# AlterPath Console Server
# User Guide

# Version 2.1.5 Revision 1c

# AlterPath Console Server Version 2.1.5 Revision 1c

October, 2003
Copyright © Cyclades Corporation, 2003

For latest manual revisions, please refer to Cyclades website on:
http://www.cyclades.com/support/downloads.php

Product Version 2.1.5 Revision 1c
Document Number 2.1.5-Draft 28c

# 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

# Table of Contents

## Chapter 4 - Server Management

## 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 - IPSEC

## Appendix H - Web User Management

# Table of Contents

## Appendix I - Connect to Serial Ports from Web

## Appendix J - Power Management

## Appendix K - Examples for Configuration Testing

# Table of Contents

## Appendix L - Wiz Application Parameters

## Appendix M - Copyrights

## Appendix N - AlterPath Bio

## Appendix O - Start and Stop Daemons

# Table of Contents

# Preface

## Purpose

The purpose of this guide is to provide instruction for users to independently install, configure, and maintain the AlterPath Console Server. 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 AlterPath Console Server. 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 AlterPath Console Server 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 AlterPath Console Server using a Web browser, however, New Users can also configure the AlterPath Console Server 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 AlterPath Console Server 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 AlterPath Console Server should be connected and what each cable is used for. It describes the basic configuration process to get the AlterPath Console Server up and running for its most common uses.

- **Chapter 3 - Additional Features** is dedicated to users wanting to explore all available features of the AlterPath Console Server. It provides configuration instructions for syslog, data buffers, authentication, filters, DHCP, NTP, SNMP, clustering, and sniffing.

- **Chapter 4 - Server Management** explains how to manage the AlterPath Console Server through your browser or via the Command Line Interface. It also instructs the user on sending warning messages.

- **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 AlterPath Console Server.

- **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.In addition, instruction is provided for connection to serial ports through the browser to view the server screen.

- **Appendix F - Certificate for HTTP Security** provides configuration information that will enable you to obtain a Signed Digital Certificate.

- **Appendix G - IPSEC** provides encryption and authentication services at the IP (Internet Protocol) level of the network protocol stack.

# Preface

- **Appendix H - Web User Management** covers default and optional configuration, and the addition/deletion of users, groups, and access limits.

- **Appendix I - Connect to Serial Ports from Web** enables this process, based on how the serial port is configured.

- **Appendix J - Power Management** instructs on using your AlterPath Console Server with various IPDUs.

- **Appendix K - Examples for Configuration Testing** provides examples for testing the Alter-Path Console Server after configuration.

- **Appendix L - Wiz Application Parameters** contains all basic and custom wizard parameters.

- **Appendix M - Copyrights** lists details about applications that were incorporated into the product.

- **Appendix N - AlterPath Bio** describes the installation and setup of a Cyclades solution that integrates biometric authentication while increasing security on the AlterPath Console Server.

- **Appendix O - Start and Stop Daemons** describes a feature that allows daemons (services) to be enabled or disabled without need of reboot the TS or ACS.

- The **Glossary** provides definitions for commonly-used terms in this manual.

## Additional Documentation and Help

There are other Cyclades documents that contain background information about Console Port Management and the Cyclades product line. These are:

- Cyclades' *Console Management in the Data Center*

- Cyclades' *Product Catalog*

For the most updated version of Cyclades' documentation, use the following Web address:

http://www.cyclades.com/support/downloads.php

# Preface

## Technical Support Centers

To reach Cyclades' Technical Support Centers, go to the following:

http://www.cyclades.com/support/technical_support.php

## 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 <u>underlined</u> (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 <u>underlined and slightly larger</u> than the rest of the text. These terms have a hypertext link to the glossary.

### Quick Steps

Step-by-step instructions for installing and configuring the AlterPath Console Server 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:

# Preface

**Step 1: Modify files.**

You will modify four Linux files to let the AlterPath Console Server 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.

# Preface

**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 AlterPath Console Server.

**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.

# Preface

This page has been left intentionally blank.

AlterPath Console Server

# Introduction and Overview

## Introducing Cyclades

Cyclades is a data center fault management company that enables remote management of servers, network equipment and automation devices. Its products help data center managers at enterprise, telecommunication and Internet companies to maximize network and server availability. This results in decreased maintenance costs, increased efficiency and productivity, along with greater control, freedom and peace of mind. Cyclades' advantage is providing scalable products leveraging Linux technology for flexibility and ease of customization.

## The AlterPath Console Server

The AlterPath Console Server is line of Console Access and Terminal Servers that allow both local and dial-in access for in-band and out-of-band network management.They run an embedded version of the Linux operating system. Configuration of the equipment is done by editing a few plain-text files, and then updating the versions of the files on the AlterPath Console Server. 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 "box profile" of the product is that of a Console Access Server.

You can access the AlterPath Console Server via three methods:

- A console directly connected to the AlterPath Console Server

- 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

# Introduction and Overview

There are two models of the AlterPath Console Server: one with a dual power supply and two PCMCIA slots, and one with a single power supply and two PCMCIA slots.

With the AlterPath Console Server set up as a Console Access Server, you can access a server connected to the ACS 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.

## What's in the box

There are several models of the AlterPath Console Server with differing numbers of serial ports. Cyclades will ship either Cable Package #1 or #2 with the product according to current availability.

RJ-45 straight-through cable

RJ-45 to DB-9 F adapter

Cyclades/Sun Netra adapter

CAT.5e INLINE COUPLER

RJ-45 to DB-25 Male adapter

RJ-45 to DB-25 Female adapter
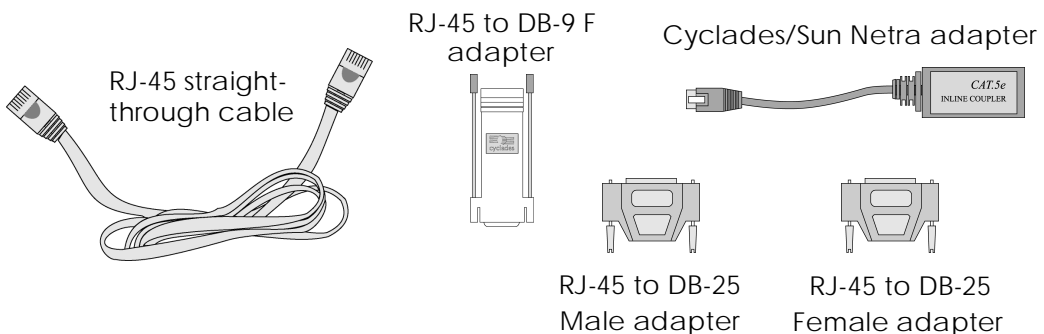
*Figure 1:  Cable Package #1*

# Introduction and Overview



Cyclades/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
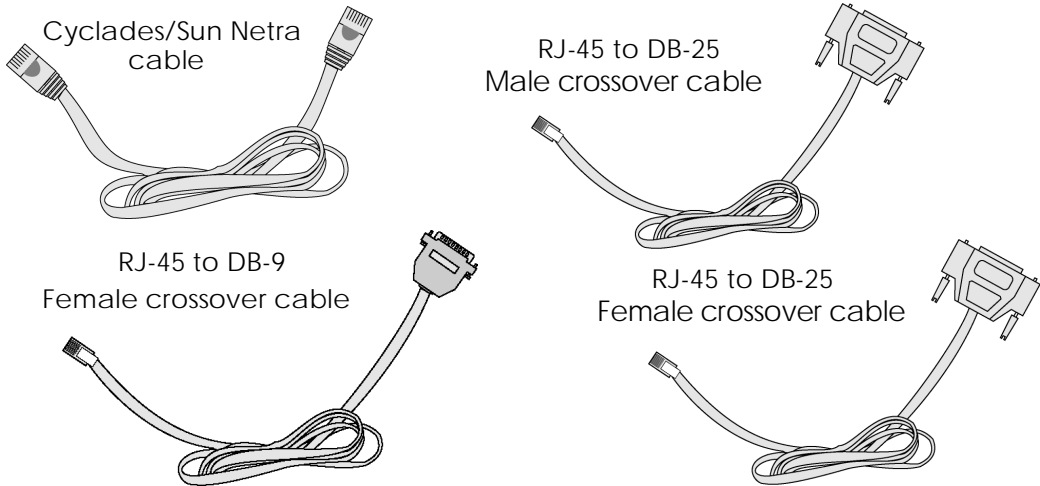
*Figure 2: Cable Package #2*

