0");};
```

Example to write a message in /var/log/messages file:

```
destination d_message { file("/var/log/messages"); };
```

5) To write messages to the session of a logged-in user:

```
destination d_user { usertty("<username>"); };
```

Example to send message to all sessions with root user logged:

```
destination d_userroot { usertty("root"); };
```

6) To send a message to a remote syslogd server:

```
destination d_udp { udp("<remote IP address>" port(514)); };
```

Example to send syslogs to syslogd located at 10.0.0.1 :

```
destination d_udp1 { udp("10.0.0.1" port(514)); };
```

Task 5: Connect all of the above.

To connect the sources, filters, and actions, use the following statement. (Actions would be any message coming from one of the listed sources. A match for each of the filters is sent to the listed destinations.)

```
log { source(S1); source(S2); ...

filter(F1);filter(F2);...
destination(D1); destination(D2);...
};
```

where :

Sx	Identifier of the sources defined before.
Fx	Identifier of the filters defined before.
Dx	Identifier of the actions/destinations defined before.

Examples:
1) To send all messages received from local syslog clients to console:

```
log { source(sysl); destination(d_console);};
```

2) To send only messages with level alert and received from local syslog clients to all logged root user:

```
log { source(sysl); filter(f_alert); destination(d_userroot); };
```

3) To write all messages with levels info, notice, or warning and received from syslog clients (local and remote) to /var/log/messages file:

```
log { source(sysl); source(s_udp); filter(f_messages); destina-
tion(d_messages); };
```

4) To send e-mail if message received from local syslog client has the string “kernel panic”:

```
log { source(sysl); filter(f_kpanic); destination(d_mail1); };
```

5) To send e-mail and pager if message received from local syslog client has the string “root login”:

```
log { source(sysl); filter(f_root); destination(d_mail1); destina-
tion(d_pager); };
```

6) To send messages with facility kernel and received from syslog clients (local and remote) to remote syslogd:

```
log { source(sysl); source(s_udp); filter(f_kern); destination(d-
udp1); };
```

Syslog-ng Configuration to use with Syslog Buffering Feature

This configuration example uses the syslog buffering feature, and sends messages to the remote syslogd (10.0.0.1).

Step 1: Configure pslave.conf parameters.

In the pslave.conf file the parameters of the syslog buffering feature are configured as:

```
conf.DB_facility 1

all.syslog_buffering 100
```

Step 2: Add lines to syslog-ng.conf.

Add the following lines by vi or browser to the file:

```
# local syslog clients

source src { unix-stream("/dev/log"); };

destination d_buffering { udp("10.0.0.1"); };

filter f_buffering { facility(local1) and level(notice); };

# send only syslog_buffering messages to remote server

log { source(src); filter(f_buffering); destina-
tion(d_buffering); };
```

Syslog-ng Configuration to use with Multiple Remote Syslog Servers

This configuration example is used with multiple remote syslog servers.

Step 1: Configure pslave.conf parameters.

In the pslave.conf file the facility parameter is configured as:

```
conf.facility 1
```

Step 2: Add lines to syslog-ng.conf.

The syslog-ng.conf file needs these lines:

```
# local syslog clients
```

```
source src { unix-stream("/dev/log"); };
# remote server 1 - IP address 10.0.0.1 port default
destination d_udp1 { udp("10.0.0.1"); };
# remote server 2 - IP address 10.0.0.2 port 1999
destination d_udp2 { udp("10.0.0.2" port(1999););};
# filter messages from facility local1 and level info to warning
filter f_local1 { facility(local1) and level(info..warn);};
# filter messages from facility local 1 and level err to alert
filter f_critic { facility(local1) and level(err .. alert);};
# send info, notice and warning messages to remote server udp1
log { source(src); filter(f_local1); destination(d_udp1); };
# send error, critical and alert messages to remote server udp2
log { source(src); filter(f_critic); destination(d_udp2); };
```

Terminal Appearance

You can change the format of the login prompt and banner that is issued when a connection is made to the system. Prompt and banner appearance can be port-specific as well.

Parameters Involved and Passed Values

Terminal Appearance involves the following parameters:

all.prompt This text defines the format of the login prompt. Expansion characters can be used here. Example value: %h login:

all.issue This text determines the format of the login banner that is issued when a connection is made to the Secure Console Port Server SSH. \n represents a new line and \r represents a carriage return. Expansion characters can be used here.
Value for this Example:

```
\r\n\
Welcome to terminal server %h port S%p \n\
\r\n
```

all.lf_suppress This activates line feed suppression. When configured as 0, line feed suppression will not be performed. When 1, extra line feed will be suppressed.

all.auto_answer _input This parameter is used in conjunction with the next parameter, auto_answer_output. If configured and if there is no session established to the port, this parameter will constantly be compared and matched up to the string of bytes coming in remotely from the server. If a match is found, the string configured in auto_answer_output is sent back to the server. To represent the ESC character as part of this string, use the control character, ^[.

all.auto_answer_output	This parameter is used in conjunction with the previous parameter, auto_answer_input. If configured, and if there is no session established to the port, this parameter is sent back to the server when there is a match between the incoming data and auto_answer_input. To represent the ESC character as part of this string, use the control character, ^[.

Configuration for CAS, TS, and Dial-in Access

Browser Method

Step 1: Point your browser to the Console Server.

In the address or location field of your browser type the Console Access Server's IP address. For example:

```
http://10.0.0.0
```

Step 2: Log in as root and type the Web root password configured by the Web server.

This will take you to the Configuration and Administration page.

Step 3: Select the Serial Ports link.

Click on the Serial Ports link on the Link Panel to the left of the page or in the Configuration section of the page. This will take you to the Port Selection page.

Step 4: Select port(s).

On the Port Selection page, choose all ports or an individual port to configure, from the dropdown menu. Click the Submit button. This will take you to the Serial Port Configuration page.

Step 5: Scroll down to the Terminal Server section.

You can change the settings for Banner Field (issue) and Login Prompt field here.

Step 6: Click on the Submit button.

Step 7: Make the changes effective.

Click on the Administration > Run Configuration link, check the Serial Ports/ Ethernet/Static Routes box and click on the Activate Configuration button.

Step 8: Click on the link Administration > Load/Save Configuration.

Step 9: Click the Save Configuration to Flash button.

The configuration was saved in flash.

Wizard Method

Step 1: Bring up the wizard.

At the command prompt, type the following to bring up the Terminal Appearance custom wizard:

```
wiz --tl
```

Screen 1 will appear.

Screen 1:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

INSTRUCTIONS for using the Wizard:
You can:
1) Enter the appropriate information for your system
and press ENTER. Enter '#' if you want to
deactivate that parameter or
2) Press ENTER if you are satisfied with the value
within the brackets [ ] and want to go on to the
next parameter or
3) Press ESC if you want to exit.


NOTE: For some parameters, if there is nothing within
the brackets, it will continue to ask for a value.
In that case, you must enter a valid value or # if you
do not wish to configure the value.
```

```
Press ENTER to continue...
```

Screen 2:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
Current configuration:
(The ones with the '#' means it's not activated.)


all.issue : \r\n\Welcome to terminal server %h port S%p \n\
\r\n\
all.prompt : %h login:
all.lf_suppress : 0
all.auto_answer_input : #
all.auto_answer_output : #

Set to defaults? (y/n) [n] :
```

Screen 3:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************
ALL.ISSUE - This text determines the format of the login
banner that is issued when a connection is made to the
system. \n represents a new line and \r respresents a
carriage return.


all.issue[\r\n\Welcome to terminal server %h port S%p \n\
\r\n\] :

ALL.PROMPT - This text defines the format of the login
prompt.


all.prompt[%h login:] :
```

Screen 4:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
ALL.LF_SUPPRESS - This activates line feed suppression.
When configured as 0, line feed suppression will not be
performed. When 1, extra line feed will be suppressed.


all.lf_suppress[0] :

ALL.AUTO_ANSWER_INPUT - This parameter is used in conjunc-
tion with the next parameter, auto_answer_output. Please
refer to the manual for more info.


If configured and if there is no session established to
the port, this parameter will constantly be compared and
matched up to the string of bytes coming in remotely from
the server. If a match is found, the string configured in
auto_answer_output is sent back to the server. To repre-
sent the ESC character as part of this string, use the
control character, ^[.


all.auto_answer_input[#] :
```

Screen 5:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


ALL.AUTO_ANSWER_OUTPUT - This parameter is used in conjunc-
tion with the previous parameter, auto_answer_input. Please
refer to the manual for more info.
If configured, and if there is no session established to
the port, this parameter is sent back to the server
when there is a match between the incoming data and
auto_answer_input. To represent the ESC character as part
of this string, use the control character, ^[.
```

```
all.auto_answer_output[#] :
```

Screen 6:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************
Current configuration:
(The ones with the '#' means it's not activated.)

all.issue : \r\n\Welcome to terminal server %h port S%p \n\
\r\n\
all.prompt : %h login:
all.lf_suppress : 0
all.auto_answer_input : #
all.auto_answer_output : #

Are these configuration(s) all correct? (y/n) [n] :
```

If you type 'N'

```
Type 'c' to go back and CORRECT these parameters
or 'q' to QUIT :
```

Typing 'c' repeats the application, typing 'q' exits the entire wiz application

If you type 'Y'

```
Discard previous port-specific parameters? (y/n) [n] :
```

Note: Answering yes to this question will discard only the parameter(s) which you are currently configuring if they were configured for a specific port in a previous session. For instance, if you are currently configuring parameter, all.x, and there was a specific port, s2.x, configured; then, answering yes to this question will discard s2.x.

```
Type 'c' to CONTINUE to set these parameters for
specific ports or 'q' to QUIT :
```

Typing 'c' leads to Screen 7, typing 'q' leads to Screen 8.

Screen 7:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************


You have 8 available ports on this system.

Type 'q' to quit, a valid port number[1-8], or anything
else to refresh :
```

Screen 8:

```
***********************************************************
********* C O N F I G U R A T I O N W I Z A R D *********
***********************************************************

(Note: If you are NOT connected to this unit through a
console, and you have just reconfigured the IP of this
unit, activating the new configurations may cause you to
lose connection. In that case, please reconnect to the
unit by the new IP address, and manually issue a saveconf
to save your configurations to flash.)


Do you want to activate your configurations now? (y/n) [y] :
```

Screen 9:

```
***********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
***********************************************************

Flash refers to a type of memory that can be erased and
reprogrammed in units of memory known as blocks rather than
one byte at a time; thus, making updating to memory easier.


If you choose to save to flash, your configurations thus
```

```
far will still be in the memory of the system even after you
reboot it. If you don't save to flash and if you were to
reboot the system, all your new configurations will be lost
and you will have to reconfigure the system.

Do you want to save your configurations to flash? (y/n) [n] :
```

CLI Method

To configure certain parameters for a specific serial port:

Step 1: At the command prompt, type in the appropriate command to configure desired parameters.

To activate the serial port. <string> should be ttyS<serial port number> :

```
config configure line <serial port number> tty <string>
```

To configure issue:

```
config configure line <serial port number> issue <string>
```

To configure prompt:

```
config configure line <serial port number> prompt <string>
```

To configure lf_suppress:

```
config configure line <serial port number> lf <number>
```

To configure auto_answer_input:

```
config configure line <serial port number> auto_input
<string>
```

To configure auto_answer_output:

```
config configure line <serial port number> auto_output
<string>
```

Tip. You can configure all the parameters for a serial port in one line.

```
config configure line <serial port number> tty <string>
issue <string> prompt <string> lf <number> auto_input
<string> auto_output <string>
```

Step 2: Activate and Save.

To activate your new configurations and save them to flash, type:

```
config write
```

(This is essentially typing *signal_ras hup* and *saveconf* from the normal terminal prompt.)

Chapter 3 - Additional Features

Time Zone

The content of the file /etc/TIMEZONE can be in one of two formats. The first format is used when there is no daylight savings time in the local time zone:

```
std offset
```

The *std* string specifies the name of the time zone and must be three or more alphabetic characters. The offset string immediately follows std and specifies the time value to be added to the local time to get *Coordinated Universal Time* (UTC). The offset is positive if the local time zone is west of the Prime Meridian and negative if it is east. The hour must be between 0 and 24, and the minutes and seconds must be between 0 and 59.

The second format is used when there is daylight savings time:

```
std offset dst [offset],start[/time],end[/time]
```

There are no spaces in the specification. The initial std and offset specify the Standard Time zone, as described above. The dst string and offset specify the name and offset for the corresponding daylight savings time zone. If the offset is omitted, it defaults to one hour ahead of Standard Time.

The start field specifies when daylight savings time goes into effect and the end field specifies when the change is made back to Standard Time. These fields may have the following formats:

Jn This specifies the Julian day, with n being between 1 and 365. February 29 is never counted even in leap years.

n This specifies the Julian day, with n being between 1 and 365. February 29 is counted in leap years.

Mm.w.d This specifies day, d (0 < d < 6) of week w (1 < w < 5) of month m (1 < m < 12). Week 1 is the first week in which day d occurs and week 5 is the last week in which day d occurs. Day 0 is a Sunday.

The time fields specify when, in the local time currently in effect, the change to the other time occurs. If omitted, the default is 02:00:00.

In the example below:

```
GST+7DST+6M4.1.0/14:30.M10.5.6/10
```

Daylight Savings Time starts on the first Sunday of April at 2:30 p.m. and it ends on the last Saturday of October at 10:00 a.m.

How to set Date and Time

The date command prints or sets the system date and time. Format of the command:

```
date [MMDDhhmm[[CC]YY]
^ ^ ^ ^ ^ ^
^ ^ ^ ^ ^ year
^ ^ ^ ^ century
^ ^ ^ minute
^ ^ hour
^ day
month
```

For example:

```
date 101014452002
```

produces:

```
Thu Oct 10 14:45:00 DST 2002
```

The DST is because it was specified in /etc/TIMEZONE.

Appendix A - New User Background Information

Users and Passwords

A username and password are necessary to log in to the Secure Console Port Server SSH. The user *root* is predefined, with a password *tslinux*. A password should be configured as soon as possible to avoid unauthorized access. Type the command:

```
passwd
```

to create a password for the root user. To create a regular user (without root privileges), use the commands:

```
adduser user_name

passwd user_password
```

To log out, type “logout” at the command prompt.

How to show who is logged in and what they are doing

The command “w” displays information about the users currently on the machine, and their processes. It calls two commands: w_ori and w_cas. The w_ori is the new name of the original command “w” and the w_cas shows the CAS sessions information.

The header of w_ori shows, in this order: the current time, how long the system has been running, how many users are currently logged on (excluded the CAS users), and the system load averages for the past 1, 5, and 15 minutes.

The following entries are displayed for each user (excluded the CAS users): login name, the tty name, the remote host, login time, idle time, JCPU time (it is the time used by all processes attached to the tty), PCPU time (it is the time used by the current process, named in the “what” field), and the command line of their current process.

The header of w_cas shows how many CAS users are currently logged on. The following entries are displayed for each CAS user: login name, the tty name, the remote host and remote port, login time, the process ID and the command line of the current process.

Appendix A - New User Background Information

Linux File Structure

The Linux file system is organized hierarchically, with the base (or root) directory represented by the symbol “/”. All folders and files are nested within each other below this base directory. The directories located just below the base directory are:

/home Contains the work directories of system users.

/bin Contains applications and utilities used during system initialization.

/dev Contains files for devices and ports.

/etc Contains configuration files specific to the operating system.

/lib Contains shared libraries.

/proc Contains process information.

/mnt Contains information about mounted disks.

/opt Location where packages not supplied with the operating system are stored.

/tmp Location where temporary files are stored.

/usr Contains most of the operating system files.

/var Contains operating system data files.

Appendix A - New User Background Information

Basic File Manipulation Commands

The basic file manipulation commands allow the user to copy, delete, and move files and create and delete directories.

cp *file_name destination*
a) cp text.txt /tmp
b) cp /chap/robo.php ./excess.php

Copies the file indicated by file_name to the path indicated by destination. a) Copies the file text.txt in the current directory to the tmp directory.
b) Copies the file robo.php in the chap directory to the current directory and renames the copy excess.php.

rm *file_name*

Removes the file indicated by file_name.

mv *file_name destination*

Moves the file indicated by file_name to the path indicated by destination.

mkdir *directory_name*
a) mkdir spot
b) mkdir /tmp/snuggles

Creates a directory named directory_name.
a) creates the directory spot in the current directory. b) creates the directory snuggles in the directory tmp.

rmdir *directory_name*

Removes the directory indicated by directory_name.

Other commands allow the user to change directories and see the contents of a directory.

pwd

Supplies the name of the current directory. While logged in, the user is always “in” a directory. The default initial directory is the user's home directory: /home/<username>

ls [options] directory_name

Lists the files and directories within directory_name. Some useful options are -l for more detailed output and -a which shows hidden system files.

cd *directory_name*

Changes the directory to the one specified.

cat *file_name*

Prints the contents of file_name to the screen.

Shortcuts:

. (one dot)	Represents the current directory.
.. (two dots)	Represents one directory above the current directory (i.e. one directory closer to the base directory).

The vi Editor

To edit a file using the vi editor, type:

```
vi file_name
```

Vi is a three-state line editor: it has a command mode, a line mode and an editing mode. If in doubt as to which mode you are in, press the <ESC> key which will bring you to the command mode.

Table 15: vi modes

Mode	What is done there	How to get there
Command mode	Navigation within the open file.	Press the <ESC> key.
Editing mode	Text editing.	See list of editing commands below.
Line mode	File saving, opening, etc. Exiting from vi.	From the command mode, type “:” (colon).

When you enter the vi program, you are automatically in command mode. To navigate to the part of the file you wish to edit, use the following keys:

Table 16: vi navigation commands

h	Moves the cursor to the left (left arrow).
j	Moves the cursor to the next line (down arrow).
k	Moves the cursor to the previous line (up arrow).
l	Moves the cursor to the right (right arrow).

Having arrived at the location where text should be changed, use these commands to modify the text (note commands “i” and “o” will move you into edit mode and everything typed will be taken literally until you press the <ESC> key to return to the command mode).

Table 17: vi file modification commands

i	Inserts text before the cursor position (everything to the right of the cursor is shifted right).
o	Creates a new line below the current line and insert text (all lines are shifted down).
dd	Removes the entire current line.
x	Deletes the letter at the cursor position.

After you have finished modifying a file, enter line mode (by typing “:” from command mode) and use one of the following commands:

Table 18: vi line mode commands

w	Saves the file (w is for write).
wq	Saves and closes the file (q is for quit).
q!	Closes the file without saving.
w *file*	Saves the file with the name <*file*>.
e *file*	Opens the file named <*file*>.

Appendix A - New User Background Information

The Routing Table

The Secure Console Port Server SSH has a static routing table that can be seen using the commands:

```
route
```

or

```
netstat -rn
```

The file /etc/network/st_routes is the Secure Console Port Server SSH's method for configuring static routes. Routes should be added to the file (which is a script run when the Secure Console Port Server SSH is initialized) or at the prompt (for temporary routes) using the following syntax:

```
route [add|del] [-net|-host] target netmask nt_msk [gw gt_way]
interf
```

[add|del] One of these tags must be present. Routes can be either added or deleted.

[-net|-host] Net is for routes to a network and -host is for routes to a single host.

target Target is the IP address of the destination host or network.

netmask
nt_msk The tag *netmask* and *nt_mask* are necessary only when subnetting is used, otherwise, a mask appropriate to the target is assumed. nt_msk must be specified in dot notation.

gw gt_way Specifies a gateway, when applicable. gt_way is the IP address or hostname of the gateway.

interf The interface to use for this route. Must be specified if a gateway is not. When a gateway is specified, the operating system determines which interface is to be used.

Appendix A - New User Background Information

Secure Shell Session

Ssh is a command interface and protocol often used by network administrators to connect securely to a remote computer. Ssh replaces its non-secure counterpart rsh and rlogin. There are two versions of the protocol, ssh and ssh2. The Secure Console Port Server SSH offers both. The command to start an ssh client session from a UNIX workstation is:

```
ssh -t <user>@<hostname>
```

where

```
<user> = <username>:ttySnn or
         <username>:socket_port or
         <username>:ip_addr or
         <username>:serverfarm
```

Note: “serverfarm” is a physical port alias. It can be configured in the file pslave.conf.
An example:

username:	mycompany
16-port IP address:	192.168.160.1
host name:	16-port
servername for port 1:	file_server

ttyS1 is addressed by IP 10.0.0.1 or socket port 7001. The various ways to access the server connected to the port are:

```
ssh -t mycompany:ttyS1@16-port
ssh -t mycompany:7001@16-port
ssh -t mycompany:10.0.0.1@16-port
ssh -t mycompany:file_server@16-port
ssh -t -l mycompany:10.0.0.116-port
```

```
ssh -t -l mycompany:7001 16-port
```

For openssh clients, version 3.1p1 or later ssh2 is the default. In that case, the -1 flag is used for ssh1.

```
ssh -t mycompany:7001@16-port
```

(openssh earlier than 3.1p1 - Secure Console Port Server SSH V_1.3.1 and earlier -> ssh1 will be used)

```
ssh -t -2 mycompany:7001@16-port
```

(openssh earlier than 3.1p1 - Secure Console Port Server SSH V_1.3.1 and earlier -> ssh2 will be used)

```
ssh -t mycompany:7001@16-port
```

(openssh 3.1p1 or later - Secure Console Port Server SSH V_1.3.2 or later -> ssh2 will be used)

```
ssh -t -1 mycompany:7001@16-port
```

(openssh 3.1p1 or later - Secure Console Port Server SSH V_1.3.2 or later -> ssh1 will be used)

To log in to a port that does not require authentication, the username is not necessary:

```
ssh -t -2 :ttyS1@16-port
```

Note: In this case, the file sshd_config must be changed in the following way:

```
PermitRootLogin Yes

PermitEmptyPassword Yes
```

Configuring sshd's client authentication using SSH Protocol version 1

Step 1: Only RhostsAuthentication yes in sshd_config.