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

# Introduction and Overview

Power Cord Sockets          Power Switch

Serial Ports          Console Port

Ethernet Port

Mounting Kit

Modem Cable
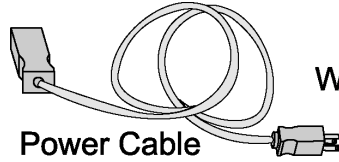
Power Cable

Wall Outlet
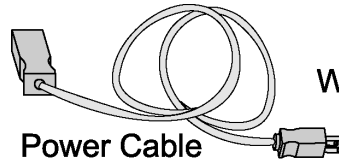
Power Cable

Wall Outlet
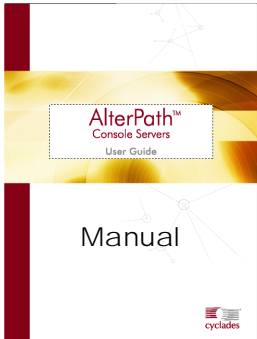
Manual

Loop-back Connector

Cable Package #1          OR          Cable Package #2

*Figure 3:  The AlterPath ACS48, its cables, connectors and other box contents*

# Introduction and Overview



**Figure 4: The AlterPath ACS32, its cables, connectors and other box contents**

Modem Cable

Wall Outlet

Power Cable

Mounting Kit

Wall Outlet

Power Cable

Loop-back
Connector

Manual
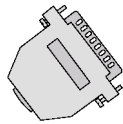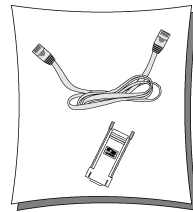
OR
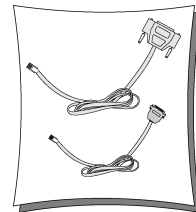
Cable Package #1    Cable Package #2

# Introduction and Overview



**Figure 5: The AlterPath ACS16, its cables, connectors and other box contents**

# Introduction and Overview
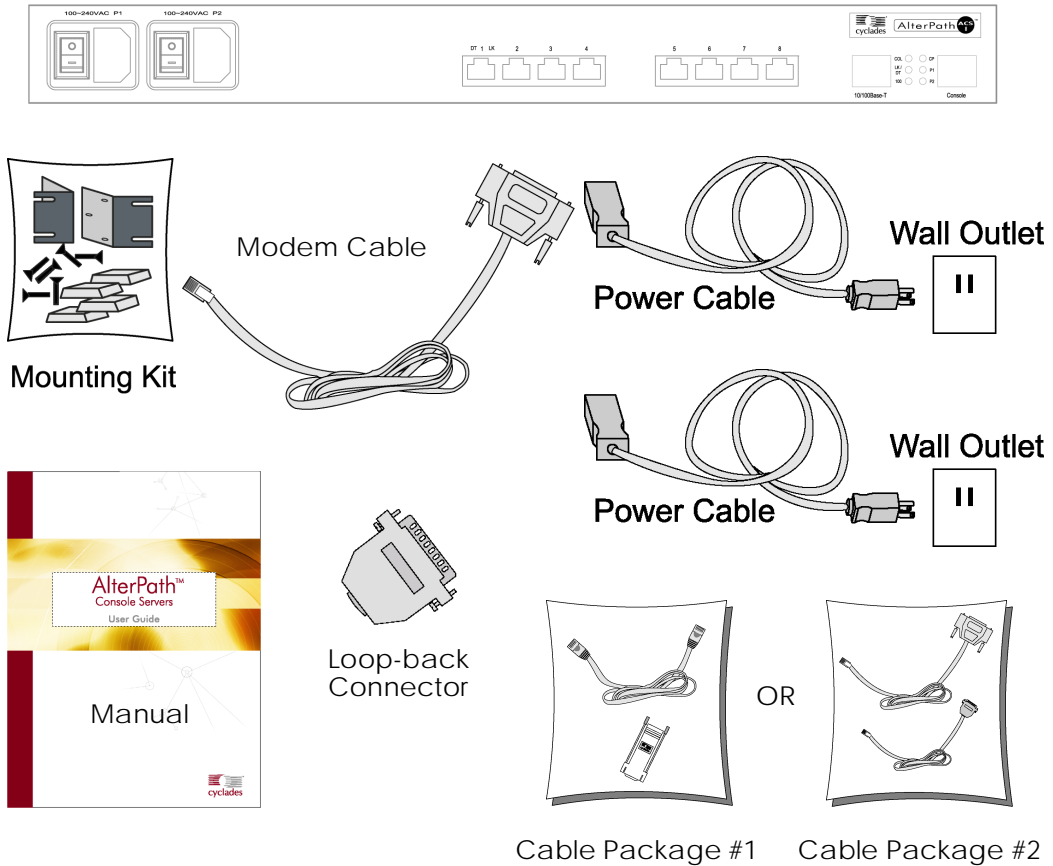


**Figure 6: The AlterPath ACS8, its cables, connectors and other box contents**

**AlterPath Console Server**

# Introduction and Overview
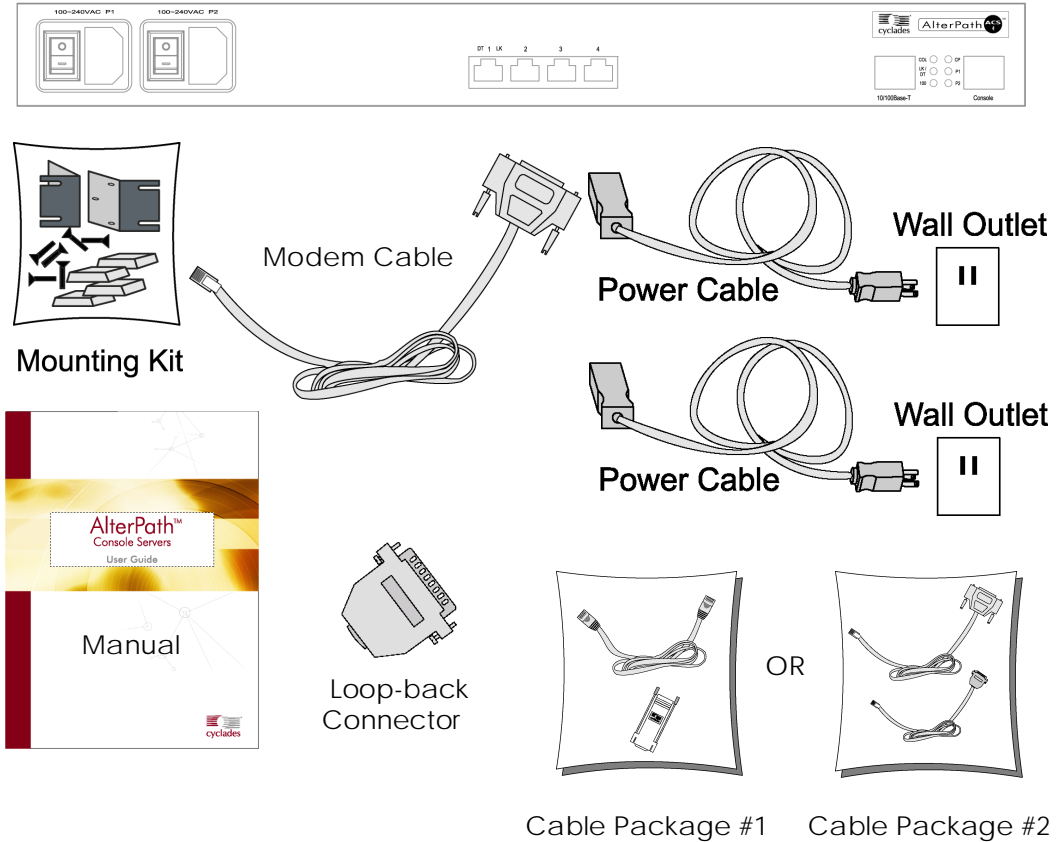


*Figure 7: The AlterPath ACS4, its cables, connectors and other box contents*

# Introduction and Overview

Side View

Top View

Terminal Block

External Power Supply

To Wall Outlet

Loop-back
Connector

Manual

DB-9 Female to
DB-25 Male connector

DB-9 Female

DB-25 Female

Cable Package #1

Crossover (console) cable
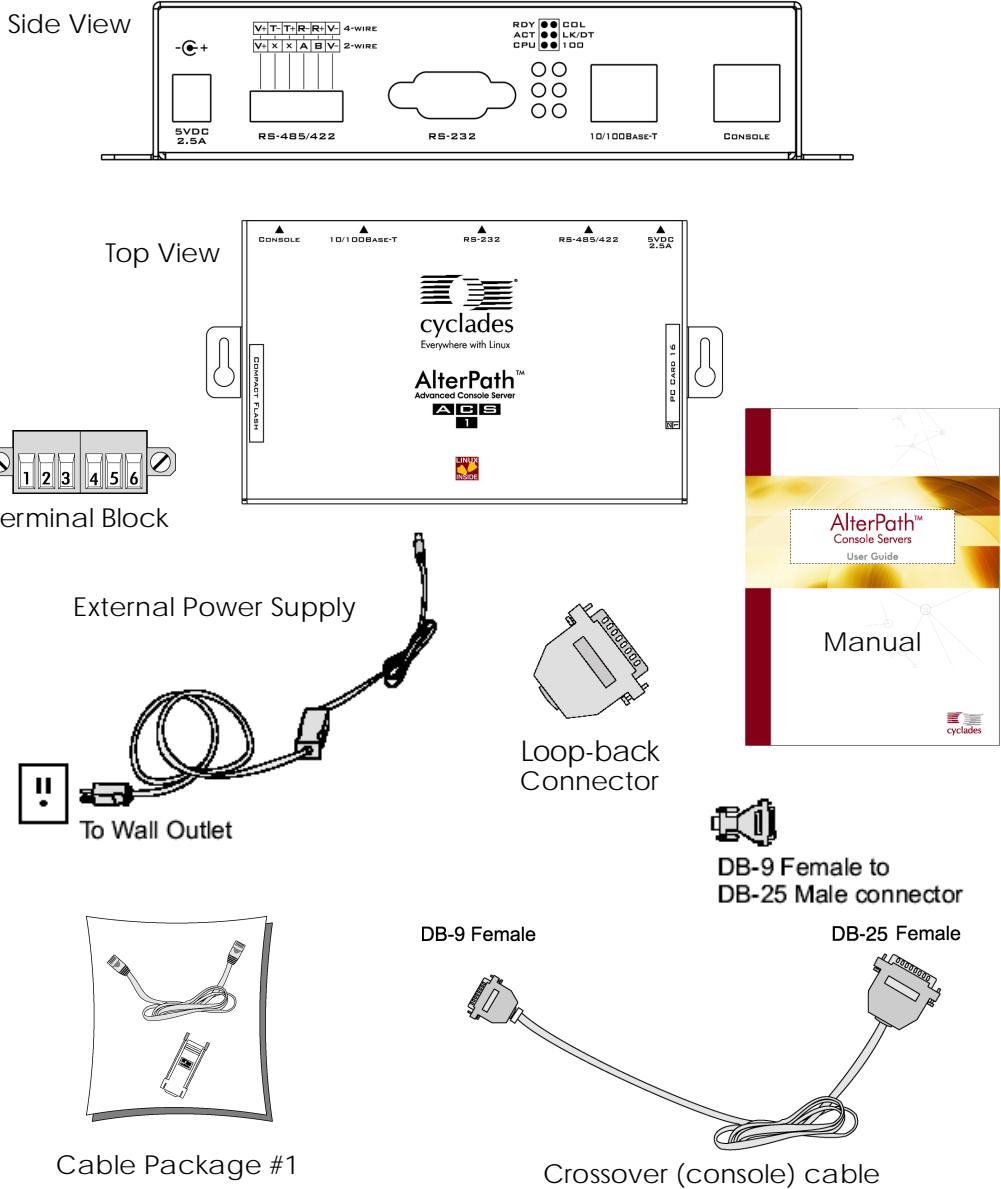
*Figure 8: The ACS1 and cables*

# Introduction and Overview

## Safety Instructions

**Read all the following safety guidelines to protect yourself and your AlterPath Console Server.**

**DANGER!**  Do not operate your AlterPath Console Server with the cover removed.

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

**DANGER!**  To help prevent electric shock, plug the AlterPath Console Server into a properly grounded power source. The cable is equipped with a three-prong plug to help ensure proper grounding. Do not use adapter plugs or remove the grounding prong from the cable. If you have to use an extension cable, use a three-wire cable with properly grounded plugs.

**Important!**  To help protect the AlterPath Console Server 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 AlterPath Console Server and that they are not located where they can be stepped on or tripped over.

# Introduction and Overview

**Important!** Do not spill food or liquids on the AlterPath Console Server. If it gets wet, contact Cyclades.

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

**Important!** Keep your AlterPath Console Server away from heat sources and do not block cooling vents.

**Important!** The AlterPath Console Server 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 AlterPath Console Server DC version. Minimum Breaker Rating: 2A. Required conductor size: 18 AWG.

**AlterPath Console Server**

# Introduction and Overview

## Working inside the AlterPath Console Server

Do not attempt to service the AlterPath Console Server yourself, except when following instructions from Cyclades Technical Support personnel. In the latter case, first take the following precautions:

- Turn the AlterPath Console Server off.

- Ground yourself by touching an unpainted metal surface on the back of the equipment before touching anything inside it.

## Battery

 **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.

**AlterPath Console Server**

# Introduction and Overview

## Federal Communications Commission & Industry Canada Radio Frequency Interference Statements

This equipment generates, uses, and can radiate radio-frequency energy, and if not installed and used properly, that is, in strict accordance with the manufacturer's instructions, may cause interference to radio communication.

It has been tested and found to comply with the limits for a Class A computing device in accordance with the specifications in Subpart B of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference when the equipment is operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference, in which case the user at his own expense will be required to take whatever measures may be necessary to correct the interference.

Changes or modifications not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

This digital apparatus does not exceed the Class A limits for radio noise emission from digital apparatus set out in the Radio Interference Regulation of Industry Canada.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de classe A prescrites dans le Règlement sur le brouillage radioélectrique publié par Industrie Canada.