In the linux host enable in the file /etc/ssh/ssh_config the parameters:

```
Host *
```

```
RhostsAuthentication yes

UsePrivilegedPort yes
```

- One of these:

```
hostname or ipaddress in /etc/hosts.equiv or
/etc/ssh/shosts.equiv

hostname or ipaddress and username in ~/.rhosts or ~/.shosts
and IgnoreRhosts no in sshd_config
```

- Client start-up command: ssh -t <Secure Console Port Server SSH_ip or Serial_port_ip> (if the ssh client is running under a session belonging to a username present both in the workstation's database and the Secure Console Port Server SSH's database).
- Client start-up command: ssh -t -l <username> <Secure Console Port Server SSH_ip or Serial_port_ip> (if the ssh client is running under a session belonging to a username present only in the workstation's database. In this case, the <username> indicated would have to be a username present in the Secure Console Port Server SSH's database).

Note: For security reasons, some ssh clients do not allow just this type of authentication. To access the serial port, the Secure Console Port Server SSH must be configured for local authentication. No root user should be used as username.

Step 2: Only RhostsRSAAuthentication yes in sshd_config.

- One of the RhostsAuthentication settings, described in Step 1.
- Client machine's host key ($ETC/ssh_host_key.pub) copied into the TS/tmp/known_hosts file. The client hostname plus the information inside this file must be appended in one single line inside the file /etc/ssh/ ssh_known_hosts or ~/.ssh/known_hosts and IgnoreUserKnownHosts no inside sshd_config. The following commands can be used for example:

```
echo 'n 'client_hostname ' >> /etc/ssh/ssh_known_hosts or ~/.ssh/
known_hosts
```

```
cat /tmp/known_hosts >> /etc/ssh/ssh_known_hosts or ~/.ssh/
known_hosts
```

- client start-up command: ssh -t <Secure Console Port Server SSH_ip or Serial_port_ip>

Note: "client_hostname" should be the DNS name. To access the serial port, the Secure Console Port Server SSH must be configured for local authentication. No root user should be used as username.

Step 3: Only RSAAuthentication yes in sshd_config.

- Removal of the Secure Console Port Server SSH's *.equiv, ~/.?hosts, and *known_hosts files.
- Client identity created by ssh-keygen and its public part (~/.ssh/identity.pub) copied into Secure Console Port Server SSH 's ~/.ssh/authorized_keys.
- Client start-up command: ssh -t <Secure Console Port Server SSH_ip or Serial_port_ip>.

Step 4: Only PasswdAuthentication yes in sshd_config.

- Removal of the Secure Console Port Server SSH's *.equiv, ~/.?hosts, *known_hosts, and *authorized_keys files.
- Client startup command: ssh –t -l <username> <Secure Console Port Server SSH_ip or Serial_port_ip> or ssh –t –l <username:alias><Secure Console Port Server SSH_ip>.

Configuring sshd's client authentication using SSH Protocol version 2

Only PasswdAuthentication yes in sshd_config DSA Authentication is the default. (Make sure the parameter PubkeyAuthentication is enabled.)

- Client DSA identity created by ssh-keygen -d and its public part (~/.ssh/id_dsa.pub) copied into the Secure Console Port Server SSH's ~/.ssh/authorized_keys2 file.
- Password Authentication is performed if DSA key is not known to the Secure Console Port Server SSH. Client start-up command: ssh -2 -t <TS_ip or Serial_port_ip>.

Appendix A - New User Background Information

Note: All files "~/*" or "~/.ssh/*" must be owned by the user and readable only by others. All files created or updated must have their full path and file name inside the file config_files and the command saveconf must be executed before rebooting the Secure Console Port Server SSH.

The Process Table

The process table shows which processes are running. Type *ps -a* to see a table similar to that below.

Table 19: Process table

PID	UID	State	Command
1	root	S	/sbin/inetd
31	root	S	/sbin/sshd
32	root	S	/sbin/cy_ras
36	root	S	/sbin/cy_wdt_led wdt led
154	root	R	/ps -a

To restart the cy_ras process use its process ID or execute the command:

```
signal_ras hup
```

This executes the ps command, searches for the cy_ras process id, then sends the signal *hup* to the process, all in one step. Never kill cy_ras with the signals -9 or SIGKILL.

Appendix A - New User Background Information

TS Menu Script

The ts_menu script can be used to avoid typing long telnet or ssh commands. It presents a short menu with the names of the servers connected to the serial ports of the Secure Console Port Server SSH. The server is selected by its corresponding number. ts_menu must be executed from a local session: via console, telnet, ssh, dumb terminal connected to a serial port, etc. Only ports configured for console access (protocols socket_server or socket_ssh) will be presented. To start having familiarity with this application, run ts_menu - h:

```
> ts_menu -h

USAGE: ts_menu options

-p : Display Ethernet Ip and Tcp port

-i : Display local Ip assigned to the serial port

-u <name> : Username to be used in ssh/telnet command

-U : Allows choosing of different usernames for different ports

-h : print this help message


> ts_menu

Master and Slaves Console Server Connection Menu

1 TSJen800

2 edson-r4.mycompany.com

3 az84.mycompanys.com

4 64.186.190.85

5 az85.mycompany.com

Type 'q' to quit, a valid option [1-5], or anything else to
refresh:
```

By selecting 1 in this example, the user will access the local serial ports on that Secure Console Port Server SSH. If the user selects 2 through 5, remote serial ports will be accessed. This is used when there is clustering (one Secure Console Port Server SSH master box and one or more Secure Console Port Server SSH slave boxes).

If the user selects 1, the following screen is displayed:

```
Serial Console Server Connection Menu for your Master Terminal
Server

1 ttyS1 2 ttyS2 3 s3serverfarm

Type 'q' to quit, 'b' to return to previous menu, a valid option[1-

3], or anything else to refresh:
```

Options 1 to 3 in this case are serial ports configured to work as a CAS profile. Serial port 3 is presented as an alias name (s3serverfarm). When no name is configured in pslave.conf, ttyS<N> is used instead. Once the serial port is selected, the username and password for that port (in case there is a per-user access to the port and -U is passed as parameter) will be presented, and access is granted.

To access remote serial ports, the presentation will follow a similar approach to the one used for local serial ports.

The ts_menu script has the following line options:

-p : Displays Ethernet IP Address and TCP port instead of server names.

```
Secure Console Port Server SSH: Serial Console Server Connection
menu

1 209.81.55.79 7001 2 209.81.55.79 7002 3 209.81.55.79 7003

4 209.81.55.79 7004 5 209.81.55.79 7005 6 209.81.55.79 7006

Type 'q' to quit, a valid option [1-6], or anything else to refresh
:
```

-i : Displays Local IP assigned to the serial port instead of server names.

```
Secure Console Port Server SSH: Serial Console Server Connection
menu
```

```
1 192.168.1.101 2 192.168.1.102 3 192.168.1.103 4 192.168.1.104

5 192.168.1.105 6 192.168.1.106

Type 'q' to quit, a valid option [1-6], or anything else to refresh
:
```

-u <name> : Username to be used in the ssh/telnet command. The default username is that used to log onto the Secure Console Port Server SSH.

-h : Lists script options.

Appendix B - Cabling, Hardware, & Electrical

General Hardware Specifications

The power requirements, environmental conditions and physical specifications of the Secure Console Port Server SSH are listed below.

Table 20: Secure Console Port Server SSH power requirements

Power Specifications						
	1-Port	4-Port	8-Port	16-Port	32-Port	48-Port
Input Voltage Range	External Universal Input Desktop Power Supply, 100-240VAC auto-range input, 5VDC output (Internal power modules available for 12VDC, 24VDC, -48VDC and Power Over Ethernet)	External Universal Input Desktop Power Supply (100-240VAC auto-range input, 5VDC output)	External Universal Input Desktop Power Supply (100-240VAC auto-range input, 5VDC output)	Internal 100-240VAC autorange (-48VDC option available)	Internal 100-240VAC autorange (-48VDC option available)	Internal 100-240VAC autorange
Input Frequency Range	50/60H	50/60H	50/60H	50/60H	50/60H	50/60H
Power @120VAC	5 W max	5 W max	6 W max	22 W max	26 W max	11 W max
Power @220 VAC	6 W max	6 W max	8 W max	28 W max	37 W max	17 W max

Appendix B - Cabling, Hardware, & Electrical

Table 21: Secure Console Port Server SSH environmental conditions

Environmental Information						
	1-Port	4-Port	8-Port	16-Port	32-Port	48-Port
Operating Temp- erature	50F to 122F (10°C to 50°C)	50F to 112F (10°C to 44°C)	50F to 112F (10°C to 44°C)	50F to 112F (10°C to 44°C)	50F to 112F (10°C to 44°C)	50F to 112F (10°C to 44°C)
Relative Humidity	10 - 90%, non-condensing	10 - 90%, non-condensing	10 - 90%, non-condensing	10 - 90%, non-condensing	10 - 90%, non-condensing	10 - 90%, non-condensing

Table 22: Secure Console Port Server SSH physical specifications

Physical Information							
	1_Port		4-Port	8-Port	16-Port	32-Port	48-Port
External Dim- ensions	2.76 x 3.35 x 1.18 in.		8.5 x 4.75 x 1 in.	8.5 x 4.75 x 1 in.	17 x 8.5 x 1.75 in.	17 x 8.5 x 1.75 in.	17 x 8.5 x 1.75 in.
Weight	0.3 lb.	.	1.5 lb.	1.6 lb.	6 lb.	6.2 lb.	8 lb.

The following section has all the information you need to quickly and successfully purchase or build cables to the Secure Console Port Server SSH. It focuses on information related to the RS-232 interface, which applies not only to the Secure Console Port Server SSH but also to any RS-232 cabling. At the end of this chapter you will also find some information about the RS-485 interface, which is available for the Secure Console Port Server SSH 1-Port models only.

Appendix B - Cabling, Hardware, & Electrical

The RS-232 Standard

RS-232C, EIA RS-232, or simply RS-232 refer to a standard defined by the Electronic Industries Association in 1969 for serial communication. More than 30 years later, more applications have been found for this standard than its creators could have imagined. Almost all electronic devices nowadays have serial communication ports.

RS-232 was defined to connect Data Terminal Equipment, (DTE, usually a computer or terminal) to Data Communication Equipment (DCE, usually a modem):

```
DTE > RS-232 > DCE > communication line > DCE > RS-232 > DTE
```

RS-232 is now mostly being used to connect DTE devices directly (without modems or communication lines in between). While that was not the original intention, it is possible with some wiring tricks. The relevant signals (or wires) in a RS-232 cable, from the standpoint of the computer (DTE), are:

Receive Data (RxD) and Transmit Data (TxD)	The actual data signals
Signal Ground (Gnd)	Electrical reference for both ends
Data Terminal Ready (DTR)	Indicates that the computer (DTE) is active
Data Set Ready (DSR)	Indicates that the modem (DCE) is active.
Data Carrier Ready (DCD)	Indicates that the connection over the communication line is active
CTS (Clear to Send, an input)	Flow control for data flowing from DTE to DCE
RTS (Request to Send, an output)	Flow control for data flowing from DCE to DTE

Not all signals are necessary for every application, so the RS-232 cable may not need all 7 wires. The RS-232 interface defines communication parameters such as parity, number of bits per character, number of stop-bits and the baud rate. Both sides must be configured with the same parameters. That is the first thing to verify if you think you have the correct cable and things still do not work. The most common configuration is 8N1 (8 bits of data per character, no parity bit included with the data, 1 stop-bit to indicate the end of a character). The baud rate in a RS-232 line translates directly into the data speed in bits per second (bps). Usual transmission speeds range between 9,600 bps and 19,200bps (used in most automation and console applications) to 115,200 bps (used by the fastest modems).

Cable Length

The original RS-232 specifications were defined to work at a maximum speed of 19,200 bps over distances up to 15 meters (or about 50 feet). That was 30 years ago. Today, RS-232 interfaces can drive signals faster and through longer cables.
As a general rule, consider:

- If the speed is lower than 38.4 kbps, you are safe with any cable up to 30 meters (100 feet)
- If the speed is 38.4 kbps or higher, cables should be shorter than 10 meters (30 feet)
- If your application is outside the above limits (high speed, long distances), you will need better quality (low impedance, low-capacitance) cables.

Successful RS-232 data transmission depends on many variables that are specific to each environment. The general rules above are empirical and have a lot of safety margins built-in.

Appendix B - Cabling, Hardware, & Electrical

Connectors

The connector traditionally used with RS-232 is the 25-pin D-shaped connector (DB-25). Most analog modems and most older computers and serial equipment use this connector. The RS-232 interface on DB-25 connector always uses the same standard pin assignment.

The 9-pin D-shaped connector (DB-9) saves some space and is also used for RS-232. Most new PC COM ports and serial equipment (specially when compact size is important) uses this connector. RS-232 interfaces on DB-9 connectors always use the same standard pin assignment.

The telephone-type modular RJ-45 plug and jack are very compact, inexpensive and compatible with the phone and Ethernet wiring systems present in most buildings and data centers. Most networking equipment and new servers use RJ-45 connectors for serial communication. Unfortunately there is no standard RS-232 pin assignment for RJ-45 connectors. Every equipment vendor has its own pin assignment.

Most connectors have two versions. The ones with pins are said to be “male” and the ones with holes are said to be “female.”

Table 23: Cables and their pin specifications

RS-232 Signal	Name/Function (Input/Output)	DB-25 pins (Standard)	DB-9 pins (Standard)	RJ-45 pins (Black Box)
Chassis	Safety Ground	1	Shell	Shell
TxD	Transmit Data (O)	2	3	3
RxD	Receive Data (I)	3	2	6
DTR	Data Terminal Ready (O)	20	4	2
DSR	Data Set Ready (I)	6	6	8
DCD	Data Carrier Detect (I)	8	1	7
RTS	Request To Send (O)	4	7	1
CTS	Clear To Send (I)	5	8	5
Gnd	Signal Ground	7	5	4

Appendix B - Cabling, Hardware, & Electrical

Straight-Through vs. Crossover Cables

The RS-232 interface was originally intended to connect a DTE (computer, printer and other serial devices) to a DCE (modem) using a straight-through cable (all signals on one side connecting to the corresponding signals on the other side one-to-one). By using some “cabling tricks,” we can use RS-232 to connect two DTEs as is the case in most modern applications.

A crossover (a.k.a. null-modem) cable is used to connect two DTEs directly, without modems or communication lines in between. The data signals between the two sides are transmitted and received and there are many variations on how the other control signals are wired. A “complete” crossover cable would connect TxD with RxD, DTR with DCD/DSR, and RTS with CTS on both sides. A “simplified” crossover cable would cross TxD and RxD and locally short-circuit DTR with DCD/DSR and RTS with CTS.

Which cable should be used?

First, look up the proper cable for your application in the table below. Next, purchase standard off-the-shelf cables from a computer store or cable vendor. For custom cables, refer to the cable diagrams to build your own cables or order them from Black Box or a cable vendor.

Table 24: Which cable to use

To Connect To	Use Cable
DCE DB-25 Female (standard) • Analog Modems • ISDN Terminal Adapters	Cable 1: RJ-45 to DB-25 M straight-through (Custom). This custom cable can be ordered from Black Box or other cable vendors. A sample is included with the product (“straight-through”).

Appendix B - Cabling, Hardware, & Electrical

Table 24: Which cable to use

To Connect To	Use Cable
DTE RJ-45 Black Box (custom) • All Black Box Console Ports	Cable 2: RJ-45 to RJ-45 crossover (custom). A sample is included with the product ("straight-through") This custom cable can be ordered from Black Box or other cable vendors using the provided wiring diagram.
DTE DB-25 to DB-9 Black Box (custom) • For the 1-Port	Cable 3: DB-9 Female to DB-25 Female crossover. This connects the 1-Port (serial port) to terminals, printers and other DTE RS-232 devices.

Cable Diagrams

Before using the following cable diagrams refer to the tables above to select the correct cable for your application. Sometimes, crossover cables are wired slightly differently depending on the application. A "complete" crossover cable would connect the TxD with RxD, DTR with DCD/DSR, and RTS with CTS across both sides. A "simplified" crossover cable would cross TxD and RxD and locally short-circuit DTR with DCD/DSR and RTS with CTS.

Most of the diagrams in this document show the "complete" version of the crossover cables, with support for modem control signals and hardware flow control. Applications that do not require such features have just to configure NO hardware flow control and NO DCD detection on their side. Both ends should have the same configuration for better use of the complete version of the cables.

These cables appear in Cable Package #1 and/or Cable Package #2. You may or may not find them in your box depending on which package you received.

Appendix B - Cabling, Hardware, & Electrical

Cable #1: Black Box RJ-45 to DB-25 Male, straight-through

Application: This cable connects Black Box products (serial ports) to modems and other DCE RS-232 devices. It is included in both Cable Package #1 and #2.

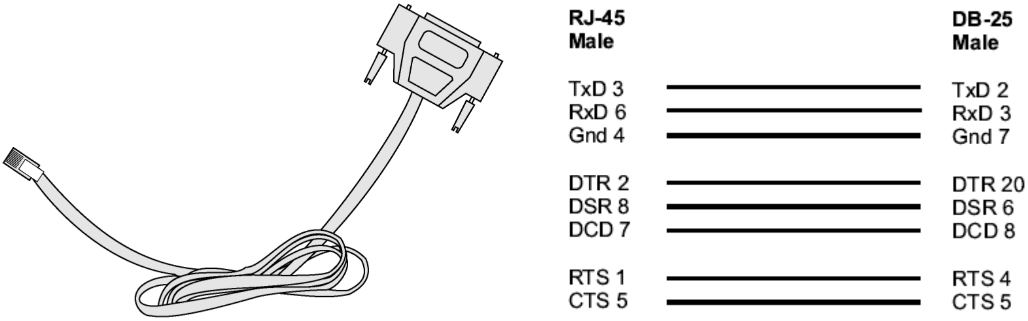


Figure 31: Cable 1 - Black Box RJ-45 to DB-25 Male, straight-through

Cable #2: Black Box RJ-45 to DB-25 Female/Male, crossover

This cable connects Black Box products (serial ports) to console ports, terminals, printers and other DTE RS-232 devices. If you are using Cable Package #1, after connecting the appropriate adapter to the RJ-45 straight-through cable, you will essentially have the cable shown in this picture. If you are using Cable Package #2, no assembly is required. You will have the cable shown below.

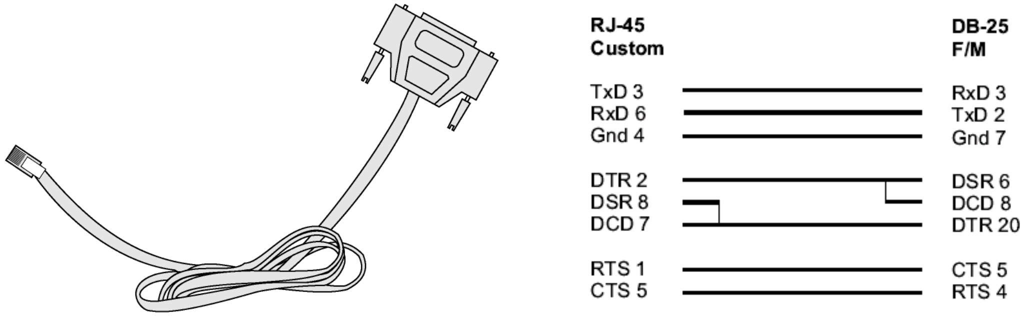


Figure 32: Cable 2 - Black Box RJ-45 to DB-25 Female/Male, crossover

Appendix B - Cabling, Hardware, & Electrical

Cable #3: Black Box RJ-45 to DB-9 Female, crossover

This cable connects Black Box products (serial ports) to console ports, terminals, printers and other DTE RS-232 devices. If you are using Cable Package #1, after connecting the appropriate adapter to the RJ-45 straight-through cable, you will essentially have the cable shown in this picture. If you are using Cable Package #2, no assembly is required. You will have the cable shown below.

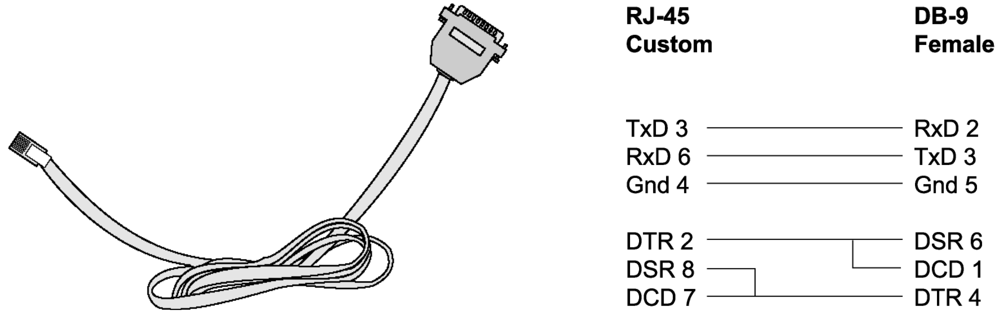


Figure 33: Cable 3 - Black Box RJ-45 to DB-9 Female, crossover

Cable #4: Black Box RJ-45 to Black Box RJ-45, straight-through

This cable is the main cable that you will use. Along with one of the adapters provided (RJ-45 to DB-9 or RJ-45 to DB-25) you can create a crossover cable like the ones explained in Cable #2 or #3 for configuration or to connect to a server. This cable is only included in Cable Package. #1.