## Notice about FCC compliance for all AlterPath ACS models

In order to comply with FCC standards the AlterPath Console Server require the use of a shielded CAT 5 cable for the Ethernet interface. Notice that this cable is not supplied with either of the products and must be provided by the customer.

## Normas Oficiales Mexicanas (NOM) Electrical Safety Statement

Instrucciones de Seguridad

1.  Todas las instrucciones de seguridad y operación deberán ser leídas antes de que el aparato eléctrico sea operado.

2.  Las instrucciones de seguridad y operación deberán ser guardadas para referencia futura.

# Introduction and Overview

3. Todas las advertencias en el aparato eléctrico y en sus instrucciones de operación deben ser respetadas.

4. Todas las instrucciones de operación y uso deben ser seguidas.

5. El aparato eléctrico no deberá ser usado cerca del agua-por ejemplo, cerca de la tina de baño, lavabo, sótano mojado o cerca de una alberca, etc..

6. El aparato eléctrico debe ser usado únicamente con carritos o pedestales que sean recomendados por el fabricante.

7. El aparato eléctrico debe ser montado a la pared o al techo sólo como sea recomendado por el fabricante.

8. Servicio-El usuario no debe intentar dar servicio al equipo eléctrico más allá a lo descrito en las instrucciones de operación. Todo otro servicio deberá ser referido a personal de servicio calificado.

9. El aparato eléctrico debe ser situado de tal manera que su posición no interfiera su uso. La colocación del aparato eléctrico sobre una cama, sofá, alfombra o superficie similar puede bloquea la ventilación, no se debe colocar en libreros o gabinetes que impidan el flujo de aire por los orificios de ventilación.

10. El equipo eléctrico deber ser situado fuera del alcance de fuentes de calor como radiadores, registros de calor, estufas u otros aparatos (incluyendo amplificadores) que producen calor.

11. El aparato eléctrico deberá ser connectado a una fuente de poder sólo del tipo descrito en el instructivo de operación, o como se indique en el aparato.

12. Precaución debe ser tomada de tal manera que la tierra fisica y la polarización del equipo no sea eliminada.

13. Los cables de la fuente de poder deben ser guiados de tal manera que no sean pisados ni pellizcados por objetos colocados sobre o contra ellos, poniendo particular atención a los contactos y receptáculos donde salen del aparato.

14. El equipo eléctrico debe ser limpiado únicamente de acuerdo a las recomendaciones del fabricante.

15. En caso de existir, una antena externa deberá ser localizada lejos de las lineas de energia.

16. El cable de corriente deberá ser desconectado del cuando el equipo no sea usado por un largo periodo de tiempo.

AlterPath Console Server

# Introduction and Overview

17. Cuidado debe ser tomado de tal manera que objectos liquidos no sean derramados sobre la cubierta u orificios de ventilación.

18. Servicio por personal calificado deberá ser provisto cuando:

    • El cable de poder o el contacto ha sido dañado; u

    • Objectos han caído o líquido ha sido derramado dentro del aparato; o

    • El aparato ha sido expuesto a la lluvia; o

    • El aparato parece no operar normalmente o muestra un cambio en su desempeño; o

    • El aparato ha sido tirado o su cubierta ha sido dañada.

## Aviso de Precaución S-Mark Argentina

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



**¡Peligro!** No hacer funcionar el AlterPath Console Server con la tapa abierta.



**¡Peligro!** Para prevenir un corto circuito en el AlterPath Console Server 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.



**¡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.

# Introduction and Overview

⚠ **¡Importante!**  Para proteger al AlterPath Console Server de fluctuaciones en corriente eléctrica, utilice una fuente eléctrica de respaldo.

⚠ **¡Importante!**  Asegurarse de que nada descanse sobre los cables del AlterPath Console Server, y que los cables no obstruyan el paso.

⚠ **¡Importante!**  Asegurarse de no dejar caer alimentos o bebidas en el AlterPath Console Server. Si esto ocurre, avise a Cyclades Corporation.

💥 **¡Peligro!**  No empuje ningún tipo de objeto en los compartimientos del Alter-Path Console Server. Hacer esto podría ocasionar un incendio o causar un corto circuito dentro del equipo.

⚠ **¡Importante!**  Mantenga el AlterPath Console Server fuera del alcancé de calentadores, y asegurarse de no tapar la ventilación del equipo.

# Introduction and Overview

 **¡Importante!** El 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 AlterPath Console Server. El limite mínimo del interruptor deberá ser 2 amperes, con conductor de 0.75 mm (18 AWG).

## Trabajar dentro del AlterPath Console Server

No intente dar servicio al AlterPath Console Server, solo que este bajo la dirección de Soporte Técnico de Cyclades Corporation. Si este es el caso, tome las siguientes precauciones:

Apague el AlterPath Console Server. 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.

.

# Introduction and Overview

This page has been left intentionally blank.

# Chapter 2 - Installation, Configuration, Usage

## Introduction

This chapter will allow you to install and configure the AlterPath Console Server 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 AlterPath Console Server 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 <u>Appendix A - New User Background Information</u>. *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:

- <u>System Requirements</u>

- <u>Default Configuration Parameters</u>

- <u>Pre-Install Checklist</u>

- <u>Task List</u>

- <u>The Wizard</u>

- <u>Quick Start</u>

- <u>The Installation and Configuration Process</u>

## System Requirements

Cyclades recommends either of the following specifications for configuration of the AlterPath Console Server:

- 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 AlterPath Console Server:

| | |
|---|---|
| *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, AlterPath Console Server, NameServer, and Gateway* | You will need to locate the IP address of your PC or workstation, the AlterPath Console Server, 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 AlterPath Console Server 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 AlterPath Console Server:

Task 1: Connect the AlterPath Console Server 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 AlterPath Console Server.

## The Wizard

The eight key tasks can also be done through a wizard in the 2.1 plus versions of the AlterPath Console Server.

### 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 AlterPath Console Server 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**

- Power Management (see Appendix J - Power Management)

- **Serial Settings**

- **Session Sniffing**

- **Syslog**

- **Terminal Appearance**

- **TS Setup Wizard** (These are additional configuration parameters applied only to the TS profile.)

# Chapter 2 - Installation, Configuration, Usage

## Quick Start

This Quick Start gives you all the necessary information to quickly configure and start using the AlterPath Console Server 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 AlterPath Console Server 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 AlterPath Console Server. 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 AlterPath Console Server with the RJ-45 connector end, and to your PC's available COM port with the serial port end.

**AlterPath Console Server**

# Chapter 2 - Installation, Configuration, Usage

**Step 2: Power on the AlterPath Console Server.**

After the AlterPath Console Server 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.

```
********************************************************
********* 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...
```

You will want to configure the following settings:

- Hostname

- DHCP enabled/disabled

- System IP (if DHCP is disabled)

- Domain Name

- Primary DNS Server

# Chapter 2 - Installation, Configuration, Usage

- **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 : cyclades.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 AlterPath Console Server 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 AlterPath Console Server features, either continue configuration using the vi editor from the console or use a browser from a workstation and point to the AlterPath Console Server, or use the CLI, if appropriate (as indicated in the relevant sections in Chapter 3).

**AlterPath Console Server**

# Chapter 2 - Installation, Configuration, Usage

## Configuration using a Web browser

The AlterPath Console Server box 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 box using your browser:

**Step 1: Connect Hub to workstation and ACS.**

Your workstation and your ACS must be on the same physical LAN. Connect one RJ-45 cable from the Ethernet port of the ACS 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 ACS IP.**

From the workstation, issue a command to add a route pointing to the network IP address of the ACS (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.

# Chapter 2 - Installation, Configuration, Usage
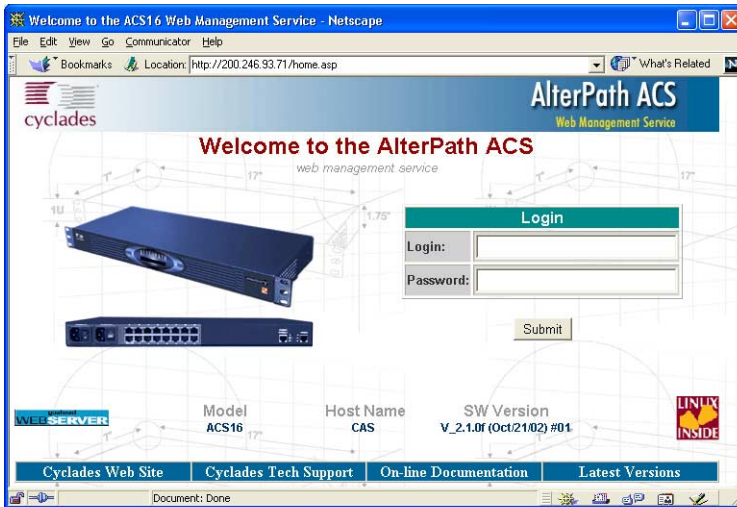


*Figure 9:  Login page of Web Configuration Manager*

**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:

# Chapter 2 - Installation, Configuration, Usage



Link for
changing
password
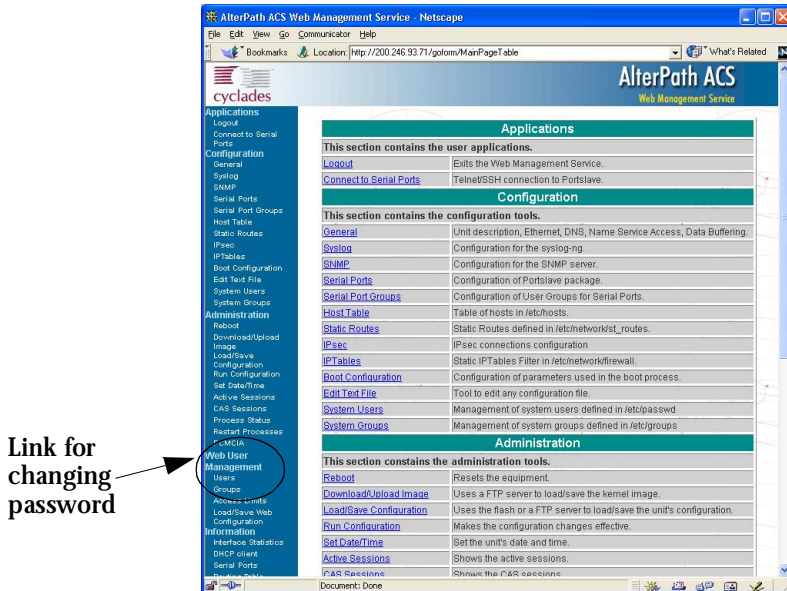
*Figure 10: Configuration & Administration Menu page*

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 H - Web User Management.

**Step 6: Click on the General link.**

# Chapter 2 - Installation, Configuration, Usage
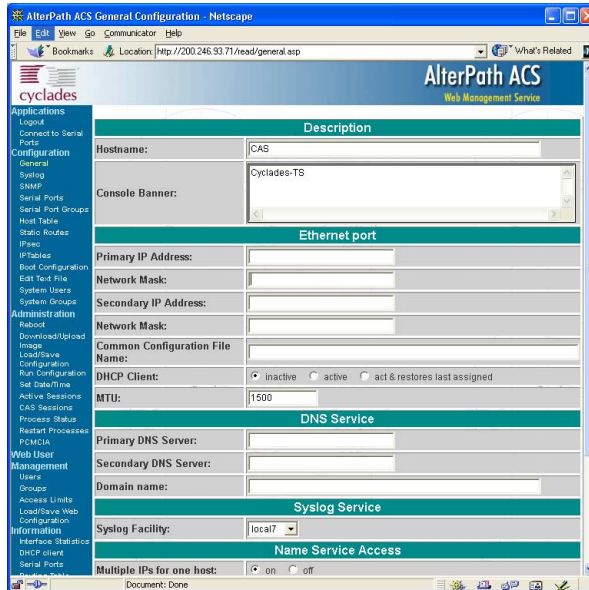


*Figure 11:  General page*

**Step 7:  Configure parameters presented in the fields.**

**Step 8:  Click on the Submit button.**

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 9:  Click on the Save Configuration to Flash button.**

> The configuration was saved in flash. The new configuration will be valid and run-
> ning. The AlterPath Console Server 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
```

# Chapter 2 - Installation, Configuration, Usage

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 AlterPath Console Server features, either continue configuration using your 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 |

# Chapter 2 - Installation, Configuration, Usage

**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 |
| *IPsec* | IPsec connections configuration |
| *IP Tables* | Static IPTables Filter in /etc/network/firewall |
| *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 |

AlterPath Console Server

# Chapter 2 - Installation, Configuration, Usage

**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 the ACS' configuration |
| *Run Configuration* | Makes the configuration changes effective |
| *Set Date/Time* | Set the ACS' date and time |
| *Active Sessions* | Shows the active sessions |
| *CAS Sessions* | Shows the CAS 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 |

# Chapter 2 - Installation, Configuration, Usage

**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 the ACS File System status |
| *System Information* | Shows information about the kernel, time, CPU, and memory |

**Note:** The link Connect to Serial Ports is only available for all ACS models. See "Appendix I - Connect to Serial Ports from Web" on page 451.

# Chapter 2 - Installation, Configuration, Usage

## Configuration using Telnet

The AlterPath Console Server box 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 box using telnet:

Step 1: **Connect Hub to workstation and ACS.**

Your workstation and your ACS must be on the same physical LAN. Connect one RJ-45 cable from the Ethernet port of the ACS 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 ACS IP.**

From the workstation issue a command to add a route pointing to the network IP address of the ACS (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.**

# Chapter 2 - Installation, Configuration, Usage

**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 : cyclades.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 AlterPath Console Server 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 AlterPath Console Server to the Network and other Devices

#### Power Users

Connect a PC or terminal to the AlterPath Console Server 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 AlterPath Console Server boots properly, a login banner will appear. Log in as *root* (default password is *tslinux*). 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 <u>Task 4: Edit the pslave.conf file</u>.

---

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

---
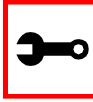
> 🔧 **Note:** If your terminal does not have ANSI emulation, select vt100; then, on the ACS, log in as root and switch to vt100 by typing:
>
> ```
> TERM=vt100;export TERM
> ```

# 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 AlterPath Console Server directly through your PC's COM port connected with the AlterPath Console Server 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 AlterPath Console Server.

After the initial configuration through the HyperTerminal connection, you will be connecting your PC (or another terminal) to the AlterPath Console Server via an Ethernet connection in order to manage the ACS. The workstation used to access the ACS 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 ACS' console port.

• PC (Ethernet): Ethernet port connects via hub to the ACS' Ethernet port.

• Use the HyperTerminal to configure the box.

• Use the PC Ethernet to access the box as client (telnet/ssh).

# Chapter 2 - Installation, Configuration, Usage

**Step 1: Plug the power cable into the AlterPath Console Server.**

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

**DANGER!** To help prevent electric shock, plug the AlterPath Console Server 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.

**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 AlterPath Console Server 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 AlterPath Console Server.**

Your workstation and ACS must be on the same physical LAN. Connect one RJ-45 cable from the Ethernet port of the ACS 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

AlterPath Console Server

# Chapter 2 - Installation, Configuration, Usage

## 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.
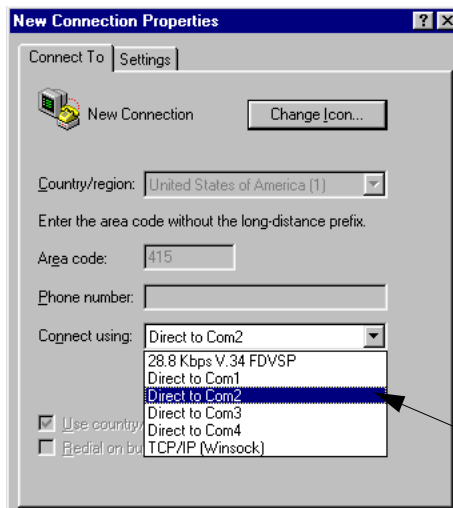


*Figure 12:  Choose a free COM port*

**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).
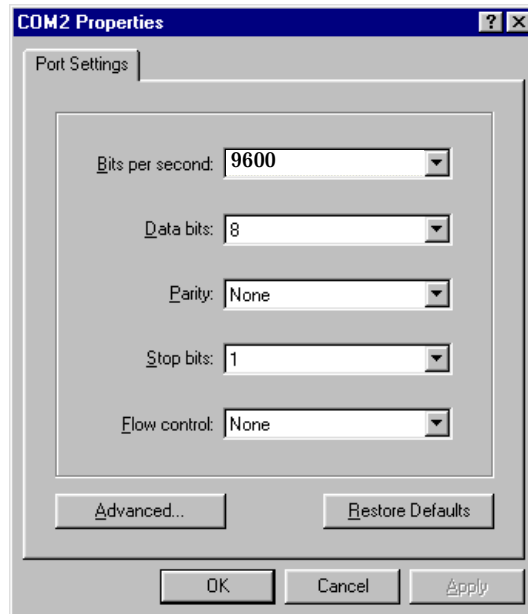
*Figure 13: Port Settings*

**Step 3: Power on the AlterPath Console Server.**

**Step 4: Click OK on the Properties window.**

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

# Chapter 2 - Installation, Configuration, Usage

## Task 3: Modify the System Files

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