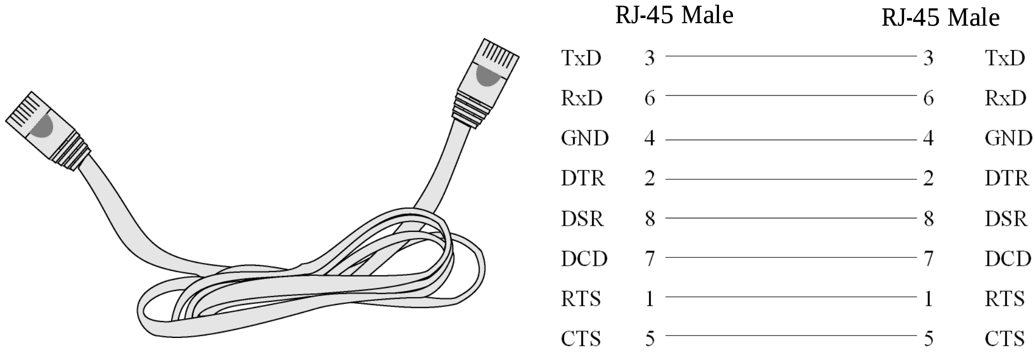


Figure 34: Cable 4 - Black Box RJ-45 to Black Box RJ-45, straight-through

Cable #5: Black Box/Sun Netra Cable

This Adapter attaches to a Cat 3 or Cat 5 network cable. It is usually used in console management applications to connect Black Box products to a Sun Netra server or to a Cisco product. This cable is included in Cable Package #2.

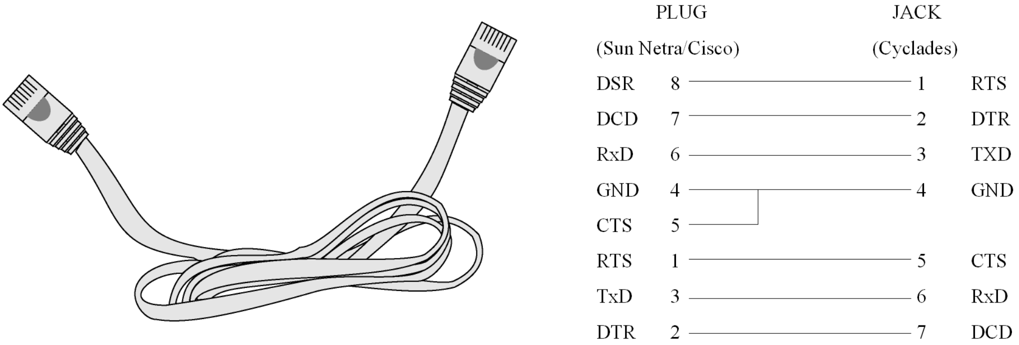


Figure 35: Cable 5 - Black Box/Sun Netra Cable

Adapters

The following four adapters are included in the product box. A general diagram is provided below and then a detailed description is included for each adapter.

Loop-Back Connector for Hardware Test

The use of the following DB-25 connector is explained in the Troubleshooting chapter. It is included in both Cable Package #1 and #2.

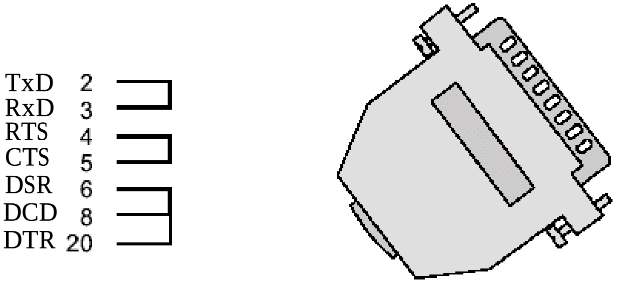
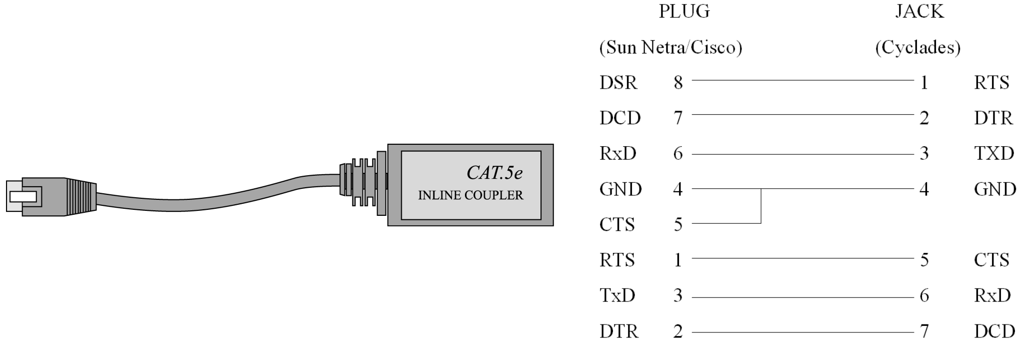


Figure 36: Loop-Back Connector

Appendix B - Cabling, Hardware, & Electrical

Black Box\Sun Netra Adapter

This Adapter attaches to a Cat 3 or Cat 5 network cable. It is usually used in console management applications to connect Black Box products to a Sun Netra server or to a Cisco product. At one end of the adapter is the black CAT.5e Inline Coupler box with a female RJ-45 terminus, from which a 3-inch-long black Sun Netra-labeled cord extends, terminating in an RJ-45 male connector. This adapter is included in Cable Package #2.

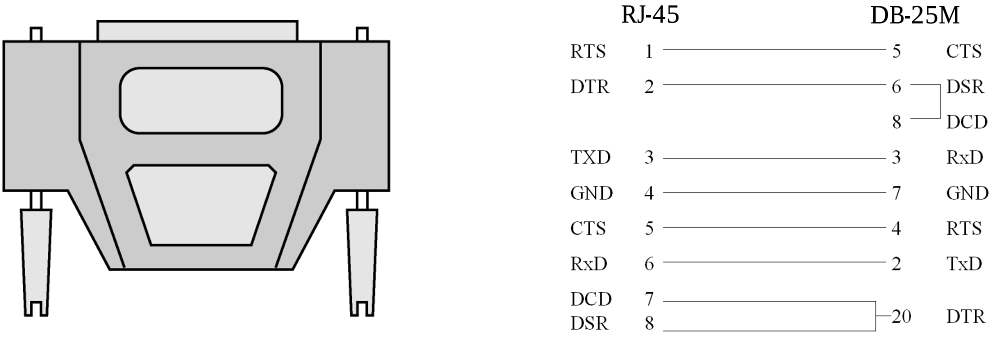


PLUG (Sun Netra/Cisco)		JACK (Cyclades)	
DSR	8	1	RTS
DCD	7	2	DTR
RxD	6	3	TXD
GND	4	4	GND
CTS	5		
RTS	1	5	CTS
TxD	3	6	RxD
DTR	2	7	DCD

Figure 37: Black Box\Sun Netra Adapter

RJ-45 Female to DB-25 Male Adapter

The following adapter may be necessary. It is included in Cable Package #1.



RJ-45		DB-25M	
RTS	1	5	CTS
DTR	2	6	DSR
		8	DCD
TXD	3	3	RxD
GND	4	7	GND
CTS	5	4	RTS
RxD	6	2	TxD
DCD	7	20	DTR
DSR	8		

Figure 38: RJ-45 Female to DB-25 Male Adapter

Appendix B - Cabling, Hardware, & Electrical

RJ-45 Female to DB-25 Female Adapter

The following adapter may be necessary. It is included in Cable Package #1.

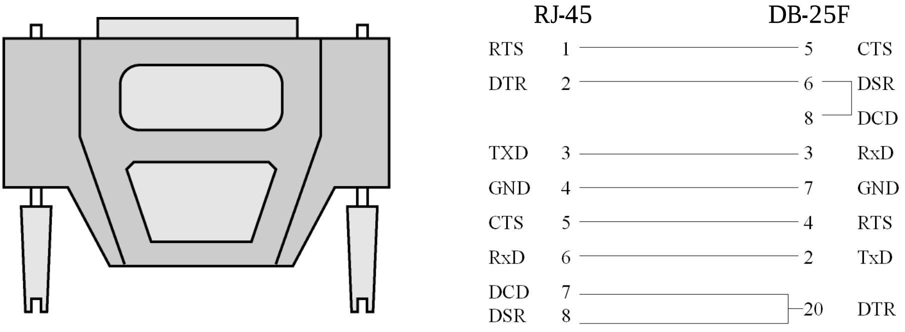


Figure 39: RJ-45 Female to DB-25 Female Adapter

RJ-45 Female to DB-9 Female Adapter

The following adapter may be necessary. This is included in Cable Package #1.

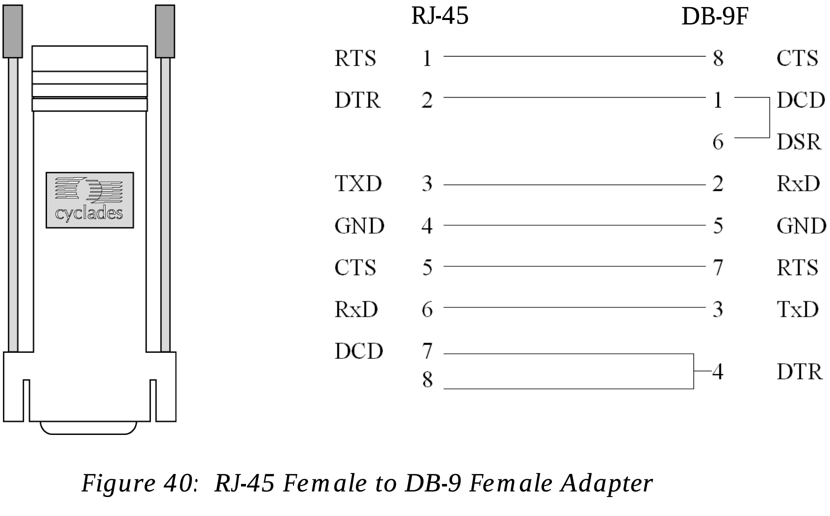


Figure 40: RJ-45 Female to DB-9 Female Adapter

Appendix B - Cabling, Hardware, & Electrical

Secure Console Port Server SSH 1-Port-only Cabling Information

The RS-485 Standard

The RS-485 is another standard for serial communication and is available only in the Secure Console Port Server SSH 1-Port. Different from the RS-232, the RS-485 uses fewer wires - either two wires (one twisted pair) for half duplex communication or four wires (two twisted pairs) for full duplex communication. Another RS-485 characteristic is the “termination.” In a network that uses the RS-485 standard, the equipment is connected one to the other in a cascade arrangement. A “termination” is required from the last equipment to set the end of this network.

Secure Console Port Server SSH 1-Port Connectors

Although the RS-485 can be provided in different kinds of connectors, the Secure Console Port Server SSH 1-Port uses a 9-pin D-shaped connector (DB-9) and a Terminal Block with the pin assignment described below.

Table 25: Secure Console Port Server SSH 1-Port Connector pin assignment

RS-485 Signal	Name/Function	DB-9 pins	Terminal Block pins
PW+		Not connected	1
TXD-	Transmit Data - (A)	7	2
TXD+	Transmit Data + (B)	3	3
RXD+	Receive Data + (B)	2	4
RXD-	Receive Data - (A)	8	5
PW-		Not connected	6

Appendix B - Cabling, Hardware, & Electrical

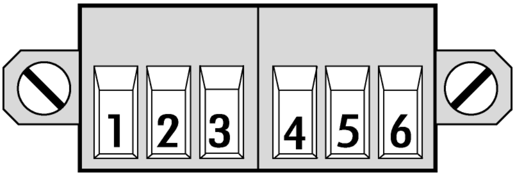


Figure 41: Terminal Block Pins

Notice that if the Secure Console Port Server SSH 1-Port is configured to use RS-485, the RS-485 signals will be available in both DB-9 and Terminal Block. In this case, the DB-9 pins used in an RS-232 connection can be considered not connected.

Cable Diagrams

Cable #1: DB-9 Female to DB-9 Female, crossover half duplex

Application: It connects the Secure Console Port Server SSH 1-Port (serial port) DTE RS-485 devices with half duplex communication.

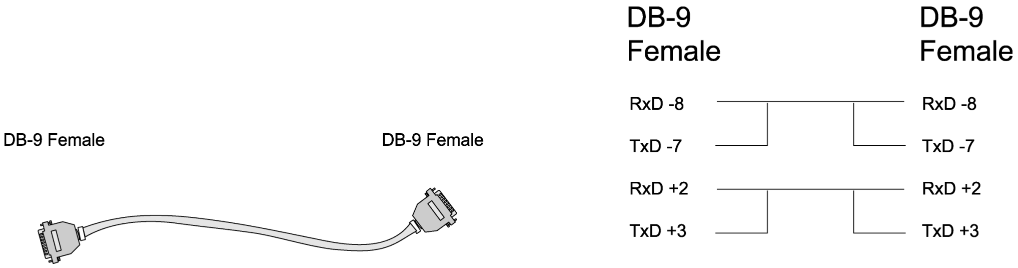


Figure 42: Cable 1 for the Secure Console Port Server SSH 1-Port - DB-9 Female to DB-9 Female, crossover half duplex

Cable #2: DB-9 Female to DB-9 Female, crossover full duplex

Application: It connects the Secure Console Port Server SSH 1-Port (serial port) to DTE RS-485 devices with full duplex communication.

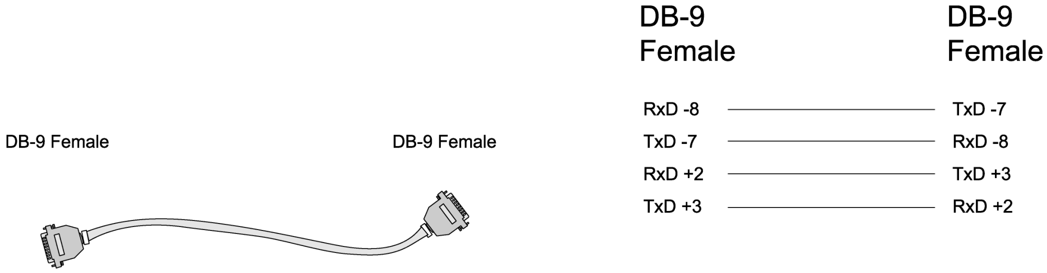


Figure 43: Cable 2 for the Secure Console Port Server SSH 1-Port - DB-9 Female to DB-9 Female, crossover full duplex

Cable #3: Terminal Block to Terminal Block, crossover half duplex

Application: It connects the Secure Console Port Server SSH 1-Port (serial port) to DTE RS-485 devices with half duplex communication.

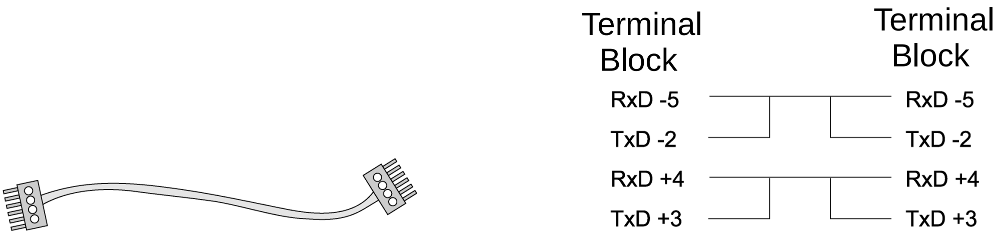


Figure 44: Cable 2 for the Secure Console Port Server SSH 1-Port - Terminal Block to Terminal Block, crossover half duplex

Appendix B - Cabling, Hardware, & Electrical

Cable #4: Terminal Block to Terminal Block, crossover full duplex

Application: It connects the Secure Console Port Server SSH 1-Port (serial port) to DTE RS-485 devices with full duplex communication.

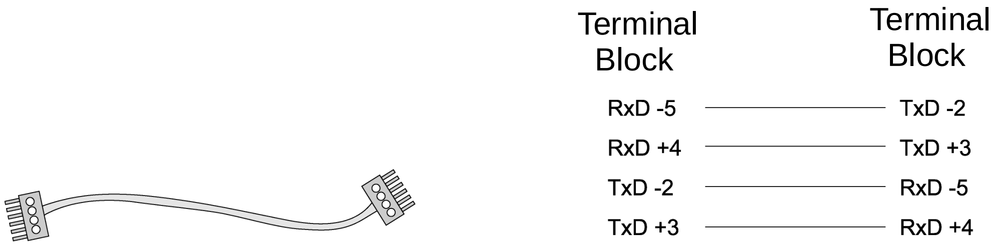


Figure 45: Cable 4 for the Secure Console Port Server SSH 1-Port - Terminal Block to Terminal Block, crossover full duplex

Cable #5: DB-9 Female to DB-25 Female, crossover

This cable connects the Secure Console Port Server SSH 1-Port to console ports, terminals, printers and other DTE RS-232 devices. You will essentially have the cable shown in this picture:

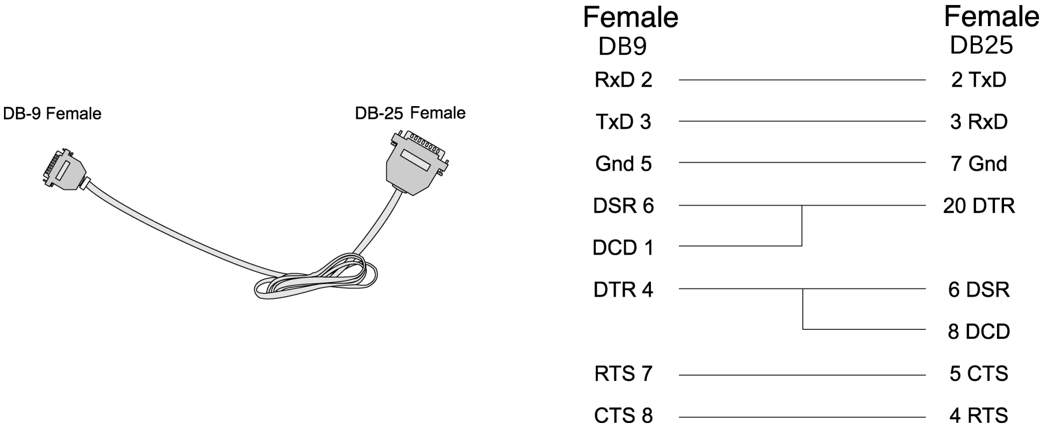


Figure 46: Cable 5 for the Secure Console Port Server SSH 1-Port - *DB-9 Female to DB-25 Female, crossover*

Appendix C - The pslave Configuration File

Introduction

This chapter begins with a table containing parameters common to all profiles, followed by tables with parameters specific to a certain profile. You can find samples of the pslave configuration files (pslave.conf, .cas, .ts, and .ras) in the /etc/portslave directory in the Secure Console Port Server SSH box.