```
/etc/hostname
```

```
/etc/hosts
```

```
/etc/resolv.conf
```

```
/etc/network/st_routes
```

The four Linux files must be modified to identify the ACS 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 AlterPath Console Server 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 AlterPath Console Server. (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 AlterPath Console Server.) 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.)
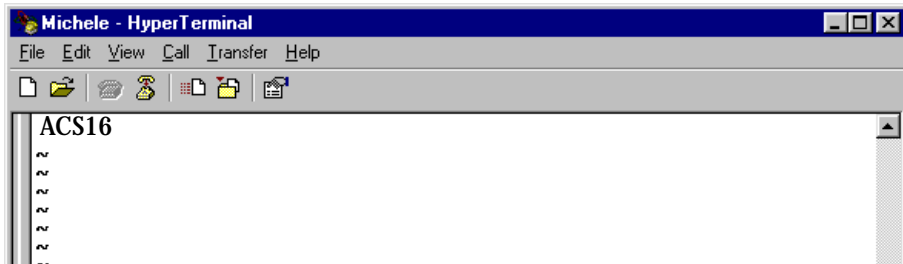
*Figure 14:  The /etc/hostname file with hostname typed in*

**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.
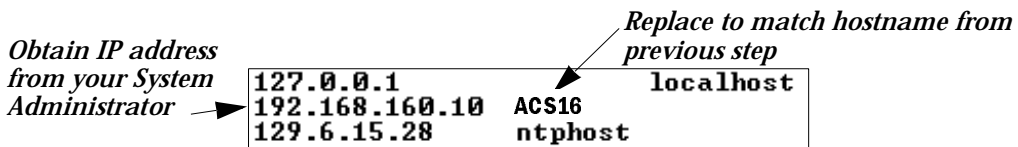


*Figure 15:  Contents of the /etc/hosts file*

**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
```

# Chapter 2 - Installation, Configuration, Usage

**Step 6:  Modify /etc/network/st_routes.**

The fourth file defines static routes. In the console server example in [Figure 61: Console Access Server diagram](#) 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 AlterPath Console Server 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
```

# Chapter 2 - Installation, Configuration, Usage

> **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 AlterPath Console Server. 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.

# Chapter 2 - Installation, Configuration, Usage

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
```

# Chapter 2 - Installation, Configuration, Usage

*all.authtype*   This parameter controls the authentication required by the AlterPath Console Server. 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 user-name 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)

AlterPath Console Server

*all.authtype (cont.)*

- *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)

- *kerberos* (authentication is performed using a kerberos server. The IP address and other details of the kerberos server are defined in the file /etc/krb5.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)

- *TacacsPlusDownLocal* (local authentication is tried only when the TacacsPlus server is down)

- *RadiusDownLocal* (local authentication is tried only when the Radius server is down)

- *kerberosDownLocal* (local authentication is tried only when the kerberos server is down)

- *ldapDownLocal* (local authentication is tried only when the ldap server is down)

- *NIS* - All authentication types but NIS follow the format all.authtype <Authentication>DownLocal or <Authentication> (e.g. all.authtype radius or radiusDownLocal or ldap or ldapDownLocal, etc). NIS requires all.authtype to be set as local, regardless if it will be "nis" or its "Downlocal" equivalent. The service related to "nis" or its "Down-local" equivalent would be configured in the /etc/nsswitch.conf file, not in the /etc/portslave/pslave.conf file.

An example value would be:
```
radius
```

# Chapter 2 - Installation, Configuration, Usage

*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 ACS from a DOS prompt.**
Open a DOS window, type in the following, and then press Enter:

```
ping <IP assigned to the ACS by DHCP or you>
```
An example would be:

```
ping 192.168.160.10
```
If you receive a reply, your ACS 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 AlterPath Console Server.**
*(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 ACS 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 AlterPath Console Server

**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 AlterPath Console Server is booted.

# 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). One can access the serial port by statically addressing it (using TCP port number, alias name or IP address) or just access the next free serial port available from an existent pool (by using the pool's TCP port number, alias or IP address). 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 or the first free serial port belonging to a pool of serial ports, 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   (Ethernet's interface) configured by the user or learned from DHCP.

<TCP port number> is the number associated to the serial port or pool of serial ports. From factory, 7001 corresponds to serial port 1, 7002 to serial port 2 and so forth, and 3000 is a pool with all serial ports.

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 or the next free serial port from a pool, 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.

AlterPath Console Server

# Chapter 2 - Installation, Configuration, Usage

<Server> can be just the TCP port number assigned for that serial port (7001, 7002, etc), pool of ports (3000, etc), the alias for the server connected to that serial port or the alias of a pool of ports.

<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  (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
```

# Chapter 2 - Installation, Configuration, Usage

```
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/SSH 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/SSH to your unit, use the  -e  option. So for example, to set Ctrl-? as the escape character, type:

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

```
ssh -e ^? user1@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. To exit from the entire SSH session to your unit, type the escape character you had set plus character "."(dot)

**AlterPath Console Server**

# 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**

- **Centralized Management**

- **Clustering**

- **CronD**

- **Data Buffering**

- **DHCP**

- **Dual Power Management**

- **Filters and Network Address Translation**

- **Generating Alarms**

- **Help**

- **NTP**

- **PCMCIA**

- **Ports Configured as Terminal Servers**

- **Serial Settings**

- **Session Sniffing**

- **SNMP**

- **Syslog**

- **TCP Keepalive**

- **Terminal Appearance**

- **Time Zone**

## Configuration Wizard - Basic Wizard

The configuration wizard application is a quicker and easier way to configure the AlterPath Console Server. 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 ACS.

The command *wiz* gets you started with some basic configuration. After executing this command, you can continue the configuration of the ACS 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 *********
********************************************************
```

# Chapter 3 - Additional Features

```
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 Cyclades defaults, or

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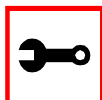
**Tip.** On most of the following configuration screens, the default or current value of the parameter is displayed inside brackets. Just press the ENTER key if you are satisfied with the value in the brackets. If not, enter the appropriate parameter and press ENTER.

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 your AC S that allows you to refer to the ACS 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 ACS. 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]:
```

# Chapter 3 - Additional Features

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

**Step 6:** If DHCP client is disabled, enter IP Address of your ACS 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 the ACS within your network. See your network administrator to obtain a valid IP address for the ACS.

```
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[#]: cyclades.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: cyclades.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: cyclades.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.

# Chapter 3 - Additional Features

**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 AlterPath Console Server 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 ACS 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 AC S.

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 ACS configurations using the Web browser by typing in the IP address of the ACS.

## 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)

The step-by-step process to configuring ports for a specific profile appear in the following sections, and the exact screen flow begins with Figure 16: Configuration and Administration page.

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(Figure 18: 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

# Chapter 3 - Additional Features

- 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 AlterPath Console Server (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 AlterPath Console Server'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 AlterPath Console Server 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 AlterPath Console Server for this period of time, the AlterPath Console Server 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. |

# Chapter 3 - Additional Features

| | |
|---|---|
| *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, and raw_data). 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:<br><group name>:<user1>{,<user2>[,<user3>]]<br>Example: conf.group group_name: user1, user2. |
| *s<n>.serverfarm* | Alias name given to the server connected to the serial port. Server_connected.<br>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.

**Serial Ports links**

**Link Panel**



*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.



*Figure 17:  Port Selection page*

**Step 4:  Select port(s).**
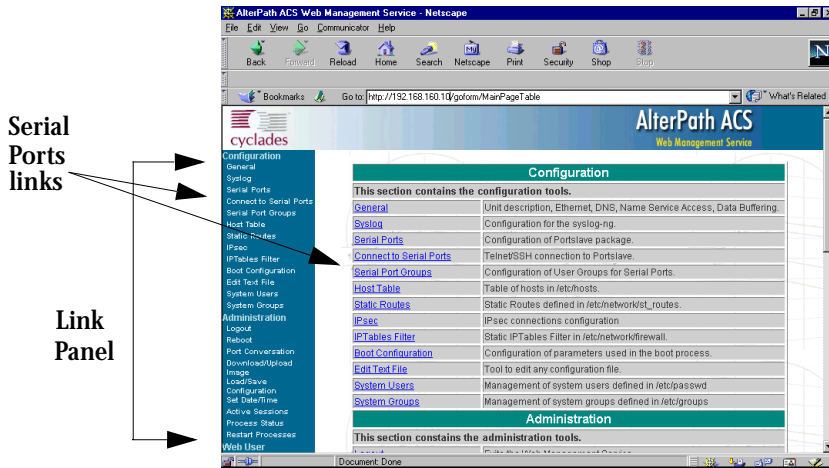
On the Port Selection page, choose all ports or an individual port from the dropdown menu. This will take you to the Serial Port Configuration page.

# Chapter 3 - Additional Features



CAS profile
button
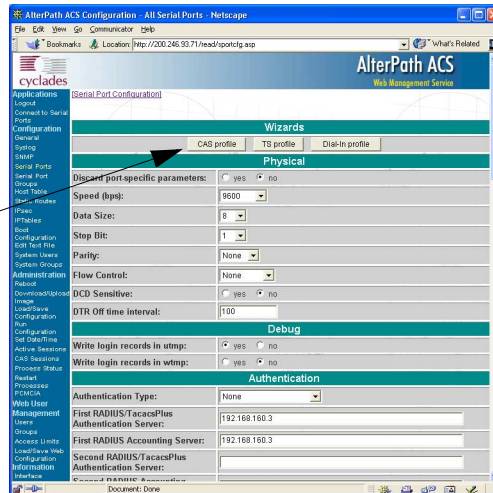
*Figure 18: Serial Port Configuration page*

**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.
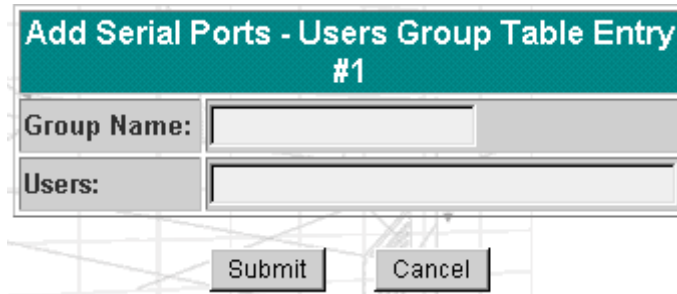
# Chapter 3 - Additional Features

**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.



*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 : #
```

# Chapter 3 - Additional Features

```
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, or raw data.
(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.)

all.protocol[socket_server] :


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

# Chapter 3 - Additional Features

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.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.*

# Chapter 3 - Additional Features

***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] :
```

# Chapter 3 - Additional Features

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 <num-
ber>
```

**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 ACS 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*<br>*(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,** <u>"Browser Method" on page 85</u>.

**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:

# Chapter 3 - Additional Features

**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 N   W 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] :

# Chapter 3 - Additional Features

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.

# Chapter 3 - Additional Features

*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>
```
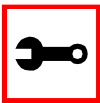
---

**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 AlterPath Console Server, 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:
```
%i:%j novj \
proxyarp modem asyncmap 000A0000 \
noipx noccp login auth require-pap refuse-chap\
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, ppp_only, and cslip.**

**Example value: using PAP**
```
%i:%j novj \
proxyarp modem asyncmap 000A0000 \
noipx noccp login auth require-pap refuse-chap\
mtu %t mru %t \
cb-script /etc/portslave/cb_script \
plugin /usr/lib/libpsr.so
```

**Example value: using CHAP**
```
%i:%j novj \
proxyarp modem asyncmap 000A0000 \
noipx noccp login auth require-chap refuse-pap\
mtu %t mru %t \
cb-script /etc/portslave/cb_script \
plugin /usr/lib/libpsr.so
```

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

# Chapter 3 - Additional Features

## 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 85.

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>
```
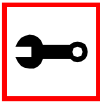
---

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 AlterPath Console Server, authentication can be performed locally, or with a remote Radius, Tacacs, or ldap database, or kerberos.