Configuration Parameters

CAS, TS, and Dial-in Common Parameters

The parameters on the following table are common to all three profiles:

Table 26: Parameters Common to CAS, TS, & Dial-in Access

Parameter	Description	Value for this Example
conf.dhcp_client	It defines the dhcp client operation mode. Valid values: 0 - DHCP disabled 1 - DHCP active 2 - DHCP active and the unit saves in flash the last IP assigned by the DHCP server (default).	1 Also see Description column .
conf.eth_ip_alias	Secondary IP address for the Ethernet interface (needed for clustering feature).	209.81.55.10
conf.eth_mask_alias	Mask for the secondary IP address above.	255.255.255.0
conf.rlogin	It defines the location of rlogin utility *Note: This is a parameter specific to TS profile.*	Ex: /bin/rlogin

Table 26: Parameters Common to CAS, TS, & Dial-in Access

Parameter	Description	Value for this Example
conf.facility	The local facility sent to syslog-ng from PortSlave.	1 - 7
conf.group	Used to group users to simplify the configuration of the parameter all.users later on. This parameter can be used to define more than one group.	group_name: user1, user2
conf.eth_ip	Configured in Task 4: Edit the pslave.conf file in Chapter 2 - Installation, Configuration, and Usage. This is the IP address of the Ethernet interface. This parameter, along with the next two, is used by the cy_ras program to OVERWRITE the file /etc/network/ifcfg_eth0 as soon as the command “signal_ras hup” is executed. The file /etc/network/ifcfg_eth0 should not be edited by the user unless the cy_ras configuration is not going to be used.	200.200.200.1
conf.eth_mask	The mask for the Ethernet network.	255.255.255.0
conf.eth_mtu	The Maximum Transmission Unit size, which determines whether or not packets should be broken up.	1500
conf.lockdir	The lock directory, which is /var/lock for the Secure Console Port Server SSH. It should not be changed unless the user decides to customize the operating system.	/var/lock

Table 26: Parameters Common to CAS, TS, & Dial-in Access

Parameter	Description	Value for this Example
all.dcd	DCD signal (sets the tty parameter CLOCAL). Valid values are 0 or 1. If all.dcd=0, a connection request will be accepted regardless of the DCD signal and the connection will not be closed if the DCD signal is set to DOWN. If all.dcd=1 a connection request will be accepted only if the DCD signal is UP and the connection will be closed if the DCD signal is set to DOWN.	0
all.users	Restricts access to ports by user name (only the users listed can access the port or, using the character “!”, all but the users listed can access the port.) In this example, the users joe, mark and members of user_group cannot access the port. A single comma and spaces/tabs may be used between names. A comma may not appear between the “!” and the first user name. The users may be local, Radius or TacacsPlus. User groups (defined with the parameter conf.group) can be used in combination with user names in the parameter list. Notice that these are common users, not administrators.	! joe, mark, user_group

Table 26: Parameters Common to CAS, TS, & Dial-in Access

Parameter	Description	Value for this Example

all.issue	This text determines the format of the login banner that is issued when a connection is made to the
	Secure Console Port Server SSH. \n represents a new line and \r represents a carriage return. Expansion characters can be used here.
	Value for this Example:
	`\r\n\`
	`Welcome to terminal server %h port S%p \r\n\`
		See Description column

all.prompt	This text defines the format of the login prompt. Expansion characters can be used here.	%h login:
all.media	It defines media type RS232/RS484 and operation mode half/full duplex. *Valid values for all products* : rs232 - RS232 (default value). rs232_half - RS232 with RTS legcy half duplex rs232_half_cts - RS232 with RTS legacy half duplex and CTS control *valid values for the 1-Port only* : rs485_half - RS485 half duplex with out terminator rs485_half_terminator - RS485 half duplex with terminator rs485_full_terminator - RS485 full duplex with terminator rs422 - alike rs485_full_terminator	See Description column

Table 26: Parameters Common to CAS, TS, & Dial-in Access

Parameter	Description	Value for this Example
all.netmask	It defines the network mask for the serial port.	255.255.255. 255
all.mtu	It defines the maximum transmit unit	1500
all.mru	It defines the maximum receive unit	1500
all.sysutmp	It defines whether portslave must write login records.	yes/no
all.syswtmp	It defines whether portslave must write login records.	yes/no
all.pmtype	Name of the IPDU manufacturer.	cyclades
all.pmusers	List of the outlets each user can access.	1-3
all.pmkey	The hotkey that identifies the power management command.	^p
all.pmNumOfOutlets	The number of outlets you have on the AlterPath PM.	8

Table 26: Parameters Common to CAS, TS, & Dial-in Access

Parameter	Description	Value for this Example
all.sttyCmd	The TTY is programmed to work as configured and this user-specific configuration is applied over that serial port. Parameters must be separated by a space. The following example sets : *-igncr* This tells the terminal not to ignore the carriage-return on input, *-onlcr* Do not map newline character to a carriage return or newline character sequence on output, *opost* Post-process output, *-icrnl* Do not map carriage-return to a newline character on input. `all.sttyCmd -igncr -onlcr opost -icrnl`	commented
all.utmpfrom	It allow the administrator to customize the field "FROM" in the login records (utmp file). It is displayed in the "w" command. Ex: "%g:%P.%3.%4" %g : process id %P : Protocol %3 : Third nibble of remote IP %J : Remote IP Note: In the pslave.conf file there is a list of all expansion variables available.	See Description Column

Table 26: Parameters Common to CAS, TS, & Dial-in Access

Parameter	Description	Value for this Example
all.radnullpass	It defines whether the access to users with null password in the radius server must be granted or not.	yes/no
all.speed	The speed for all ports.	9600
all.datasize	The data size for all ports.	8
all.stopbits	The number of stop bits for all ports.	1
all.parity	The parity for all ports.	none
all.authhost1	This address indicates the location of the Radius/TacacsPlus authentication server and is only necessary if this option is chosen in the previous parameter. A second Radius/ TacacsPlus authentication server can be configured with the parameter all.authhost2.	200.200.200.2
all.accthost1	This address indicates the location of the Radius/TacacsPlus accounting server, which can be used to track how long users are connected after being authorized by the authentication server. Its use is optional. If this parameter is not used, accounting will not be performed. If the same server is used for authentication and accounting, both parameters must be filled with the same address. A second Radius/ TacacsPlus accounting server can be configured with the parameter all.accthost2.	200.200.200.2

Table 26: Parameters Common to CAS, TS, & Dial-in Access

Parameter	Description	Value for this Example
all.authtype	Configured in Task 4: Edit the pslave.conf file in Chapter 2 - Installation, Configuration, and Usage. Type of authentication used. There are several authentication type options: • *none* (no authentication) • *local* (authentication is performed using the /etc/passwd file) • *remote* (This is for a terminal profile only. The unit takes in a username but does not use it for authentication. Instead it passes it to the remote server where it is then used for authentication.) • *radius* (authentication is performed using a Radius authentication server) • *TacacsPlus* (authentication is performed using a TacacsPlus authentication server) • *ldap* (authentication is performed against an ldap database using an ldap server. The IP address and other details of the ldap server are defined in the file /etc/ldap.conf)	local

Table 26: Parameters Common to CAS, TS, & Dial-in Access

Parameter	Description	Value for this Example
	• *local/radius* (authentication is performed locally first, switching to Radius if unsuccessful) • *radius/local* (the opposite of the previous option) • *local/TacacsPlus* (authentication is performed locally first, switching to TacacsPlus if unsuccessful) • *TacacsPlus/local* (the opposite of the previous option) • *RadiusDownLocal* (local authentication is tried only when the Radius server is down) • *TacacsPlusDownLocal* (local authentication is tried only when the TacacsPlus server is down) Note that this parameter controls the authentication required by the Secure Console Port Server SSH. The authentication required by the device to which the user is connecting is controlled separately.	
all.radtimeout	This is the timeout (in seconds) for a Radius/ TacacsPlus authentication query to be answered. The first server (authhost1) is tried “radretries” times, and then the second (authhost2), if configured, is contacted “radretries” times. If the second also fails to respond, Radius/ TacacsPlus authentication fails.	3

Table 26: Parameters Common to CAS, TS, & Dial-in Access

Parameter	Description	Value for this Example
all.radretries	Defines the number of times each Radius/ TacacsPlus server is tried before another is contacted. The default, if not configured, is 5.	5
all.secret	This is the shared secret necessary for communication between the Secure Console Port Server SSH and the Radius/ TacacsPlus servers.	secret
all.flow	This sets the flow control to hardware, software, or none.	hard
all.protocol	The default CAS setup was explained in Chapter 2, Task 4: Edit the pslave.conf file. The TS configuration settings are in Table 28, "TS Parameters," on page 299. The Dial-in configuration settings are in Table 29, "Dial-in configuration Parameters," on page 301. For Power Management, see the section "Appendix J - Power Management" on page 451.	socket_server
sX.pmoutlet	sX indicates the serial port number to which the PM hardware is connected. The pmoutlet part of the parameter indicates the outlet # on the PM hardware that manages the server/network equipment in question.	8
s1.tty	The device name for the port is set to the value given in this parameter. If a device name is not provided for a port, it will not function.	ttyS1

Appendix C - The pslave Configuration File

CAS Parameters

You can configure additional CAS features with the parameters given on the following tables. (The Figure 56: CAS diagram with various authentication methods is used as an example in some parameters.

In addition to the above parameters which are common to all local and remote access scenarios, you can also configure the following parameters for additional options. Many of the parameters are unique to CAS, but some also apply to TS and Dial-in port profiles. This is indicated in these instances.

Table 27: Mostly CAS-specific Parameters

Parameter	Description	Value for this Example
conf.nfs_data_ buffering	This is the Remote Network File System where data captured from the serial port will be written instead of being written to the local directory */var/run/ DB*. The directory tree to which the file will be written must be NFS-mounted, so the remote host must have NFS installed and the administrator must create, export and allow reading/writing to this directory. The size of this file is not limited by the value of the parameter all.data_buffering, though the value cannot be zero since a zero value turns off data buffering. The size of the file is dependent on the NFS server only (hard drive, partition size, etc.).	commented
conf.DB_facility	This value (0-7) is the Local facility sent to the syslog with the data when syslog_buffering is active. The file /etc/syslog-ng/syslog-ng.conf contains a mapping between the facility number and the action (see more on Syslog in Chapter 3).	0

Table 27: Mostly CAS-specific Parameters

Parameter	Description	Value for this Example
all.ipno	This is the default IP address of the 's serial ports. The “+” indicates that the first port should be addressed as 192.168.1.101 and the following ports should have consecutive values. Any host can access a port using its IP address as long as a path to the address exists in the host's routing table.	192.168.170.101+
all.netmask	It defines the network mask for the serial port.	255.255.255.255
all.DTR_reset	This parameter specifies the behavior of the DTR signal in the serial port. If set to zero the DTR signal will be ON if there is a connection to the serial port, otherwise OFF. If set from 1 to 99 the DTR signal will be always ON. A value greater or equal 100 specifies for how long (in milliseconds) the DTR signal will be turned off before it is turned back on again when a connection to the serial port is closed.	100
all.break_sequence	This parameter is the string that is used to send a break to the TTY. It is only valid if TTY protocol is socket_ssh or socket_server.	~break
all.break_interval	This parameter defines the break duration in miliseconds. It is valid if TTY protocol is socket_ssh,	socket_server or ssh-2 (client)
all.modbus_smode	Communication mode through the serial ports. This parameter is meaningful only when modbus protocol is configured. The valid options are ascii (normal TX/RX mode) and rtu (some time constraints are observed between characters while transmitting a frame). If not configured, ASCII mode will be assumed.	commented

Table 27: Mostly CAS-specific Parameters

Parameter	Description	Value for this Example
all.lf_suppress	This can be useful because telneting (from DOS) from some OS such as Windows 98 causes produces an extra line feed so two prompts appear whenever you press Enter. When set to 1, line feed suppression is active which will eliminate the extra prompt. When set to 0 (default), line feed suppression is not active.	0
all.auto_ answer_input	This parameter works in conjunction with all.auto_answer_output. It allows you to configure a string that will be matched against all data coming in from the tty (remote server). If there is a match, the configured output string (auto_answer_output) will then be send back to the tty. This parameter works only when there is no session to the port. If uncommented and a string of bytes is set, matching occurs whenever there is not session established to the port. If this parameter is commented out, then no checking and matching occurs. (See more on the usage of this parameter in Terminal Appearance in Chapter 3.)	commented

Table 27: Mostly CAS-specific Parameters

Parameter	Description	Value for this Example
all.auto_ answer_output	This parameter works in conjunction with all.auto_answer_input. It allows you to configure a string that is sent back to the remote server whenever the incoming data remote server matches with all.auto_answer_input. This parameter works only when there is no session to the port. If this parameter is commented, then nothing will be sent back to the remote server even if all.auto_answer_input is uncommented. If this parameter is uncommented and if all.auto_answer_input is also uncommented, then the string configured will be sent back to the remote server. (See more on the usage of this parameter in Terminal Appearance in Chapter 3.)	commented
all.poll_interval	Valid only for protocols socket_server and raw_data. When not set to zero, this parameter sets the wait for a TCP connection keep-alive timer. If no traffic passes through the Secure Console Port Server SSH for this period of time, the Secure Console Port Server SSH will send a line status message to the remote device to see if the connection is still up. If not configured, 1000 ms is assumed (the unit for this parameter is ms). If set to zero, line status messages will not be sent to the socket client.	0

Table 27: Mostly CAS-specific Parameters

Parameter	Description	Value for this Example
all.socket_port	In the CAS profile, this defines an alternative labeling system for the Secure Console Port Server SSH ports. The “+” after the numerical value causes the serial interfaces to be numbered consecutively. In this example, serial interface 1 is assigned the port value 7001,serial interface 2 is assigned the port value 7002, etc. One example on how this could be used is in the case of all.protocol or s<n>.protocol socket_ssh and the port value (7001, 7002, etc), if supplied by the ssh client like username:port value, the ssh client will be directly connected with the serial interface. For TS, this parameter is valid only all.protocol is configured as socket_cliente or telnet. It is the TCP port number of the application that will accept connection requested by this serial port.	7001+

Table 27: Mostly CAS-specific Parameters

Parameter	Description	Value for this Example
all.data_ buffering	A non zero value activates data buffering (local or remote, according to what was configured in the parameter conf.nfs_data_buffering see Data Buffering in Chapter 3). If local data buffering, a file is created on the Secure Console Port Server SSH; if remote, a file is created through NFS in a remote server. All data received from the port is captured in this file. If local data buffering, this parameter means the maximum file size (in bytes). If remote, this parameter is just a flag to activate (greater than zero) or deactivate data buffering. When local data buffering is used, each time the maximum is reached the oldest 10% of stored data is discarded, releasing space for new data (FIFO system) - circular file. When remote data buffering is used, there's no maximum file size other than the one imposed by the remote server - linear file. This file can be viewed using the normal Unix tools (cat, vi, more, etc.). *Size is in bytes not kilobytes*. See Data Buffering for details.	0

Table 27: Mostly CAS-specific Parameters

Parameter	Description	Value for this Example
all.DB_mode	When configured as cir for circular format, the buffer works like a revolving file at all times. The file is overwritten whenever the limit of the buffer size (as configured in all.data_buffering or s<n>.data_buffering) is reached. As for linear format (lin), once the limit of the kernel buffer size is reached (4k), a flow control stop (RTS off or XOFF-depending on how all.f low or s<n>.flow is set) is issued automatically to the remote device so that it will stop sending data to the serial port. Then, when a session is established to the serial port, the data in the buffer is shown to the user if not empty (dont_show_DBmenu parameter assumed to be 2), cleared, and a flow control start (RTS on or XON) is issued to resume data transmission. Once exiting the session, linear data buffering resumes. If all.flow or s<n>.flow is set to none, linear buffering is not possible as there is no way to stop reception through the serial line. Default is cir.	cir
all.DB_ timestamp	Records the time stamp in the data buffering file (1) or not (0). If it is configured as 1, the software will accumulate input characters until it receives a CR and LF from the serial port or the accumulated data reaches 256 characters. Either way, the accumulated data will be recorded in the data buffering file along with the current time. The parameter all.data_buffering has to be with a non-zero value for this parameter to be meaningful.	0

Table 27: Mostly CAS-specific Parameters

Parameter	Description	Value for this Example
all.syslog_ buffering	When non zero, the contents of the data buffer are sent to the syslogng every time a quantity of data equal to this parameter is collected. The syslog level for data buffering is hard coded to level 5 (notice) and facility local[0+conf.DB_facility]. The file /etc/syslog-ng/syslog-ng.conf should be set accordingly for the syslog-ng to take some action. (See Syslog-ng Configuration to use with Syslog Buffering Feature.)	0
all.syslog_sess	Syslog_buffering must be activated for the following to work. When 0, syslog messages are always generated whether or not there is a session to the port sending data to the unit. When 1, syslog messages are NOT generated when there IS a session to the port sending data to the unit, but resumes generation of syslog messages when there ISN'T a session to the port.	0
all.dont_show_ DBmenu	When zero, a menu with data buffering options is shown when a nonempty data buffering file is found. When 1, the data buffering menu is not shown. When 2, the data buffering menu is not shown but the data buffering file is shown if not empty. When 3, the data buffering menu is shown, but without the erase and show and erase options.	1

Table 27: Mostly CAS-specific Parameters

Parameter	Description	Value for this Example
all.alarm	When non zero, all data received from the port are captured and sent to syslog-ng with level INFO and local[0+conf.DB_facility]facility. The syslogng.conf file should be set accordingly, for the syslog-ng to take some action (please see Generating Alarms in Chapter 3 - Additional Features for the syslog-ng configuration file).	0
all.billing_ records	Billing file size configuration. A non-zero value defines the maximum number of billing records within a billing file. Zero stops billing recording. The billing files are located at /var/ run/DB and are named cycXXXXX-YYMMDD.hhmmss.txt (e.g., cycTS100-030122.153611.txt.	50
all.billing_ timeout	Billing timeout configuration. A non-zero value defines how long (minutes) a billing file should be waiting for records before close. After a file is closed, this file is available for transfer and a new one is opened. Zero means “no timeout” and so the file is only closed after “billing_records” are received.	60
all.billing_eor	Defines the character sequence that terminates each billing record. Any character sequence is valid, including '\r' or '^M' (carriage return), '\n' or '^J' (new line), etc..."	Default value: "\n"

Table 27: Mostly CAS-specific Parameters

Parameter	Description	Value for this Example
all.sniff_mode	This parameter determines what other users connected to the very same port (see parameter admin_users below) can see of the session of the first connected user (main session): *in* shows data written to the port, *out* shows data received from the port, and *i/o* shows both streams. The second and later sessions are called sniff sessions and this feature is activated whenever the protocol parameter is set to socket_ssh or socket_server.	out
all.admin_users	This parameter determines which users can receive the sniff session menu. Then they have options to open a sniff session or cancel a previous session. When users want access per port to be controlled by administrators, this parameter is obligatory and authtype must not be none. User groups (defined with the parameter conf.group) can be used in combination with user names in the parameter list.	peter, john, user_group
all.multiple_ sessions	Allows users to open more than one common and sniff session on the same port. The options are “yes,” “no,” “RW_session,” or “sniff_session.” Default is set to “no.” Please see Session Sniffing in Chapter 3 for details.	no
all.escape_char	This parameter determines which character must be typed to make the session enter “menu mode”. The possible values are <CTRL-a> to <CTRL-z>. Represent the CTRL with '^'. This parameter is only valid when the port protocol is socket_server or socket_ssh. Default value is '^z'.	^z

Table 27: Mostly CAS-specific Parameters

Parameter	Description	Value for this Example
all.tx_interval	Valid for protocols socket_server and raw_data. Defines the delay (in milliseconds) before transmission to the Ethernet of data received through a serial port. If not configured, 100ms is assumed. If set to zero or a value above 1000, no buffering will take place.	100
all.idletimeout	Specifies how long (in minutes) a connection can remain inactive before it is cut off. If it set to zero, the connection will not time out.	0
s1.serverfarm	Alias name given to the server connected to the serial port. Server_connected.	serial1
s2.tty	It defines the physical device name associated to the serial port (without the /dev/).	ttyS2
s8.tty	It defines the physical device name associated to the serial port (without the /dev/).	ttyS8

TS Parameters

The following parameters are unique to a TS setup except where indicated.

Table 28: TS Parameters

Parameter	Description	Value for this Example
conf.telnet	Location of the telnet utility	/usr/bin/telnet

Table 28: TS Parameters

Parameter	Description	Value for this Example
conf.ssh	Location of the ssh utility.	/bin/ssh
conf.locallogins	This parameter is only necessary when authentication is being performed for a port. When set to one, it is possible to log in to the Secure Console Port Server SSH directly by placing a “!” before your login name, then using your normal password. This is useful if the Radius authentication server is down.	0
all.host	The IP address of the host to which the terminals will connect.	200.200.200.3
all.term	This parameter defines the terminal type assumed when performing rlogin or telnet to other hosts.	vt100
all.userauto	Username used when connected to a UNIX server from the user’s serial terminal.	
all.protocol (for TS)	For the terminal server configuration, the possible protocols are login (which requests username and password), rlogin (receives username from the Secure Console Port Server SSH and requests a password), telnet, ssh, ssh2, or socket_client. See all.socket_port definition if all.protocol is configured as socket_client.	rlogin
all.socket_port	The socket_port is the TCP port number of the application that will accept connection requested by this serial port. That application usually is telnet (23).	

Table 28: TS Parameters

Parameter	Description	Value for this Example
all.telnet_client_ mode	When the protocol is TELNET, this parameter configured as BINARY (1) causes an attempt to negotiate the TELNET BINARY option on both input and output with the Telnet server. So it puts the telnet client in binary mode. The acceptable values are "0" or "1", where "0" is text mode (default) and "1" is a binary mode.	
s16.tty (TS)	It defines the physical device name associated to the serial port (without the /dev/).	ttyS16

Dial-in Access Parameters

The following parameters are unique to a Dial-in setup except where indicated.

Table 29: Dial-in configuration Parameters

Parameter	Description	Value for this Example
conf.pppd	Location of the ppp daemon with Radius.	/usr/local/sbin/ pppd
all.netmask	It defines the network mask for the serial port.	255.255.255.255
all.ipno (CAS and Dial-in)	See description in CAS section.	

Table 29: Dial-in configuration Parameters

Parameter	Description	Value for this Example
all.initchat	Modem initialization string.	TIMEOUT 10 "" \d\l\dATZ \ OK\r\n-ATZ-OK\r\n “” \ “” ATMO OK\R\N “” \ TIMEOUT 3600 RING “” \ STATUS Incoming %p:I.HANDSHAKE “” ATA\ TIMEOUT 60 CONNECT@ “” \ STATUS Connected %p:I.HANDSHAKE
all.autoppp	all.autoppp PPP options to auto-detect a ppp session. The cb-script parameter defines the file used for callback and enables negotiation with the callback server. Callback is available in combination with Radius Server authentication. When a registered user calls the Secure Console Port Server SSH, it will disconnect the user, then call the user back. The following three parameters must be configured in the Radius Server: attribute Service_type(6): Callback Framed; attribute Framed_Protocol(7): PPP; attribute Callback_Number(19): the dial number (example: 50903300).	%j novj \ proxyarp modem asyncmap 000A0000 \ noipx noccp login auth require-pap refusechap\ mtu %t mru %t \ cb-script /etc/portslave/cb_script \ plugin /usr/lib/libpsr.so

Table 29: Dial-in configuration Parameters

Parameter	Description	Value for this Example
all.pppopt	all.pppopt PPP options when user has already been authenticated.	%i:%j novj \ proxyarp modem asyncmap 000A0000 \ noipx noccp mtu %t mru %t netmask%m \ idle %I maxconnect %T \ plugin /usr/lib/libpsr.so
all.protocol	For the Dial-in configuration, the available protocols are PPP, SLIP and CSLIP.	ppp
s32.tty	See the s1.tty entry in the CAS section.	ttyS32

This page has been left intentionally blank.

Appendix D - Linux-PAM

Introduction

Linux-PAM (Pluggable Authentication Modules for Linux) is a suite of shared libraries that enable the local system administrator to choose how applications authenticate users. In other words, without (rewriting and) recompiling a PAM-aware application, it is possible to switch between the authentication mechanism(s) it uses. Indeed, one may entirely upgrade the local authentication system without touching the applications themselves.

It is the purpose of the Linux-PAM project to separate the development of privilege-granting software from the development of secure and appropriate authentication schemes. This is accomplished by providing a library of functions that an application may use to request that a user be authenticated. This PAM library is configured locally with a system file, /etc/pam.conf (or a series of configuration files located in /etc/pam.d/) to authenticate a user request via the locally available authentication modules. The modules themselves will usually be located in the directory /lib/security and take the form of dynamically loadable object files.

The Linux-PAM authentication mechanism gives to the system administrator the freedom to stipulate which authentication scheme is to be used. S/he has the freedom to set the scheme for any/all PAM-aware applications on your Linux system. That is, s/he can authenticate from anything as generous as simple trust (pam_permit) to something as severe as a combination of a retinal scan, a voice print and a one-time password!

Linux-PAM deals with four separate types of (management) task. These are: authentication management, account management, session management, and password management. The association of the preferred management scheme with the behavior of an application is made with entries in the relevant Linux-PAM configuration file. The management functions are performed by modules specified in the configuration file.

Following is a figure that describes the overall organization of Linux-PAM:

Appendix D - Linux-PAM

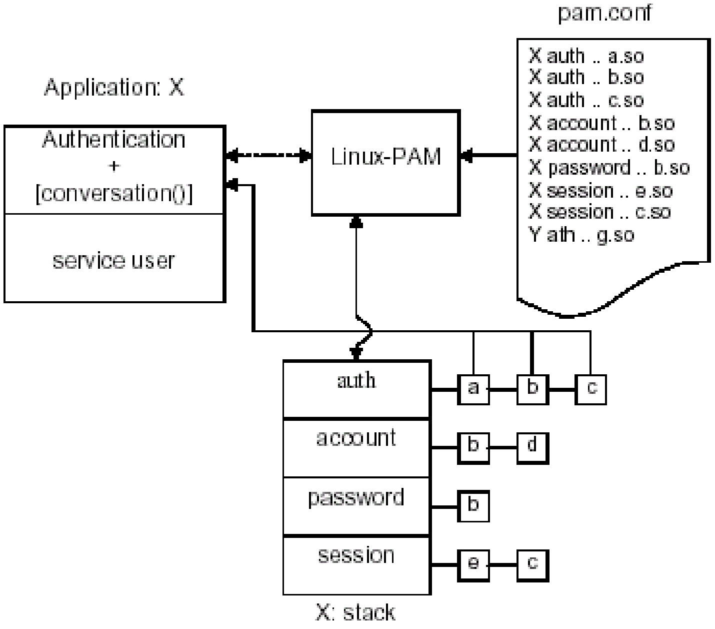


Figure 47: Data flow diagram of Linux-PAM

The left of the figure represents the application: Application X. Such an application interfaces with the Linux-PAM library and knows none of the specifics of its configured authentication method. The Linux-PAM library (in the center) consults the contents of the PAM configuration file and loads the modules that are appropriate for Application X. These modules fall into one of four management groups (lower center) and are stacked in the order they appear in the configuration file. These modules, when called by Linux-PAM, perform the various authentication tasks for the application. Textual information, required from or offered to the user can be exchanged through the use of the application-supplied conversation function.

Appendix D - Linux-PAM

The Linux-PAM Configuration File

Linux-PAM is designed to provide the system administrator with a great deal of flexibility in configuring the privilege-granting applications of their system. The local configuration of those aspects of system security controlled by Linux-PAM is contained in one of two places: either the single system file /etc/pam.conf or the /etc/pam.d/ directory. In this section we discuss the correct syntax of and generic options respected by entries to these files.

Configuration File Syntax

The reader should note that the Linux-PAM-specific tokens in this file are case-insensitive. The module paths, however, are case-sensitive since they indicate a file's name and reflect the case-dependence of typical Linux file systems. The case-sensitivity of the arguments to any given module is defined for each module in turn.

In addition to the lines described below, there are two special characters provided for the convenience of the system administrator:

#	Comments are preceded by this character and extend to the next end-of-line.
\	This character extends the configuration lines.

A general configuration line of the /etc/pam.conf file has the following form:

```
Service-name module-type control-flag module-path arguments
```

The meaning of each of these tokens is explained below. The second (and more recently adopted) way of configuring Linux-PAM is via the contents of the /etc/pam.d/ directory. After the meaning of the above tokens is explained, the method will be described.

Service-name The name of the service associated with this entry. Frequently the service name is the conventional name of the given application. For example, ‘ftpd’, ‘rlogind’, ‘su’, etc. There is a special service-name, reserved for defining a default authentication mechanism. It has the name ‘OTHER’ and may be specified in either lower or upper case characters. Note, when there is a module specified for a named service, the ‘OTHER’ entries are ignored.

Module-type One of (currently) the four types of module. The four types are as follows:

Auth- This module type provides two aspects of authenticating the user. First, it establishes that the user is who they claim to be, by instructing the application to prompt the user for a password or other means of identification. Second, the module can grant group membership, independently of the /etc/groups, or other privileges through its credential-granting properties.

Account- This module performs non-authentication-based account management. It is typically used to restrict or permit access to a service based on the time of day, currently available system resources (maximum number of users) or perhaps the location of the applicant user—‘root’ login only on the console.

Session- Primarily, this module is associated with doing things that need to be done for the user before or after they can be given service. Such things include the logging of information concerning the opening or closing of some data exchange with a user, mounting directories, etc.

Password- This last module type is required for updating the authentication token associated with the user. Typically, there is one module for each ‘challenge/response’ based authentication (auth) module-type.

Control-flag The control-flag is used to indicate how the PAM library will react to the success or failure of the module it is associated with. Since modules can be stacked (modules of the same type execute in series, one after another), the control-flags determine the relative importance of each module. The application is not made aware of the individual success or failure of modules listed in the '/etc/pam.conf' file. Instead, it receives a summary of success or fail responses from the Linux-PAM library. The order of execution of these modules is that of the entries in the /etc/pam.conf file: earlier entries are executed before later ones. The control-flag can be defined with one of two syntaxes. The simpler (and historical) syntax for the control-flag is a single keyword defined to indicate the severity of concern associated with the success or failure of a specific module. There are four such keywords: required, requisite, sufficient and optional.

The Linux-PAM library interprets these keywords in the following manner:

Required This indicates that the success of the module is required for the module-type facility to succeed. Failure of this module will not be apparent to the user until all of the remaining modules (of the same module-type) have been executed.

Requisite This is similar to *required*. However, in the case that such a module returns a failure, control is directly returned to the application. The return value is that associated with the first required or requisite module to fail. Note that this flag can be used to protect against the possibility of a user getting the opportunity to enter a password over an unsafe medium. It is conceivable that such behavior might inform an attacker of valid accounts on a system. This possibility should be weighed against the significant concerns of exposing a sensitive password in a hostile environment.

Sufficient The success of this module is deemed 'sufficient' to satisfy the Linux-PAM library that this moduletype has succeeded in its purpose. In the event that no previous required module has failed, no more 'stacked' modules of this type are invoked. (Note: in this case subsequent required modules are not invoked.) A failure of this module is not deemed as fatal to satisfying the application.

Optional As its name suggests, this control-flag marks the module as not being critical to the success or failure of the user's application for service. In general, Linux-PAM ignores such a module when determining if the module stack will succeed or fail. However, in the absence of any definite successes or failures of previous or subsequent stacked modules this module will determine the nature of the response to the application. One example of this latter case is when the other modules return something like PAM_IGNORE.

Newest Syntax

The more elaborate (newer) syntax is much more specific and gives the administrator a great deal of control over how the user is authenticated. This form of the control flag is delimited with square brackets and consists of a series of value=action tokens:

```
[value1=action1 value2=action2 ...]
```

Here, valueI is one of the following return values: success; open_err; symbol_err; service_err; system_err; buf_err; perm_denied; auth_err; cred_insufficient; authinfo_unavail; user_unknown; maxtries; new_authtok_reqd; acct_expired; session_err; cred_unavail; cred_expired; cred_err; no_module_data; conv_err; authtok_err; authtok_recover_err; authtok_lock_busy; authtok_disable_aging; try_again; ignore; abort; authtok_expired; module_unknown; bad_item; and default. The last of these (default) can be used to set the action for those return values that are not explicitly defined.

The action can be a positive integer or one of the following tokens: ignore, ok, done, bad, die, and reset.

A positive integer When specified as the action, can be used to indicate that the next J modules of the current type will be skipped. In this way, the administrator can develop a moderately sophisticated stack of modules with a number of different paths of execution. Which path is taken can be determined by the reactions of individual modules.

Ignore When used with a stack of modules, the module's return status will not contribute to the return code the application obtains.

Bad This action indicates that the return code should be thought of as indicative of the module failing. If this module is the first in the stack to fail, its status value will be used for that of the whole stack.

Die Equivalent to *bad* with the side effect of terminating the module stack and PAM immediately returning to the application.

OK This tells PAM that the administrator thinks this return code should contribute directly to the return code of the full stack of modules. In other words, if the former state of the stack would lead to a return of PAM_SUCCESS, the module's return code will override this value. Note: if the former state of the stack holds some value that is indicative of a module failure, this 'OK' value will not be used to override that value.

Done Equivalent to OK with the side-0effect of terminating the module stack and PAM immediately returning to the application.

Reset Clear all memory of the state of the module stack and start again with the next stacked module.

Module Path

Module Path is the path-name of the dynamically loadable object file--the pluggable module itself. If the first character of the module path is '/', it is assumed to be a complete path. If this is not the case, the given module path is appended to the default module path: /lib/security.

Currently, the has the following modules available:

pam_access Provides logdaemon style login access control.

pam_deny Deny access to all users.

Appendix D - Linux-PAM

pam_env This module allows the (un)setting of environment variables. The use of previously set environment variables as well as PAM_ITEMs such as PAM_RHOST is supported.

pam_filter This module was written to offer a plug-in alternative to programs like ttysnoop (XXX - need a reference). Since a filter that performs this function has not been written, it is currently only a toy. The single filter provided with the module simply transposes upper and lower case letters in the input and output streams. (This can be very annoying and is not kind to termcap-based editors.)

pam_group This module provides group settings based on the user's name and the terminal they are requesting a given service from. It takes note of the time of day.

pam_issue This module presents the issue file (/etc/issue by default) when prompting for a username.

pam_lastlog This session module maintains the /var/log/lastlog file. It adds an open entry when called via the pam_open_session()function and completes it when pam_close_session() is called. This module can also display a line of information about the last login of the user. If an application already performs these tasks, it is not necessary to use this module.

pam_limits This module, through the Linux-PAM open-session hook, sets limits on the system resources that can be obtained in a user session. Its actions are dictated more explicitly through the configuration file discussed in /etc/security/pam_limits.conf.

pam_listfile The listfile module provides a way to deny or allow services based on an arbitrary file.

pam_motd This module outputs the motd file (/etc/motd by default) upon successful login.

pam_nologin Provides standard Unix nologin authentication.

pam_permit This module should be used with extreme caution. Its action is to always permit access. It does nothing else.

pam_radius Provides Radius server authentication and accounting.

pam_rootok This module is for use in situations where the superuser wishes to gain access to a service without having to enter a password.

pam_securetty Provides standard UNIX securetty checking.

pam_time Running a well-regulated system occasionally involves restricting access to certain services in a selective manner. This module offers some time control for access to services offered by a system. Its actions are determined with a configuration file. This module can be configured to deny access to (individual) users based on their name, the time of day, the day of week, the service they are applying for and their terminal from which they are making their request.

pam_tacplus Provides TacacsPlus Server authentication, authorization (account management), and accounting (session management).

pam_unix This is the standard UNIX authentication module. It uses standard calls from the system’s libraries to retrieve and set account information as well as authentication. Usually this is obtained from the etc/passwd and the /etc/shadow file as well when shadow is enabled.

pam_warn This module is principally for logging information about a proposed authentication or application to update a password.

pam_ldap Pam_ldap looks for the ldap client configuration file “ldap.conf” in /etc/. Here's an example of the ldap.conf file (partial):

```
# file name: ldap.conf

# This is the configuration file for the LDAP
nameservice

# switch library and the LDAP PAM module.

#

# Your LDAP server. Must be resolvable without using
LDAP.

host 127.0.0.1

# The distinguished name of the search base.
base dc=padl,dc=com
```

Appendix D - Linux-PAM

Arguments

The arguments are a list of tokens that are passed to the module when it is invoked. They are much like arguments to a typical Linux shell command. Generally, valid arguments are optional and are specific to any given module. Invalid arguments are ignored by a module, however, when encountering an invalid argument, the module is required to write an error to syslog(3).

The following are optional arguments which are likely to be understood by any module. Arguments (including these) are in general, optional.

debug Use the syslog(3) call to log debugging information to the system log files.

no_warn Instruct module to not give warning messages to the application.

use_first_pass The module should not prompt the user for a password. Instead, it should obtain the previously typed password (from the preceding auth module), and use that. If that doesn't work, then the user will not be authenticated. (This option is intended for auth and password modules only).

try_first_pass The module should attempt authentication with the previously typed password (from the preceding auth module). If that doesn't work, then the user is prompted for a password. (This option is intended for auth modules only).

use_mapped_pass This argument is not currently supported by any of the modules in the Linux-PAM distribution because of possible consequences associated with U.S. encryption exporting restrictions.

expose_account In general, the leakage of some information about user accounts is not a secure policy for modules to adopt. Sometimes information such as user names or home directories, or preferred shell, can be used to attack a user's account. In some circumstances, however, this sort of information is not deemed a threat: displaying a user's full name when asking them for a password in a secured environment could- also be called being 'friendly'. The expose_account argument is a standard module argument to encourage a module to be less discrete about account information as deemed appropriate by the local administrator. Any line in (one of) the configuration file(s), that is not formatted correctly will generally tend (erring on the side of caution) to make the authentication process fail. A corresponding error is written to the system log files with a call to syslog(3).

Directory-based Configuration

It is possible to configure libpam via the contents of the /etc/ pam.d/ directory. This is more flexible than using the single configuration file. In this case, the directory is filled with files--each of which has a filename equal to a service-name (in lower-case)--the personal configuration file for the named service. The Secure Console Port Server SSH Linux-PAM was compiled to use both
/etc/pam.d/ and /etc/pam.conf in sequence. In this mode, entries in /etc/pam.d/ override those of /etc/pam.conf.

The syntax of each file in /etc/pam.d/ is similar to that of the /etc/pam.conf file and is made up of lines of the following form:

```
module-type control-flag module-path arguments
```

The only difference between the two is that the service-name is not present. The service-name is of course the name of the given configuration file. For example, /etc/pam.d/login contains the configuration for the login service.

Appendix D - Linux-PAM

Default Policy

If a system is to be considered secure, it had better have a reasonably secure ‘OTHER’ entry. The following is a “severe” setting (which is not a bad place to start!):

```
#

# default; deny access

#

OTHER auth required pam_deny.so

OTHER account required pam_deny.so

OTHER password required pam_deny.so

OTHER session required pam_deny.so
```

While fundamentally a secure default, this is not very sympathetic to a misconfigured system. For example, such a system is vulnerable to locking everyone out should the rest of the file become badly written.

The module pam_deny not very sophisticated. For example, it logs no information when it is invoked, so unless the users of a system contact the administrator when failing to execute a service application, the administrator may not know for a long while that his system is misconfigured.

The addition of the following line before those in the above example would provide a suitable warning to the administrator.

```
#

# default; wake up! This application is not configured

#

OTHER auth required pam_warn.so

OTHER password required pam_warn.so
```

Having two “OTHER auth” lines is an example of stacking.

Appendix D - Linux-PAM

On a system that uses the /etc/pam.d/ configuration, the corresponding default setup would be achieved with the following file:

```
#
# default configuration: /etc/pam.d/other
#
auth required pam_warn.so
auth required pam_deny.so
account required pam_deny.so
password required pam_warn.so
password required pam_deny.so
session required pam_deny.so
```

On a less sensitive computer, the following selection of lines (in /etc/pam.conf) is likely to mimic the historically familiar Linux setup:

```
#
# default; standard UNIX access
#
OTHER auth required pam_unix_auth.so
OTHER account required pam_unix_acct.so
OTHER password required pam_unix_passwd.so
OTHER session required pam_unix_session.so
```

In general this will provide a starting place for most applications.

In addition to the normal applications: login, su, sshd, passwd, and pppd. Black Box also has made portslave a PAM-aware application. The portslave requires four services configured in pam.conf. They are local, remote, radius, and tacplus. The portslave PAM interface takes any parameter needed to perform the authentication in the serial ports from the file pslave.conf. The pslave.conf parameter all.authtype determines which service(s) should be used.

Appendix D - Linux-PAM

```
# ---------------------------------------------------------------------------#
# /etc/pam.conf                                                              #
#                                                                            #
# Last modified by Andrew G. Morgan <morgan@kernel.org>                      #
# ---------------------------------------------------------------------------#
# $Id: pam.conf,v 1.9 2003/06/12 20:34:13 regina Exp $
# ---------------------------------------------------------------------------#
# serv.module   ctrl      module [path]...[args..]                  #
# nametype   flag                                                      #
# ---------------------------------------------------------------------------#

# WARNING. The services tacacs, s_tacacs, radius, s_radius, local, s_local,
#          and remote are used by the Cyclades applications portslave,
#          socket_server, socket_ssh, and raw_data and should not be changed
#          by the administrators unless he knows what he is doing.

#
# The PAM configuration file for the `kerberos' service
#
kerberosauthrequiredpam_krb5.so no_ccache
kerberos auth    optional  pam_auth_srv.so
kerberos accountrequiredpam_krb5.so no_ccache
kerberossessionrequiredpam_krb5.so no_ccache
#
#
# The PAM configuration file for the `kerberosdownlocal' service
# If Kerberos server is down, uses the local service
#
kerberosdownlocal auth  requisite  pam_securetty.so
kerberosdownlocal auth  optionalpam_auth_srv.so
kerberosdownlocal auth\
   [ success=done new_authtok_reqd=done authinfo_unavail=ignore default=die ] \
   pam_krb5.so no_ccache
kerberosdownlocal auth  requiredpam_unix2.so
kerberosdownlocal account \
   [ success=done new_authtok_reqd=done authinfo_unavail=ignore default=die ] \
   pam_krb5.so no_ccache
kerberosdownlocal account requiredpam_unix2.so
kerberosdownlocal session \
   [ success=done new_authtok_reqd=done authinfo_unavail=ignore default=die ] \
   pam_krb5.so no_ccache
kerberosdownlocal session requiredpam_unix2.so
#
# The PAM configuration file for the `ldap' service
#
ldapauth    sufficientpam_ldap.so
ldapaccount required pam_ldap.so
ldapsession required pam_ldap.so

#
# The PAM configuration file for the `ldapdownlocal' service
# If LDAP server is down, uses the local service
#
ldapdownlocal auth\
   [ success=done new_authtok_reqd=done authinfo_unavail=ignore default=die ] \
   pam_ldap.so
ldapdownlocal auth  requiredpam_unix2.so
ldapdownlocal account \
   [ success=done new_authtok_reqd=done authinfo_unavail=ignore default=die ] \
   pam_ldap.so
```

```
ldapdownlocal account requiredpam_unix2.so
ldapdownlocal session \
   [ success=done new_authtok_reqd=done authinfo_unavail=ignore default=die ] \
   pam_ldap.so
ldapdownlocal session requiredpam_unix2.so

#
# The PAM configuration file for the `tacplus' service
#
tacplus auth     requisite  pam_securetty.so
tacplus auth     required   pam_tacplus.so encrypt
tacplus auth     optional   pam_auth_srv.so
tacplus account  required   pam_tacplus.so encrypt service=ppp protocol=lcp
tacplus session  required   pam_tacplus.so encrypt service=ppp protocol=lcp

s_tacplus auth     requisite  pam_securetty.so
s_tacplus auth     required   pam_tacplus.so encrypt use_first_pass
s_tacplus account  required   pam_tacplus.so encrypt service=ppp protocol=lcp
s_tacplus session  required   pam_tacplus.so encrypt service=ppp protocol=lcp

#
# The PAM configuration file for the `radius' service
#
radius auth       requisite  pam_securetty.so
radius auth       required   pam_radius_auth.so
radius auth       optional   pam_auth_srv.so
radius account    required   pam_radius_auth.so
radius session    required   pam_radius_auth.so

s_radius auth       requisite  pam_securetty.so
s_radius auth       required   pam_radius_auth.so use_first_pass
s_radius account    required   pam_radius_auth.so
s_radius session    required   pam_radius_auth.so

#
# The PAM configuration file for the `local' service
#
local auth       requisite  pam_securetty.so
local auth       required   pam_unix2.so
local account    required   pam_unix2.so
local password   required   pam_unix2.so md5 use_authtok
local session    required   pam_unix2.so

s_local auth       requisite  pam_securetty.so
s_local auth       required   pam_unix2.so use_first_pass
s_local account    required   pam_unix2.so
s_local password   required   pam_unix2.so md5 use_authtok
s_local session    required   pam_unix2.so

#
# The PAM configuration file for the `remote' service
#
remoteauth       required  pam_permit.so
remoteaccount    required  pam_permit.so
remotepassword   required  pam_permit.so
remotesession    required  pam_permit.so

#
# The PAM configuration file for the `login' service
#
```

```
loginauth       requisite  pam_securetty.so
loginauth       required   pam_unix2.so
loginauth       optional   pam_group.so
loginaccount    requisite  pam_time.so
loginaccount    required   pam_unix2.so
loginpassword   required   pam_unix2.so md5 use_authtok
loginsession    required   pam_unix2.so
login   session    required   pam_limits.so

#
# The PAM configuration file for the `xsh' service
#
sshdauth       required   pam_unix2.so
sshdauth       optional   pam_group.so
sshdaccount    requisite  pam_time.so
sshdaccount    required   pam_unix2.so
sshdpassword   required   pam_unix2.so md5 use_authtok
sshdsession    required   pam_unix2.so
sshd    session    required   pam_limits.so

#
# The PAM configuration file for the `passwd' service
#
passwdpassword   required   pam_unix2.so md5
#
# The PAM configuration file for the `samba' service
#
sambaauth       required   pam_unix2.so
sambaaccount    required   pam_unix2.so
#
```

```
# The PAM configuration file for the `su' service
#
suauth       required   pam_wheel.so
suauth       sufficient pam_rootok.so
suauth       required   pam_unix2.so
suaccount    required   pam_unix2.so
susession    required   pam_unix2.so

#
# Information for the PPPD process with the 'login' option.
#
ppp     auth    required    pam_nologin.so
ppp     auth    required    pam_unix2.so
ppp     account required    pam_unix2.so
ppp     session required    pam_unix2.so

#
# Information for the ipppd process with the 'login' option: local authent.
#
ippp    auth    required    pam_nologin.so
ippp    auth    required    pam_unix2.so
ippp    account required    pam_unix2.so
ippp    session required    pam_unix2.so

# Information for the ipppd process with the 'login' option: radius authent.
#ippp auth required    pam_radius_auth.so conf=/etc/raddb/server
#ippp auth optional    pam_auth_srv.so
#ippp account required    pam_radius_auth.so conf=/etc/raddb/server
#ippp session required    pam_radius_auth.so conf=/etc/raddb/server
```

Appendix D - Linux-PAM

```
#
# The PAM configuration file for the `other' service
#
otherauth       required   pam_warn.so
otherauth       required   pam_deny.so
otheraccount    required   pam_deny.so
otherpassword   required   pam_warn.so
otherpassword   required   pam_deny.so
othersession    required   pam_deny.so
```

Reference

The Linux-PAM System Administrators' Guide
Copyright (c) Andrew G. Morgan 1996-9. All rights reserved.
Email: morgan@linux.kernel.org

Appendix E - Upgrades and Troubleshooting

Upgrades

Users should upgrade the Secure Console Port Server SSH whenever there is a bug fix or new features that they would like to have. Below are the six files added by Black Box to the standard Linux files in the /proc/flash directory when an upgrade is needed. They are:

- boot_ori - original boot code
- boot_alt - alternate boot code
- syslog - event logs (not used by Linux)
- config - configuration parameters, only the boot parameters are used by the boot code
- zImage - Linux kernel image
- script - file where all Secure Console Port Server SSH configuration information is stored

The Upgrade Process

To upgrade the Secure Console Port Server SSH, follow these steps:

Step 1: Log in to the Secure Console Port Server SSH as root.

Provide the root password if requested.

Step 2: Go to the /proc/flash directory using the following command:

```
cd /proc/flash
```

Step 3: FTP to the host where the new firmware is located.

Log in using your username and password. Go to the directory where the firmware is located. Select binary transfer and “get” the firmware file.

Note: The destination file name in the /proc/flash directory must be zImage. Example (hostname = server; directory = /tftpboot; username= admin; password = adminpw; firmware filename on that server = zImage.134).

```
ftp
> open server
> user admin
> Password: adminpw
> cd /tftpboot
> bin
> get zImage.134 zImage
> quit
```

Note: Due to space limitations, the new zImage file may not be downloaded with a different name, then renamed. The Secure Console Port Server SSH searches for a file named zImage when booting and there is no room in flash for two zImage files.

Step 4: Run zImage.

To make sure the downloaded file is not corrupted or that the zImage saved in flash is OK the user should run:

```
md5sum -b /proc/flash/zImage
```

Step 5: Check text file information.

Now the user should check with the information present in the text file saved in the Black Box site (e.g. zImage.134.md5sum). If the numbers match, the downloaded file is not corrupted.

Step 6: Issue the command reboot.

```
reboot
```

Step 7: Confirm that the new Linux kernel has taken over.

After rebooting, the new Linux kernel will take over. This can be confirmed by typing

```
cat /proc/version
```

to see the Linux kernel version.

Appendix E - Upgrades and Troubleshooting

Troubleshooting

Flash Memory Loss

If the contents of flash memory are lost after an upgrade, please follow the instructions below to restore your system:

Step 1: Turn the Secure Console Port Server SSH OFF, then back ON.

Step 2: Using the console, wait for the self test messages.

If you haven't got any, make sure you have the right settings. If you really get no boot message, press <s> right after powering ON and skip ALTERNATE boot code. That will make the boot run its ORIGINAL boot code.

Step 3: During the self test, press <Esc> after the Ethernet test.

Step 4: When the Watch Dog Timer prompt appears, press <Enter>.

Step 5: Choose the option Network Boot when asked.

Step 6: Enter the IP address of the Ethernet interface.

Step 7: Enter the IP address of the host where the new zImage file is located.

Step 8: Enter the file name of the zImage file on the host.

Step 9: Select the TFTP option instead of BOOTP.

The host must be running TFTPD and the new zImage file must be located in the proper directory. e.g. /tftpboot for Linux.

Step 10: Accept the default MAC address by pressing <Enter>.

The Secure Console Port Server SSH should begin to boot off the network and the new image will be downloaded and begin running in RAM. At this point, follow the upgrade steps above (login, cd /proc/flash, ftp, and so forth) to save the new zImage file into flash again.

Appendix E - Upgrades and Troubleshooting

Note: Possible causes for the loss of flash memory may include: downloaded wrong zImage file, downloaded as ASCII instead of binary; problems with flash memory.

If the Secure Console Port Server SSH booted properly, the interfaces can be verified using *ifconfig* and *ping*. If ping does not work, check the routing table using the command route. Of course, all this should be tried after checking that the cables are connected correctly.

The file /etc/config_files contains a list of files acted upon by saveconf and restoreconf. If a file is missing, it will not be loaded onto the ramdisk on boot. The following table lists files that should be included in the /etc/config_files file and which programs use each.

Table 30: Files to be included in /etc/config_file and the program to use

File	Program
/etc/securetty	telnet, login, su
/etc/issue	getty
/etc/getty_ttyS0	login (via console)
/etc/hostname	tcp
/etc/hosts	tcp
/etc/host.conf	tcp
/etc/nsswitch.conf	dns
/etc/resolv.conf	dns
/etc/config_files	saveconf
/etc/passwd	login, passwd, adduser...
/etc/group	login, passwd, adduser...

Table 30: Files to be included in /etc/config_file and the program to use

File	Program
/etc/ssh/ssh_host_key.pub	sshd
/etc/ssh/sshd_config	sshd
/etc/ssh/ssh_config	ssh client
/etc/ssh/ssh_host_key	sshd (ssh1)
/etc/ssh/ssh_host_key.pub	sshd (ssh1)
/etc/ssh/ssh_host_dsa_key	sshd (ssh2)
/etc/ssh/ssh_host_dsa_key.pub	sshd (ssh2)
/etc/snmp/snmpd.conf	snmpd
/etc/portslave/pslave.conf	cy_ras, portslave, configuration information
/etc/network/ifcfg_eth0	ifconfig eth0, cy_ras, rc.sysinit
*/etc/network/ifcfg**	ifconfig, cy_ras, rc.sysinit
/etc/network/ifcfg_lo ifconfig	lo, cy_ras, rc.sysinit
/var/run/radsession.id	radinit, radius authentication process
/home	adduser, passwd
/etc/network/st_routes	ifconfig, cy_ras, rc.sysinit
/etc/syslog-ng/syslog-ng.conf	syslog-ng

Important! If any of the files listed in /etc/config_files is modified, the Secure Console Port Server SSH administrator must execute the command *saveconf* before rebooting the Secure Console Port Server SSH or the changes will be lost. If a file is created (or a filename altered), its name must be added to this file before executing saveconf and rebooting.

Appendix E - Upgrades and Troubleshooting

Important! Black Box Technical Support is always ready to help with any configuration problems. Before calling, execute the command

```
cat /proc/version
```

and note the Linux version and Secure Console Port Server SSH version written to the screen. This will speed the resolution of most problems.

Hardware Test

A hardware test called *tstest* is included with the Secure Console Port Server SSH firmware. It is a menu-driven program, run by typing tstest at the command prompt. The various options are described below. Note that the Secure Console Port Server SSH should not be tested while in use as the test will inactivate all ports. You should inactivate all processes that may use the serial ports: inetd, sshd, cy_ras, and cy_buffering. Following are the hardware test steps:

Step 1: signal_ras stop.

Step 2: Perform all hardware tests needed.

Step 3: signal_ras start.

Port Test

Either a cross cable or a loop-back connector is necessary for this test. Their pinout diagrams are supplied in Appendix B - Cabling, Hardware, and Electrical Specifications. Connect the loop-back connector to the modem cable and then connect the modem cable to the port to be tested (or connect a cross cable between two ports to be tested). In the case of the Secure Console Port Server SSH 1-Port, connect the DB-25 loop-back connector to the console cable using a DB-9 - DB-25 convertor. When tstest senses the presence of the cable or connector, the test will be run automatically and the result shown on the screen.

line of data corresponds to a port in test. The last four columns (DATA, CTS, DCD, and DSR) indicate errors. The values in these columns should be zero. Below is an example of the output screen.

```
         <- Packets ->                  <- Errors ->
From    To    Sent  Received    Passes     Data    CTS    DCD      DSR
 2 <-> 2       35      35         35         0      0      0        0
 4 <-> 5       35      35         35         0      0      0        0
 5 <-> 4       35      35         35         0      0      0        0
```

When this test is run with a cable or connector without the DSR signal (see the pinout diagram for the cable or connector being used), errors will appear in the DSR column. This does not indicate a problem with the port. In the example above, tstest perceived that a loop-back connector was attached to port 2 and that a cross cable was used to connect ports 4 and 5.

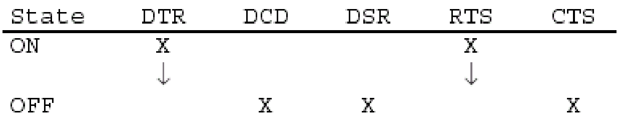
Port Conversation

This test sends and receives data on the selected port. One way to run this test is to place a loop-back connector on the port to be tested and begin. Enter the number of the port and a baud rate (9600 is a typical value). Type some letters, and if the letters appear on the screen, the port is working. If the letters do not appear on the screen (which also occurs if the loop-back connector is removed), the port is not functioning correctly.

A second method that can be used to test the port is to connect it to a modem with a straight cable. Begin the test and type “at”. The modem should respond with “OK”, which will appear on the screen. Other commands can be sent to the modem or to any other serial device. Press Ctrl-Q to exit the terminal emulation test.

Test Signals Manually

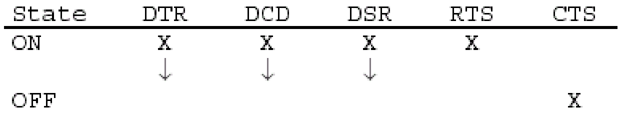
This test confirms that signals are being sent and received on the selected port. Neither the loop-back connector nor the cross cable are necessary. Enter the number of the port to be tested and begin the test.



State	DTR	DCD	DSR	RTS	CTS
ON	X			X	
	↓			↓	
OFF		X	X		X

Figure 48: Initial test

First, type Ctrl-D to see the X in the DTR column move position, then type Ctrl-R to see the X in the RTS column change position. If each of the Xs moves in response to its command, the signals are being sent. Another method to test the signals is to use a loop-back connector. Enter the number of the port with the loopback connector and start the test. In this case, when Ctrl-D is typed, the Xs in the first three columns will move as shown below.



```
State      DTR      DCD      DSR      RTS      CTS
ON          X        X        X        X
            ↓        ↓        ↓

OFF                                             X
```

Figure 49: Second screen, showing changed positions

This is because the test is receiving the DTR signal sent through the DCD and DSR pins. When Ctrl-R is typed, the Xs in the RTS and CTS columns should move together. If the Xs change position as described, the signals are being sent and received correctly.

Test Analog Ports (for the Secure Console Port Server SSH 1-Port only)

This test consecutively reads the Analog-to-Digital converters on both analog ports and compares the variance between the current reading and the first reading (pattern). One way to run this test is to place a short-circuit connector on the ports. The reading should be at the bottom of the Analog-to-Digital scale. Another way is to place 10K ohms on the ports. The reading should be at half scale. A third way is to place no connector at all. The reading should be at full scale. Below is an example of the output screen, when using 10K ohms.

```
           <--------- VALUE --------->         <---- ERRORS ---->

   ANALOG  Initial   Current   Correct  Passes       Data

------------------------------------------------------------

     1       3fff      3fff       7        7             0

     2       3fff      3fff       7        7             0


   Press <ESC> to stop the test.
```

Test Digital Ports (for the Secure Console Port Server SSH 1-Port only)

This test consecutively reads the digital ports and compares the variance between the current reading and the first reading (pattern). One way to run this test is to place a loop-back connector on the ports. The port reading on this condition should be 1. It means that pin+ and pin- have a closed loop. For the ports without loop-back, the reading should be 0. It means that pin+ and pin- have an open loop. Below is an example of the output screen. Digital ports 1, 2, 3, 4 and 5 have loop-back connectors. Digital ports 6, 7 and 8 have not.

```
        <--------- VALUE --------->         <---- ERRORS ---->

DIGITAL Initial   Current   Correct  Passes        Data

--------------------------------------------------------------

  1         1         1        10       10           0

  2         1         1        10       10           0

  3         1         1        10       10           0

  4         1         1        10       10           0

  5         1         1        10       10           0

  6         0         0        10       10           0

  7         0         0        10       10           0

  8         0         0        10       10           0


Press <ESC> to stop the test.
```

Single User Mode

The Secure Console Port Server SSH has a single user mode used when:

- The name or password of the user with root privileges is lost or forgotten,
- After an upgrade or downgrade which leaves the Secure Console Port Server SSH unstable,

- After a configuration change which leaves the Secure Console Port Server SSH inoperative or unstable.

Type the word “single” (with a blank space before the word) during boot using a console connection. This cannot be done using a telnet or other remote connection. The initial output of the boot process is shown below.

```
Entry Point = 0x00002120

loaded at: 00002120 0000D370

relocated to: 00300020 0030B270

board data at: 003052C8 0030537C

relocated to: 002FF120 002FF1D4

zimage at: 00008100 0006827E

relocated to: 00DB7000 00E1717E

initrd at: 0006827E 0024F814

relocated to: 00E18000 00FFF596

avail ram: 0030B270 00E18000

Linux/PPC load: root=/dev/ram
```

After printing “Linux/PPC load: root=/dev/ram,” the Secure Console Port Server SSH waits approximately 10 seconds for user input. This is where the user should type “<sp>single” (spacebar, then the word “single”). When the boot process is complete, the Linux prompt will appear on the console:

```
[root@(none) /]#
```

If the password or username was forgotten, execute the following commands:

```
passwd

saveconf

reboot
```

For configuration problems, you have two options:

Step 1: Edit the file(s) causing the problem with vi, then execute the commands:

```
saveconf
reboot
```

Step 2: Reset the configuration by executing the commands:

```
echo 0 > /proc/flash/script
reboot
```

If the problem is due to an upgrade/downgrade, a second downgrade/upgrade will be necessary to reverse the process. First, the network must be initialized in order to reach a ftp server. Execute the following script, replacing the parameters with values appropriate for your system. If your ftp server is on the same network as the Secure Console Port Server SSH, the gw and mask parameters are optional.

```
config_eth0 ip 200.200.200.1 mask 255.255.255.0 gw 200.200.200.5
```

At this point, the DNS configuration (in the file /etc/resolv.conf) should be checked. Then, download the kernel image using the ftp command.

Troubleshooting the Web Configuration Manager

What to do when the initial Web page does not appear

Try pinging, telnetting, or tracerouting to the Secure Console Port Server SSH to make sure it is reachable. If not, the problem is probably in the network or network configuration. Are the interfaces up? Are the IP addresses correct? Are filters configured which block the packets? If the Secure Console Port Server SSH is reachable, see if the /bin/webs process is running by executing the command ps. If it is not, type /bin/webs & to start it. If the /bin/webs process is not being initialized during boot, change the file /etc/inittab.

How to restore the Default Configuration of the Web Configuration Manager

This would be required only when the root password was lost or the configuration file /etc/websum.conf was damaged. From a console or telnet session, edit the file /etc/config_files. Find the reference to /etc/websum.conf and delete it. Save the modified /etc/config_files file. Execute the command saveconf. Reboot the system. Enter into the Web Configuration Manager with the default username and password (root/tslinux). Edit the file /etc/config_files and insert the reference to /etc/websum.conf.

Appendix E - Upgrades and Troubleshooting

Recover access to the Secure Console Port Server SSH 1-Port console port

There is no dedicated console port available in the Secure Console Port Server SSH 1-Port. As factory default the serial port is set to work as a console port to allow initial product configuration. After that, changes can still be made through the Ethernet port and a Telnet command. If for some reason this access is lost (usually misconfiguration), the product can only be configured if the
steps bellow are followed.

Step 1: Power the Secure Console Port Server SSH 1-Port off.

Step 2: Connect the Secure Console Port Server SSH 1-Port to a terminal configured to work at 9600 bps, with 8 bits, no parity and 1 stop bit.

Step 3: Press and hold the ADM button and power on the Secure Console Port Server SSH 1-Port.

There's a small hole in the box containing an internal ADM button that can be reached by a thin, sharp object.

Step 4: Release the ADM button when the self test starts on the terminal's screen.

The Secure Console Port Server SSH 1-Port will be now in single user mode, the serial port will work as a console port and the product can de reconfigured. Notice that no previous configuration is lost. After finishing, save the configuration (saveconf), power the Secure Console Port Server SSH 1-Port off, and reconnect the original device to the serial port.

Using a different speed for the Serial Console

The serial console is originally configured to work at 9600 bps. If you want to change that, it is necessary to change the configuration following the steps:

Step 1: Run bootconf. The user will be presented with the screen:

```
Current configuration

MAC address assigned to Ethernet [00:60:2e:00:16:b9]

IP address assigned to Ethernet interface [192.168.160.10]

Watchdog timer ((A)ctive or (I)nactive) [A]

Firmware boot from ((F)lash or (N)etwork) [F]
```

```
Boot type ((B)ootp,(T)ftp or Bot(H)) [T]

Boot File Name [zvmppctsbin]

Server's IP address [192.168.160.1]

Console speed [9600]

(P)erform or (S)kip Flash test [P]

(S)kip, (Q)uick or (F)ull RAM test [F]

Fast Ethernet ((A)uto Neg, (1)00 BtH, 100 Bt(F), 10 B(t)F, 10
Bt(H)) [A]

Fast Ethernet Maximum Interrupt Events [0]
```

Type <Enter> for all fields but the Console Speed. When presented the following line:

```
Do you confirm these changes in flash ( (Y)es, (N)o (Q)uit )
[N] :
```

Step 2: Enter Y and the changes will be saved in flash.

Step 3: Logout and login again to use the console at the new speed.

Appendix E - Upgrades and Troubleshooting

CPU LED

Normally the CPU status LED should blink consistently one second on, one second off. If this is not the case, an error has been detected during the boot. The blink pattern can be interpreted via the following table:

Table 31: CPU LED Code Interpretation

Event	CPU LED Morse code
Normal Operation	S (short, short, short . . .)
Flash Memory Error - Code	L (long, long, long . . .)
Flash Memory Error - Configuration	S, L
Ethernet Error	S, S, L
No Interface Card Detected	S, S, S, L
Network Boot Error	S, S, S, S, L
Real-Time Clock Error	S, S, S, S, S, L

Note: The Ethernet error mentioned in the above table will occur automatically if the Fast Ethernet link is not connected to an external hub during the boot. If the Fast Ethernet is not being used or is connected later, this error can be ignored.

Appendix F - Certificate for HTTP Security

Introduction

The following configuration will enable you to obtaining a Signed Digital Certificate. A certificate for the HTTP security is created by a CA (Certificate Authority). Certificates are most commonly obtained through *generating public and private keys*, using a public key algorithm like RSA or X509. The keys can be generated by using a key generator software.

Procedure

Step 1: Enter OpenSSL command.

On a Linux computer, key generation can be done using the OpenSSL package, through the following command:

```
# openssl req -new -nodes -keyout private.key -out public.csr
```

If this command is used, the following information is required:

Table 32: Required information for the OpenSSL package

Parameter	Description
Country Name (2 letter code) [AU]:	The country code consisting of two letters.
State or Province Name (full name) [Some-State]:	Provide the full name (not the code) of the state.
Locality Name (e.g., city) []:	Enter the name of your city.
Organization Name (e.g., company) [Internet Widgits Ltd]:	Organization that you work for or want to obtain the certificate for.
Organizational Unit Name (e.g., section) []:	Department or section where you work.
Common Name (e.g., your name or your server's hostname) []:	Name of the machine where the certificate must be installed.

Appendix F - Certificate for HTTP Security

Table 32: Required information for the OpenSSL package

Parameter	Description
Email Address []:	Your email address or the administrator's email address.

The other requested information can be skipped.

The certificate signing request (CSR) generated by the command above contains some personal (or corporate) information and its public key.

Step 2: Submit CSR to the CA.

The next step is to submit the CSR and some personal data to the CA. This service can be requested by accessing the CA Web site and is not free. There is a list of CAs at the following URL

```
pki-page.org
```

The request will be analyzed by the CA, for policy approval and to be signed.

Step 3: Upon receipt, install certificate.

After the approval, the CA will send a certificate file to the origin, which we will call Cert.cer, for example purposes. The certificate is also stored on a directory server. The certificate must be installed in the GoAhead Web server, by following these instructions:

Step A: Open a Black Box Terminal Server session and do the login.

Step B: Join the certificate with the private key into the file /web/server.pem.

```
#cat Cert.cer private.key > /web/server.pem
```

Step C: Copy the certificate to the file /web/cert.pem.

```
#cp Cert.cer /web/cert.pem
```

Step D: Include the files /web/server.pem and /web/cert.pem in /etc/config_files.

Step E: Save the configuration in flash.

```
#saveconf
```

Step F: The certification will be effective in the next reboot.

This page has been left intentionally blank.

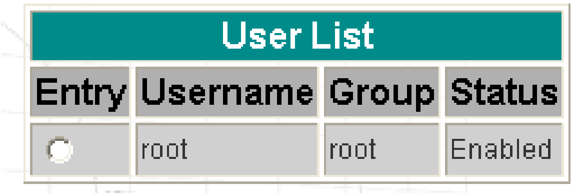
Appendix G - Web User Management

Introduction

In the Secure Console Port Server SSH Web server, the user database is completely separated from the system's (as defined in the /etc/passwd file), and the logic used for managing permissions is also different. The Web's user database is stored in the /etc/websum.conf file, and it has basically three lists: *users, user groups* and *access limits*.

Default Configuration for Web User Management

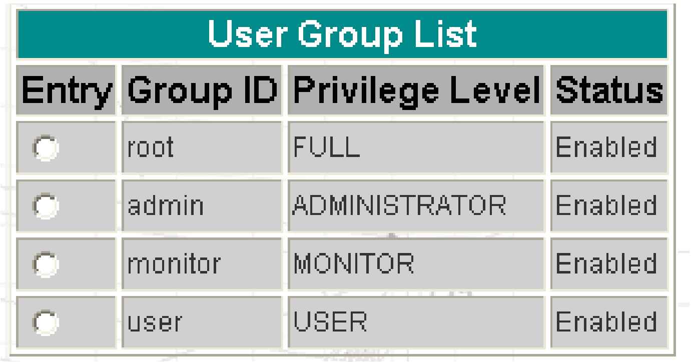
The following three screen shots show the default configuration for User List, User Group List, and Access Limit List pages, respectively.



User List

Entry	Username	Group	Status
○	root	root	Enabled

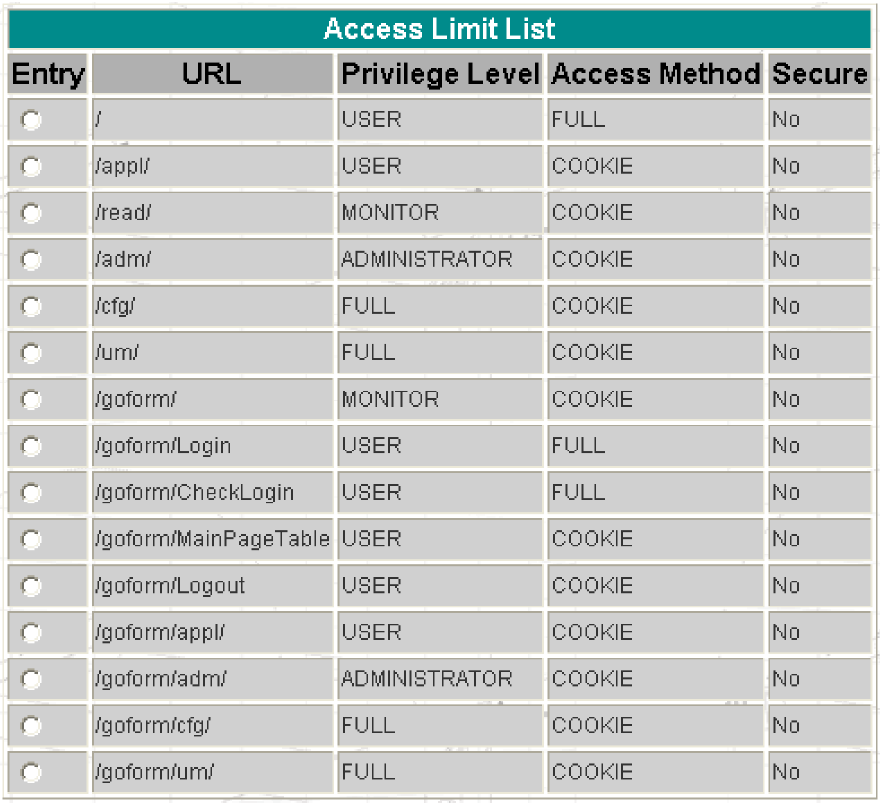
Figure 50: User List default page



User Group List

Entry	Group ID	Privilege Level	Status
○	root	FULL	Enabled
○	admin	ADMINISTRATOR	Enabled
○	monitor	MONITOR	Enabled
○	user	USER	Enabled

Figure 51: User Group List default page



Access Limit List				
Entry	URL	Privilege Level	Access Method	Secure
○	/	USER	FULL	No
○	/appl/	USER	COOKIE	No
○	/read/	MONITOR	COOKIE	No
○	/adm/	ADMINISTRATOR	COOKIE	No
○	/cfg/	FULL	COOKIE	No
○	/um/	FULL	COOKIE	No
○	/goform/	MONITOR	COOKIE	No
○	/goform/Login	USER	FULL	No
○	/goform/CheckLogin	USER	FULL	No
○	/goform/MainPageTable	USER	COOKIE	No
○	/goform/Logout	USER	COOKIE	No
○	/goform/appl/	USER	COOKIE	No
○	/goform/adm/	ADMINISTRATOR	COOKIE	No
○	/goform/cfg/	FULL	COOKIE	No
○	/goform/um/	FULL	COOKIE	No

Figure 52: Access Limit List default page

Appendix G - Web User Management

How Web User Management works

When a user logs in, the username and the password are encrypted and stored in the browser. Whenever a URL is requested, the User Manager will perform the following tasks:

Task 1: Check the URL in the Access Limit List

The Web server first scans for the full URL, and then it looks for the subdirectories, until reaching the root directory “/.” (In the URL http://CAS/goform/cfg/IPTablesRulesHandle, the access limits will be scanned in the following order: /goform/cfg/IPTablesRulesHandle, /goform/cfg, /goform and /.) When the URL matches an Access Limit, the following information will be available:

Accessibility When configured as FULL ACCESS, the URL can be accessed without any authentication; otherwise, the user can authenticate with BASIC, DIGEST or COOKIE authentication. The last type is recommended, because it allows the user to log out in the end of the session. The page will not be accessible when the accessibility is configured as NO ACCESS.

Security When set to be secure, the page will be accessed only through HTTPS, which will encrypt the pages through OpenSSL. If the browser is in unsecure mode, the protocol and the port will change to HTTPS.

Privilege This is the level of accessibility of the page. If the privilege is USER, any user will be able to access the page. If the privilege is FULL, only users with full access will be able to access the page. There are two levels between them: MONITOR and ADMINISTRATOR.

Task 2: Read the Username and the Password

This is done when the page must be accessed through authentication. If the username matches an entry in the users list, the following information will be available:

Enabled The username must be enabled to be authenticated.

Encrypted password The password passed by the browser must match the one registered in the entry.

Group Each username is linked to a user group.

Task 3: Look for the group retrieved in the user groups list

The user group entry will have the following information:

Enabled The group must be enabled to grant access to the URL.

Privilege The group can have four privileges: in increasing order, they are USER, MONITOR, ADMINISTRATOR and FULL. The group privilege will be compared with the URL privilege. If it is greater or equal, the URL can be accessed by the user; otherwise, access is denied.

Web User Management Configuration - Getting Started

The users, groups and access limits for Web User Management are configurable with your browser, though it is not recommended to change the groups and the access limits. In the default configuration:

- The access limits have privileges based on the functionality of the Web page.
- There are four different groups (root, monitor, admin and user), each one with a specific privilege.
- There is one root user (username is root and password is linux).

Appendix G - Web User Management

Changing the Root Password

The first thing to do after logging into a Web session the first time must be to change the root password. See Security Issue under Figure 10: Configuration & Administration Menu page.

Step 1: Click on the link Web User Management > Users.

Step 2: Select the root user and click the Change Password button.

Step 3: Type the password twice and click the Submit button.

Step 4: Click on the link Web User Management > Load/Save Web Configuration.

The Login page will appear.

Step 5: Type the username *root* **and the password that was configured, then click on the Login button.**

Step 6: After the authentication, click on the Save Configuration button.

Step 7: Click on the link Administration > Load/Save Configuration.

Step 8: Click on the Save to Flash button.

Adding and Deleting Users

Adding a User

Step 1: Click on the link Web User Management > Users.

Step 2: Click on the Add User button.

Step 3: Configure the new user.

Type the username, the password (twice) and select a user group, depending on the access privilege desired. Leave the item Enabled checked.

Step 4: Click on the Submit button.

A confirmation message will appear.

Step 5: If there are more users to be added, repeat the steps 1 to 4.

Step 6: Click on the link Web User Management > Load/Save Web Configuration.

Step 7: Click on the Save Configuration button.

This will save the users added in the file /etc/websum.conf.

Step 8: Click on the link Administration > Load/Save Configuration.

Step 9: Click on the Save to Flash button.

Step 10: Test the user(s) added.

Log out the current user (Go to the link Application > Logout) and log in again, with the new user.

Deleting a User

The root user is delete-protected, and, because of that, it cannot be removed from the user list. The other users can be deleted.

Step 1: Click on the link Web User Management > Users.

Step 2: Select the user to be deleted and click on the Delete User button.

A confirmation message will appear.

Step 3: If there are more users to be deleted, repeat the steps 1 and 2.

Step 4: Click on the link Web User Management > Load/Save Web Configuration.

Step 5: Click on the Save Configuration button.

This will save the users added in the file /etc/websum.conf

Step 6: Click on the link Administration > Load/Save Configuration.

Step 7: Click on the Save to Flash button.

Appendix G - Web User Management

Adding and Deleting User Groups

The default configuration already comes with four user groups, and, for most of the cases, they will be enough. However, you have the option of editing the user groups.

Adding a group

Step 1: Click on the link Web User Management > Groups.

Step 2: Click on the Add Group button

Step 3: Configure the new group.

Type the group name and select the access privilege this group will have. Leave the Enabled item checked.

Step 4: Click on the Submit button.

A confirmation message will appear.

Step 5: If there are more groups to be added, repeat the steps 1 to 4.

Step 6: Click on the link Web User Management > Load/Save Web Configuration.

Step 7: Click on the Save Configuration button.

This will save the users added in the file /etc/websum.conf

Step 8: Click on the link Administration > Load/Save Configuration.

Step 9: Click on the Save to Flash button.

Deleting a group

Before deleting a group, make sure that there are no users using that group.

Step 1: Click on the link Web User Management > Groups.

Step 2: Select the group to be deleted and click on the Delete Group button.

A confirmation message will appear.

Step 3: If there are more groups to be deleted, repeat the steps 1 and 2.

Step 4: Click on the link Web User Management > Load/Save Web Configuration.

Step 5: Click on the Save Configuration button.

This will save the users added in the file /etc/websum.conf

Step 6: Click on the link Administration > Load/Save Configuration.

Step 7: Click on the Save to Flash button.

Adding and Deleting Access Limits

The default configuration has the access limits set according to the functionality of the Web page.

- Pages or forms which causes the configuration to change will have FULL privilege (only high-privileged users will have access to it).
- Pages which change the status of the board without changing the configuration will have ADMINISTRATOR privilege;
- Pages with the system information will have MONITOR privilege.
- Only application pages will have USER privilege.

Changing access limits is not recommended, unless you need to create or change the web server pages; even so, the user should place the web pages in the subdirectories with the privilege desired. For example, a page with ADMINISTRATOR privilege should be placed in /adm.

Adding an Access Limit

Step 1: Click on the link Web User Management > Access Limits.

Step 2: Click on the Add Access Limit button.

Step 3: Configure the new access limit.

Type the URL (or the subdirectory), and select the access privilege. If authentication is required to access the page, select COOKIE ACCESS; otherwise, select FULL ACCESS. If this page is confidential, check the Secure box.

Step 4: Click on the Submit button.

A confirmation message will appear.

Step 5: If there are more access limits to be added, repeat the steps 1 to 4.

Step 6: Click on the link Web User Management > Load/Save Web Configuration.

Step 7: Click on the Save Configuration button.

This will save the users added in the file /etc/websum.conf.

Step 8: Click on the link Administration > Load/Save Configuration.

Step 9: Click on the Save to Flash button.

Deleting an access limit

Step 1: Click on the link Web User Management > Access Limits.

Step 2: Select the access limit to be deleted and click on the Delete Access Limit button.

A confirmation message will appear.

Step 3: If there are more access limits to be deleted, repeat the steps 1 and 2.

Step 4: Click on the link Web User Management > Load/Save Web Configuration.

Step 5: Click on the Save Configuration button.

This will save the users added in the file /etc/websum.conf

Step 6: Click on the link Administration > Load/Save Configuration.

Step 7: Click on the Save to Flash button.

This page has been left intentionally blank.

Appendix H - Connect to Serial Ports from Web

Introduction

Depending on how the serial port is configured, connecting to a serial port will either open up a telnet or ssh connection. A serial port configured as socket_server or raw_data will open up a telnet connection while socket_ssh will open up a ssh connection. Any Web user configured in the Web User Management section of the WMI will be able to use this application.

Tested Environment

Table 33: Windows XP + JREv1.4.0_01 or 02

Internet Explorer 6.0	Success
Netscape 6/6.2.3	Success
Netscape 7.0	Success
Mozilla 1.1	Success

Requirements: Java 2 Runtime Environment (JRE) SE v1.4.0_01 or v1.4.0_02 (which can be found at http://java.sun.com/) installed on your PC with your browser acknowledged to use it. You can first check if the browser you are using acknowledges the Java version by following the procedures given in the next sections.

Appendix H - Connect to Serial Ports from Web

On Windows

From Internet Explorer

Go to Tools → Internet Options → Advanced. Scroll down and look for a section on Java. There should be a checkbox that says “Use Java 2 v1.4.0 ...." If there isn't, this could either mean your browser is not activated to use the Java plug-in that came with the JRE you have installed or it just means that you don't have any JRE installed, in which case please install and repeat the check.

If you have already installed JRE and you just want to activate your browser to use it, go to your system's Control Panel → Java Plug-in icon → Browser → check on the browser(s) you want to activate to use the Java Plug-in. Now repeat the check to see if your browser will now use the correct Java Plug-in.

From Netscape or Mozilla

Check to see if Java is enabled. Go to Edit → Preferences → Advanced → Check on Enable Java. To see what version of JRE Plug-in is used, go to Help → About Plug-ins. Scroll down to Java Plug-in section. Check if the Java Plug-in is the version you have installed.

Tip. When installing Netscape 7.0, it will ask if you want to install Sun Java. If you click on the box to install it, a version of JRE will be installed into your system; however, this does not mean that other browsers such as IE will recognize it. If you choose not to install Sun Java through Netscape but do it separately, Netscape 7.0 should automatically detect the JRE, and this can be checked by the instructions mentioned above.

Appendix H - Connect to Serial Ports from Web

Step-by-Step Process

Step 1: Point your browser to the Console Server.

In the address field of your browser type the Console Access Server's IP address. For example:

```
http://10.0.0.0
```

Step 2: Log in.

Log in with a user configured in the Web User Management section, and its password. This will take you to the Configuration and Administration page.

Step 3: Select the Connect to Serial Ports link.

Click on the Connect to Serial Ports link on the Link Panel to the left of the page in the Configuration section. This will take you to the Port Selection page. The ports will be listed by their server farm name if it were configured.

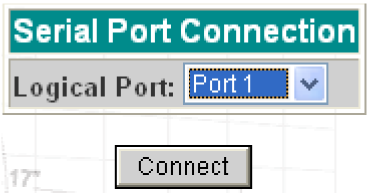
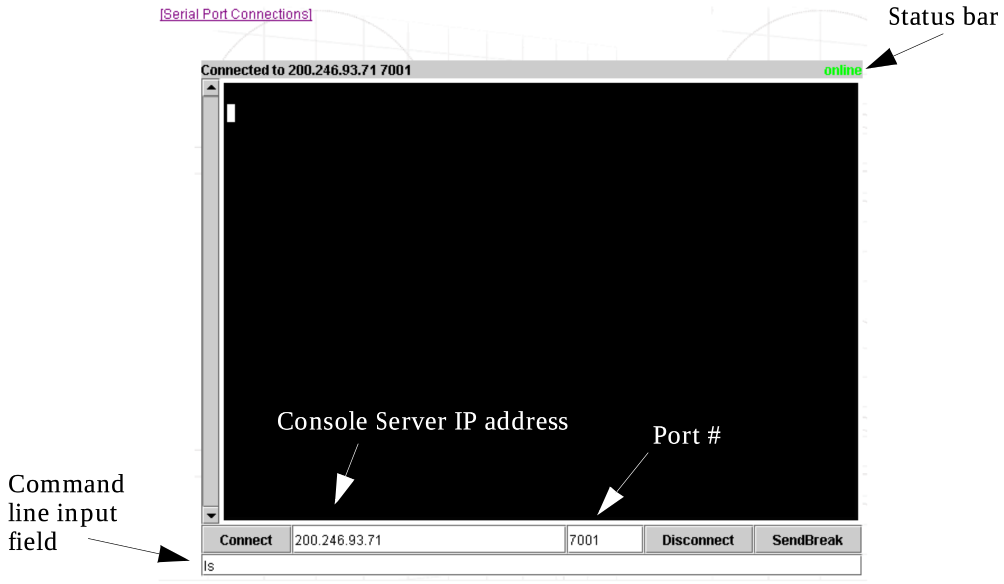


Figure 53: Serial Port Connection page

Step 4: Select port.

On the Port Selection page, choose a port to connect to from the dropdown menu and click the Connect button. This will open a new browser window that contains the applet connecting to the server chosen.



[Serial Port Connections]

Connected to 200.246.93.71 7001 online

Connect 200.246.93.71 7001 Disconnect SendBreak

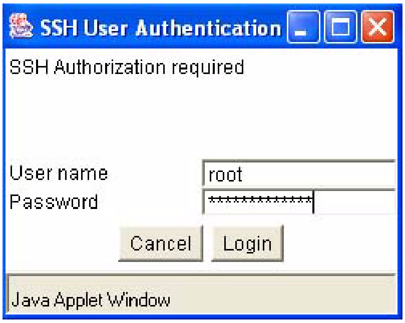
ls

Figure 54: Port Connection page

Step 5: Log in.

If the port selected was configured as socket_server or raw_data, and depending on how it is configured to be authenticated, log in by typing into the terminal.

If the port selected was configured for a ssh connection, a Login window will pop up. If you don't see it pop up, check your taskbar. Enter in the username and the username's password.



SSH User Authentication

SSH Authorization required

User name: root

Password: **************

Figure 55: SSH User Authentication Popup Window

Step 6: Enter command.

Click in the terminal window and start entering commands.

Step 7: To send a break to the terminal.

Click on the SendBreak button.

Step 8: Disconnect connection.

Click on the Disconnect button. Make sure the Status bar shows an Offline status. Closing the popup window will also disconnect you from the server.

Step 9: Reconnect to port.

Refresh the current page by clicking on the refresh icon at the upper right hand corner of the window.

This page has been left intentionally blank.

Appendix I - Examples for Config Testing

Introduction

The following three examples are just given to *test* a configuration. The steps should be followed *after* configuring the Secure Console Port Server SSH.

Console Access Server

With the Secure Console Port Server SSH set up as a CAS you can access a server connected to the Secure Console Port Server SSH through the server's serial console port from a workstation on the LAN or WAN. There is no authentication by default, but the system can be configured for authentication to be performed by a Radius server, a TacacsPlus server, or even by a local database. Either telnet or ssh can be used.

See Appendix A - New User Background Information for more information about ssh. The instructions in Chapter 2 - Installation, Configuration, and Usage will set up a fully-functional, default CAS environment. More options can be added after the initial setup, as illustrated in Chapter 3 - Additional Features.

An example of a CAS environment is shown in Figure 56: CAS diagram with various authentication methods This configuration example has local authentication, an Ethernet interface provided by a router, and serially-connected workstations.

Appendix I - Examples for Config Testing

The following diagram, shows additional scenarios for the Secure Console Port Server SSH: both remote and local authentication, data buffering, and remote access.

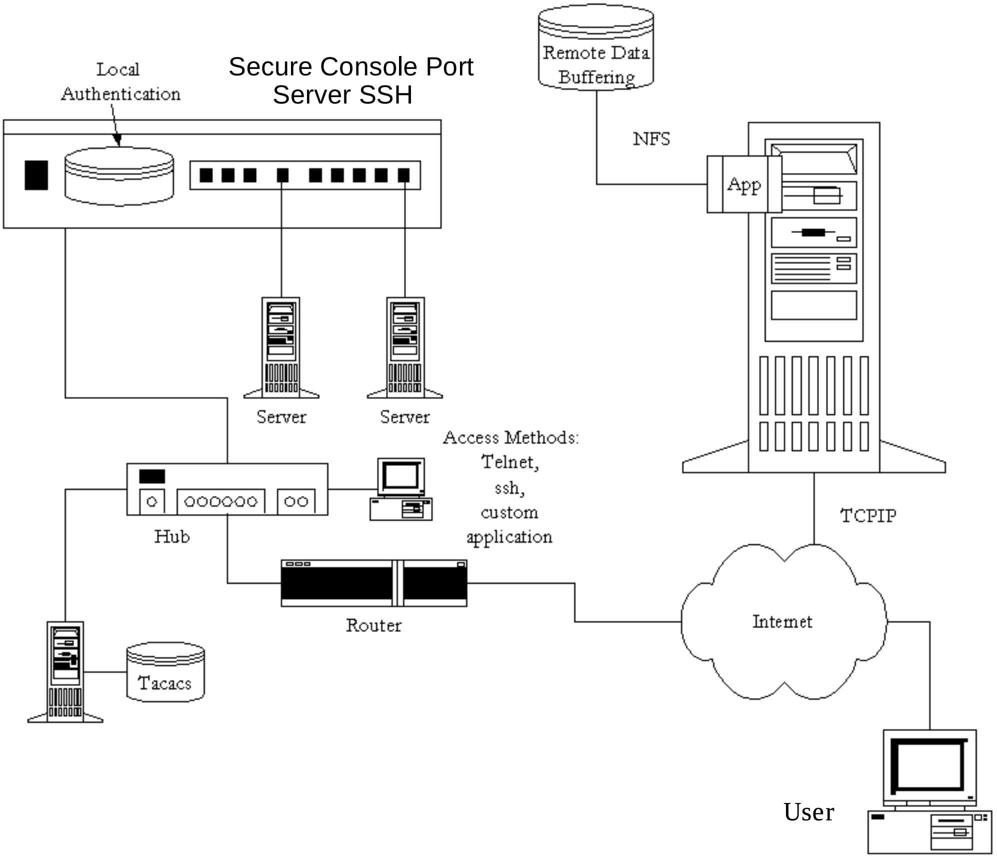


Figure 56: CAS diagram with various authentication methods

As shown in the above figure, our “CAS with local authentication” scenario has either telnet or ssh (a secure shell session) being used. After configuring the serial ports as described in Chapter 3 - Additional Features or in Appendix C - The pslave Configuration File, the following step-by-step check list can be used to test the configuration.

Appendix I - Examples for Config Testing

Step 1: Create a new user.

Run the *adduser <username>* to create a new user in the local database. Create a password for this user by running *passwd <username>*.

Step 2: Confirm physical connection.

Make sure that the physical connection between the Secure Console Port Server SSH and the servers is correct. A cross cable (not the modem cable provided with the product) should be used. Please see Appendix B - Cabling, Hardware, and Electrical Specifications for pin-out diagrams.

Step 3: Confirm that server is set to same parameters as the LES.

The Secure Console Port Server SSH has been set for communication at 9600 bps, 8N1. The server must also be configured to communicate on the serial console port with the same parameters.

Step 4: Confirm routing.

Also make sure that the computer is configured to route console data to its serial console port (Console Redirection).

Step 5: Telnet to the server connected to port 1.

From a server on the LAN (not from the console), try to telnet to the server connected to the first port of the Secure Console Port Server SSH using the following command:

```
telnet 200.200.200.1 7001
```

For both telnet and ssh sessions, the servers can be reached by either:

1. Ethernet IP of the Secure Console Port Server SSH and assigned socket port.

or

2. Individual IP assigned to each port.

If everything is configured correctly, a telnet session should open on the server connected to port 1. If not, check the configuration, follow the steps above again, and check the troubleshooting appendix.

Step 6: Activate the changes.

Now continue on to Task 5: Activate the changes through Task 8: Reboot the Secure Console Port Server SSH listed in Chapter 2 - Installation, Configuration, and Usage.

Note: It is possible to access the serial ports from Microsoft stations using some off-the-shelf packages. Although Black Box is not liable for those packages, successful tests were done using at least one of them. From the application's viewpoint running on a Microsoft station, the remote serial port works like a regular COM port. All the I/O with the serial device attached to the Secure Console Port Server SSH is done through socket connections opened by these packages and a COM port is emulated to the application.

Terminal Server

The Secure Console Port Server SSH provides features for out-of-band management via the configuration of terminal ports. All ports can be configured as terminal ports. This allows a terminal user to access a server on the LAN.

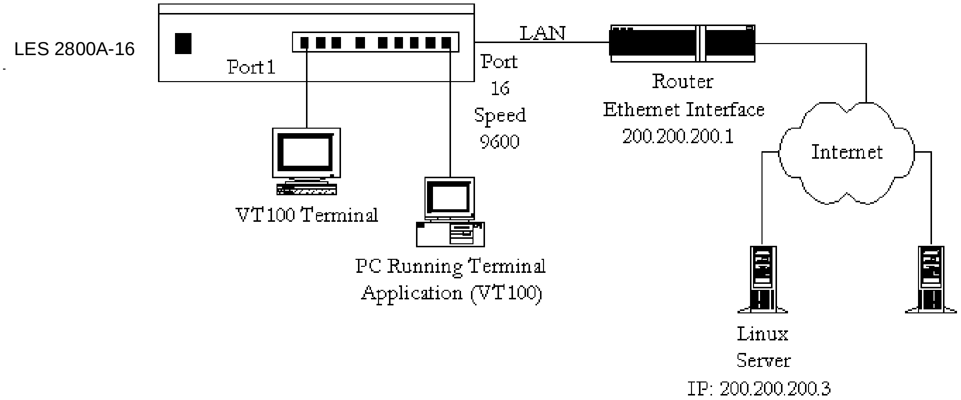


Figure 57: Terminal Server diagram

The terminal can be either a dumb terminal or a terminal emulation program on a PC.

No authentication is used in the example shown above and rlogin is chosen as the protocol. After configuring the serial ports as described in Chapter 3 - Additional Features or in Appendix C - The pslave Configuration File, the following step-by-step check list can be used to test the configuration.

Step 1: Create a new user.

Since authentication was set to none, theSecure Console Port Server SSH will not authenticate the user. However, the Linux Server receiving the connection will. Create a new user on the server called *test* and provide him with the password *test*.

Step 2: Confirm that the server is reachable.

From the console, ping 200.200.200.3 to make sure the server is reachable.

Step 3: Check physical connections.

Make sure that the physical connection between the Secure Console Port Server SSH and the terminals is correct. A cross cable (not the modem cable provided with the product) should be used. Please see the Appendix B - Cabling, Hardware, and Electrical Specifications for pin-out diagrams.

Step 4: Confirm that terminals are set to same parameters as the Secure Console Port Server SSH.

The Secure Console Port Server SSHhas been set for communication at 9600 bps, 8N1. The terminals must also be configured with the same parameters.

Step 5: Log onto server with new username and password.

From a terminal connected to the Secure Console Port Server SSH, try to login to the server using the username and password configured in step one.

Step 6: Activate changes.

Now continue on to Task 5: Activate the changes through Task 8: Reboot the Secure Console Port Server SSH listed in Chapter 2 - Installation, Configuration, and Usage.

Appendix I - Examples for Config Testing

Dial-in Access

The Secure Console Port Server SSH can be configured to accommodate out-of-band management. Ports can be configured on the Secure Console Port Server SSH to allow a modem user to access the LAN. Radius authentication is used in this example and ppp is chosen as the protocol on the serial (dial-up) lines. Black Box recommends that a maximum of two ports be configured for this option.

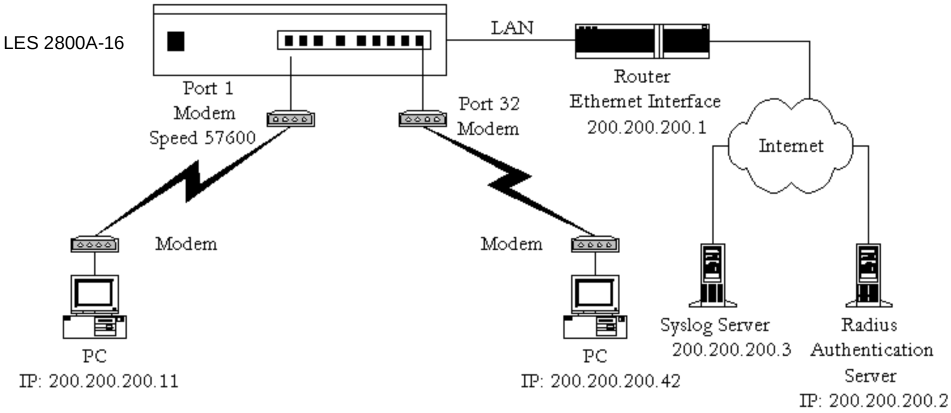


Figure 58: Ports configured for Dial-in Access

After configuring the serial ports as described in Chapter 3 - Additional Features or in Appendix C - The pslave Configuration File, the following step-by-step check list can be used to test the configuration.

Step 1: Create a new user.

Since Radius authentication was chosen, create a new user on the Radius authentication server called *test* and provide them with the password *test*.

Step 2: Confirm that the Radius server is reachable.

From the console, ping 200.200.200.2 to make sure the Radius authentication server is reachable.

Step 3: Confirm physical connections.

Make sure that the physical connection between the Secure Console Port Server SSH and the modems is correct. The modem cable provided with the product should be used. Please see Appendix B - Cabling, Hardware, and Electrical Specifications for pinout diagrams.

Step 4: Confirm modem settings.

The Secure Console Port Server SSHhas been set for communication at 57600 bps, 8N1. The modems should be programmed to operate at the same speed on the DTE interface.

Step 5: Confirm routing.

Also make sure that the computer is configured to route console data to the serial console port.

Step 6: Perform a test dial-in.

Try to dial in to the Secure Console Port Server SSH from a remote computer using the username and password configured in step one. The computer dialing in must be configured to receive its IP address from the remote access server (the Secure Console Port Server SSH in this case) and to use PAP authentication.

Step 7: Activate changes.

Now continue on to Task 5: Activate the changes through Task 8: Reboot the Secure Console Port Server SSH listed in Chapter 2 - Installation, Configuration, and Usage.

This page has been left intentionally blank.

Appendix J - Billing Feature

Introduction

The Secure Console Port Server SSH 1-Port can also be simply used as an intermediate buffer to collect serial data (like billing tickets from a PABX), making them available for a posterior file transfer.

General Feature Description

The Secure Console Port Server SSH 1-Port reads the serial port and saves information to Ramdisk files, limited to a maximum number of records per file or a maximum lifetime. After they are closed, these files are available for file transfer at /var/run/DB.

Configuration

The plsave.conf file has one more “protocol” and three new parameters shown on the Data Buffering section of the Web interface. They are:

all.protocol	billing

Data Buffering section:

all.billing_records	50
all.billing_timeout	60 (min.)
all.billing_eor	"\n"

Once the cy_ras program detects the protocol as “billing,” it starts the billing application. The billing application then opens the port (as configured in pslave.conf) and starts reading it.

Records terminated by "billing_eor string" are expected to be received. The Secure Console Port Server SSH 1-Port doesn't change the termination method, transferring the same sequence to the file. The name of the temporary file used to write these records is:

```
cycXXXXX-YYMMDD.hhmmss.tmp
```

where:

- *XXXXX* is the "hostname" or "serverfarm"
- *YYMMDD* is the year/month/day
- *hhmmss* is the hour:min:sec

This name helps the user archive and browse their directory as the file can be chronologically listed, not based on its creation or modification times, but based on when its contents were recorded. Also, whenever "hostname" is not significant, the user can use the "serverfarm" name (s1.serverfarm in pslave.conf) to match their actual plant (like PABX-trunk9). The temporary file described above is closed and renamed to cycXXXXX-YYMMDD.hhmmss.txt and a new temporary file is opened when:

1. The maximum number of records specified by "billing_records" is reached;
2. The lifetime specified by "billing_timeout" finishes.

If no record is received within a file lifetime period, no file will be actually saved.

Note: A zero-value for "billing_records" stops the application and a zero-value for "billing_timeout" means no timeout is desired and so the file will only be closed after "billing_records" are received.

Disk Space Issue

Finally, it is important to note that there is a protection against disk space problems. If you configure flow control to "hardware" for the serial port (*all.flow = hard* in the pslave.conf file), the application monitors the available disk space and if it is less than 100 Kb, the serial interface deactivates "RTS" signal on the RS-232. "RTS" is reactivated once the disk free space is greater than 120 Kb.

Appendix K - Wiz Application Parameters

Basic Parameters (wiz)

- Hostname
- System IP
- Domain Name
- DNS Server
- Gateway IP
- Network Mask

Access Method Parameters (wiz --ac <type>)

(CAS profile)

- Ipno
- Socket_port
- Protocol
- Modbus_smode
- Users
- Poll_interval
- Tx_interval
- Idletimeout
- Conf.group
- <sN>.serverfarm
- pool_ipno
- pool_socket_port

- pool_serverfarm
- web_WinEMS
- translation

(TS profile)

- Protocol
- Socket_port
- Userauto
- Telnet_client_mode

Alarm Parameter (wiz --al)

- Alarm
- xml_monitor

Authentication Parameters (wiz --auth)

- Authtype
- Authhost1
- Accthost1
- Authhost2
- Accthost2
- Radtimeout

- Radretries
- Secret

Data Buffering Parameters (wiz --db)

- Data_buffering
- Conf.nfs_data_buffering
- Syslog_buffering
- Dont_show_DBmenu
- DB_timestamp
- DB_mode
- Syslog_sess

Power Management Parameters (wiz --pm)

- pmkey
- pmNumOfOutlets
- pmoutlet
- pmtype
- pmusers

Appendix K - Wiz Application Parameters

Serial Settings Parameters (wiz --sset <type>)

(CAS profile)

- Speed
- Datasize
- Stopbits
- Parity
- Flow
- Dcd
- SttyCmd
- DTR_reset

(TS profile)

- Speed
- Datasize
- Stopbits
- Parity
- Flow
- Dcd

Appendix K - Wiz Application Parameters

Sniffing Parameters (wiz --snf)

- Admin_users
- Sniff_mode
- Escape_char
- Multiple_sessions

Syslog Parameters (wiz --sl)

- Conf.facility
- Conf.DB_facility

Terminal Appearance Parameters (wiz --tl)

- Issue
- Prompt
- Lf_suppress
- Auto_answer_input
- Auto_answer_output

Appendix K - Wiz Application Parameters

Terminal Server Profile Other Parameters (wiz --tso)

- Host
- Term
- Conf.locallogins

Appendix L - Copyrights

References

The Secure Console Port Server SSH is based in the HardHat Linux distribution, developed by Montavista Software for embedded systems. Additionally, several other applications were incorporated into the product, in accordance with the free software philosophy.

The list below contains the packets and applications used in the Secure Console Port Server SSH and a reference to their maintainers. The copyrights notices required in some packets are placed in the /COPYRIGHTS directory of the Secure Console Port Server SSH image.

Bootparamd

NetKit Bootparamd version 0.17
ftp://ftp.uk.linux.org/pub/linux/Networking/netkit

Busybox

BusyBox version 0.60.2
ftp://ftp.lineo.com/pub/busybox/

Cron

Paul Vixie's cron version 3.0.1.
paul@vix.com

DHCPCD

PhysTech DHCP Client Daemon version 1.3.20.p10.
http://www.phystech.com/download/dhcpcd.html

Flex

Flex version 2.5.4
vern@ee.lbl.gov
COPYRIGHT: This product includes software developed by the University of California, Berkeley and its contributors

Appendix L - Copyrights

GNU

The GNU project
http://www.gnu.org

HardHat Linux

MontaVista Software - HardHat version 1.2
http://www.montavista.com

IPChains

Netfilter IPChains version 1.3.9. Extracted from the HardHat Linux
http://www.netfilter.org

Linux Kernel

Linux Kernel version 2.2.17. Extracted from the HardHat Linux distribution
http://www.kernel.org

NTP

NTP client
http://doolittle.faludi.com/ntpclient/

OpenSSH

OpenSSH version 3.5p1
http://www.openssh.org
COPYRIGHT: This product includes software developed by the University of California, Berkeley and its contributors.

OpenSSL

OpenSSL Project version 0.9.6g
http://www.openssl.org
COPYRIGHT: This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)
COPYRIGHT: This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)

Appendix L - Copyrights

PAM

Linux PAM version 0.75
http://www.kernel.org/pub/linux/libs/pam/

Portslave

SourceForge Portslave project version 2000.12.25. (modified). Includes pppd version 2.4.1 and rlogin version 8.10
http://sourceforge.net/projects/portslave/

RSYNC

rsync version 2.5.5
http://rsync.samba.org/rsync/

Syslog-ng

Syslog new generation version 1.5.17
http://www.balabit.hu/products/syslog-ng/

Tinylogin

TinyLogin version 0.80
ftp://ftp.lineo.com/pub/tinylogin/

WEBS

GoAhead WEBS version 2.1 (modified)
http://goahead.com/webserver/webserver.htm
Copyright (c) 20xx GoAhead Software, Inc. All Rights Reserved

ZLIB

zlib version 1.1.4
http://www.gzip.org/zlib/

This page has been left intentionally blank

List of Figures

List of Figures

List of Figures

List of Figures

This page has been left intentionally blank.

List of Tables

List of Tables

Glossary

Authentication

Authentication is the process of identifying an individual, usually based on a username and password. In security systems, authentication is distinct from authorization, which is the process of giving individuals access to system objects based on their identity. Authentication merely ensures that the individual is who he or she claims to be, but says nothing about the access rights of the individual. (Source: www.webopedia.com)

Break Signal

A break signal is generated in an RS-232 serial line by keeping the line in zero for longer than a character time. Breaks at a serial console port are interpreted by Sun servers as a signal to suspend operation and switch to monitor mode.

Console Access Server (CAS)

A CAS has an Ethernet LAN connection and many RS-232 serial ports. It connects to the console ports of servers and networking equipment and allows convenient and secure access from a single location.

Console Port

Most of the equipment in a data center (servers, routers, switches, UPS, PBX, etc.) has a serial console port for out-of-band management purposes.

Cluster

A cluster is a group of one or more computers working as a group to execute a certain task. From the user standpoint, a cluster acts as a large computer system.

Flash

Flash refers to a type of memory that can be erased and reprogrammed in units of memory known as blocks rather than one byte at a time; thus, making updating to memory easier.

In-band network management

In a computer network, when the management data is accessed using the same network that carries the data, this is called “in-band management.”

Glossary

IP packet filtering

This is a set of facilities in network equipment that allows the filtering of data packets based on source/destination addresses, protocol, TCP port number and other parameters. Packet filtering is one of the main functions of a firewall.

KVM Switch (KVM)

Keyboard-Video-Mouse Switches connect to the KVM ports of many computers and allow the network manager to access them from a single KVM station.

Mainframe

Large, monolithic computer system.

MIBs

Management Information Bases. SNMP-compliant devices, called agents, store data about themselves in MIBs and return this data to the SNMP requesters.

Out-of-band network management

In a computer network, when the management data is accessed through a network that is independent of the network used to carry data, this is called “out-of-band network management.”

Off-line data buffering

This is a CAS feature that allows capture of console data even when there is no one connected to the port.

Profile

Usage setup of the Secure Console Port Server SSH: either as a Console Access Server (CAS), a Terminal Server, or a Remote Access Server.

RADIUS

Protocol between an authentication server and an access server to authenticate users trying to connect to the network.

Glossary

RISC

Reduced Instruction Set Computer. This describes a computer processor architecture that uses a reduced set of instructions (and achieves performance by executing those instructions very fast.) Most UNIX servers (Sun Sparc, HP, IBM RS6000, Compaq Alpha) were designed with a processor using a RISC architecture. The Intel ® x86 architecture.

RS-232

A set of standards for serial communication between electronic equipment defined by the Electronic Industries Association in 1969. Today, RS-232 is still widely used for low-speed data communication.

Secure Shell (SSH)

SSH has the same functionality as Telnet (see definition below), but adds security by encrypting data before sending it through the network.

Server Farm

A collection of servers running in the same location (see Cluster).

SNMP

Short for Simple Network Management Protocol, a set of protocols for managing complex networks. The first versions of SNMP were developed in the early 80s. SNMP works by sending messages, called protocol data units (PDUs), to different parts of a network. SNMP-compliant devices, called agents, store data about themselves in Management Information Bases (MIBs) and return this data to the SNMP requesters. (Source: Webopedia)

Telnet

Telnet is the standard set of protocols for terminal emulation between computers over a TCP/IP connection. It is a terminal emulation program for TCP/IP networks such as the Internet. The Telnet program runs on your computer and connects your PC to a server on the network. You can then enter commands through the Telnet program and they will be executed as if you were entering them directly on the server console. This enables you to control the server and communicate with other servers on the network. To start a Telnet session, you must log in to a server by entering a valid username and password. Telnet is a common way to remotely control Web servers. (from webopedia.com)

Glossary

Terminal Server

A terminal server has one Ethernet LAN port and many RS-232 serial ports. It is used to connect many terminals to the network. Because they have the same physical interfaces, terminal servers are sometimes used as console access servers.

TTY

The UNIX name for the COM (Microsoft) port.

U Rack height unit

A standard computer rack has an internal width of 17 inches. Rack space on a standard rack is measured in units of height (U). One U is 1.75 inches. A device that has a height of 3.5 inches takes 2U of rack space.

Index

Index