### 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)

| | |
|---|---|
| *all.authtype (cont.)* | • *kerberos* (authentication is performed using a kerberos server. The IP address and other details of the kerberos server are defined in the file /etc/krb5.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) |
| | • *kerberosDownLocal* (local authentication is tried only when the kerberos server is down) |
| | • *ldapDownLocal* (local authentication is tried only when the ldap server is down) |
| | • *NIS NIS* - All authentication types but NIS follow the format all.authtype <Authentication>DownLocal or <Authentication> (e.g. all.authtype radius or radiusDownLocal or ldap or ldapDownLocal, etc). NIS requires all.authtype to be set as local, regardless if it will be "nis" or its "Downlocal" equivalent. The service related to "nis" or its "Downlocal" equivalent would be configured in the /etc/nsswitch.conf file, not in the /etc/portslave/pslave.conf file. |
| | Note that this parameter controls the authentication required by the AlterPath Console Server. 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. |

# Chapter 3 - Additional Features

| | |
|---|---|
| *all.accthost1*<br>*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 AlterPath Console Server and the Radius/TacacsPlus servers. |
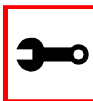
> **Note:** If you want to dial in to the serial port on an ACS series with CHAP authentication, you need to do the following:
>
> 1.Configure Sxx.authtype as local.
>
> 2.Add users in ACS.
>
> 3.Insert the users in the file /etc/ppp/chap-secrets.
>
> 4.Insert the file /etc/ppp/chap-secrets in the file /etc/config_files.
>
> 5.Execute the *saveconf* command.

## 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 85.

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
```

# Chapter 3 - Additional Features

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, kerberos, etc.


all.authtype[none] :
```
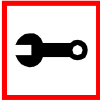
> Note:  If authtype is configured as *none*, *local*, *ldap*, or *kerberos* the applica-
> tion will skip immediately to the summary screen because the rest of the
> parameters pertain only if the system is configured to use a Radius or Tacacs-
> Plus server. Configurations for ldap and kerberos are done in /etc/ldap.conf
> and /etc/krb5.conf, respectively.

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

all.authhost1[200.200.200.2] :
```

# Chapter 3 - Additional Features

*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] :
```

# Chapter 3 - Additional Features

*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] :

# Chapter 3 - Additional Features

## 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
```

## Access Control via Radius Attribute NAS-Port-id

This feature provides an additional way to control the access to serial ports other than the one based in usernames or groups. The authentication type must be Radius for this feature to function. The Radius server administrator must configure the user (in the radius server database) with one NAS-PORT-Id attribute for each serial port that the user is allowed to access.

In the example below the user alfred can access the serial ports ttyS11, ttyS13, and ttyS17:

```
alfred Auth-Type = Local, Password = 'alfred'

        Service-Type = Framed-User,

        Framed-Protocol = PPP,

        NAS-Port-Id = 11,

        NAS-Port-Id = 13,

        NAS-Port-Id = 17
```

The pam_radius module will check whether the NAS-Port-Id matches one of those sent by the radius server. If the radius server does not send the NAS-Port-Id attribute, no check is performed.

# Chapter 3 - Additional Features

No configuration is needed for the AlterPath Console Server or the Cyclades-TS. However, the authentication type must be "radius". Authentications like radiusDownLocal, radius/local, etc. will not validate the NAS-port-Id if the user was locally authenticated.

## NIS Client

NIS (Network Information System) provides simple and generic client-server database access facilities that can be used to distribute information. This makes the network appear as a single system, with the same accounts on all hosts. The objective of this feature is to allow the administrator to manage ACS accounts on a NIS server.

The NIS client feature needs these following files/commands:

| | |
|---|---|
| /etc/yp.conf | This file contains the configuration used by ypbind. |
| /etc/domainname.conf | This file contains the NIS domain name (set by the command domainname). |
| /usr/sbin/ypbind | Finds the server for NIS domains and maintains the NIS binding information. |
| /usr/bin/ypwhich | Returns the name of the NIS server that supplies the NIS services. |
| /usr/bin/ypcat | Prints the values of all keys from the NIS database specified by map name. |
| /usr/bin/ypmatch | Prints the values of one or more keys from the NIS database specified by map name. |
| /usr/sbin/domainname | Shell script to read/write the NIS domain name. |

## NIS Client Configuration

**Step 1: Run the command _domainname_.**
You ll want to make sure that you have the NIS domain name set.

```
Command :    domainname [NIS domain name]

show or set the system's NIS/YP domain name

Ex : #domainname cyclades-nis
```

**Step 2: Edit the /etc/yp.conf file.**

You will need to configure the NIS server.

```
Command :    vi /etc/yp.conf

Example :  NIS server has IP address 192.168.160.110, to

add the following line in the file

ypserver 192.168.160.110
```

**Step 3: Edit the /etc/nsswitch.conf file.**

Change the /etc/nsswitch.conf file ("System Databases and Name service Switch "configuration file) to include the NIS  in the lookup order of the  databases.

**Step 4: Configure the parameter "<all/sxx>.authype" as "local."**

## How to Test the Configuration

To test the configuration do the following:

**Step 1: Start up the following command:**

```
/usr/sbin/ypbind
```

**Step 2: Display the NIS server name.**

Display the name of NIS server by running the following command:

```
/usr/bin/ypwhich
```

**Step 3: Display the "all users" entry.**

Displays the all users' entry in the NIS database by running the following command:

```
/usr/bin/ypcat -t passwd.byname
```

**Step 4: Display the user's entry in the NIS passwd file.**

/usr/bin/ypmatch -t <userid/username> passwd.byname

If the preceding steps were performed successfully, you now need to change the /etc/inittab file by uncommenting the line that performs a ypbind upon startup.

# Chapter 3 - Additional Features

## nsswitch.conf file format

The /etc/nsswitch.conf file has the following format:

```
<database> : <service> [ <actions> <service> ]
```

where:
<database> - available: aliases, ethers, group, hosts, netgroup, network, passwd, protocols, publickey, rpc, services and shadow
<service> - available: nis (use NIS version 2) , dns (use Domain Name Service) and files (use the local files)
<actions> - Has this format: [ <status> = <action> ]

> where:
>
> > <status> = SUCCESS, NOTFOUND, UNAVAIL or TRYAGAIN
> > <action> = return or continue
>
> SUCCESS  - No error occurred and the desired entry is returned. The default action for this status is 'return'
> NOTFOUND - The lookup process works fine, but the needed value was not found. The default action for this status is "continue."
> UNAVAIL - The service is permanently unavailable.
> TRYAGAIN - The service is temporarily unavailable.

To use NIS only to authenticate users, you need to change the lines in */etc/nsswitch.conf* that reference passwd, shadow, and group.

Examples

1. You wish to authenticate the user first in the local database. If the user is not found, then use NIS:
   passwd: files nis
   shadow: files nis
   group: files nis

2. You wish to authenticate the user first using NIS. If the user is not found, then use the local database:
   passwd: nis file
   shadow: nis files
   group: nis files

3. You wish to authenticate the user first using NIS. If the user was not found or the NIS server is down, then use the local database:
   passwd: nis [UNAVAIL=continue TRYAGAIN=continue] files
   shadow: nis [UNAVAIL=continue TRYAGAIN=continue] files
   group: nis [UNAVAIL=continue TRYAGAIN=continue] files

# CAS Port Pool

This feature is available for the ACS 2.1.3 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

# Chapter 3 - Additional Features

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 ACS 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.

## 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

## Centralized Management

The AlterPath Console Server allows centralized management through the use of a Master pslave.conf file. Administrators should consider this approach to configure multiple AlterPath Console Server. 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 the following figure Figure 21: Example of Centralized Management, is /etc/portslave/TScommon.conf. It must be downloaded to each AlterPath Console Server.

> **Note:** Centralized management can mean one big configuration file (the common file) that is placed in a management host. This same file would be downloaded into all TS/ACS boxes (each of those boxes would include a tiny config file and that big common file). In this application, there may or may not be clustering involved. The user may want to access each box individually, without passing through a central point (master), using the common file just to make his/her life easier in regard to maintain the config file. This user could ALSO add the clustering application on a daily basis. Clustering does NOT require a common config file. A common config file does NOT apply to clustering, however, common config files can be used in an integrated manner.
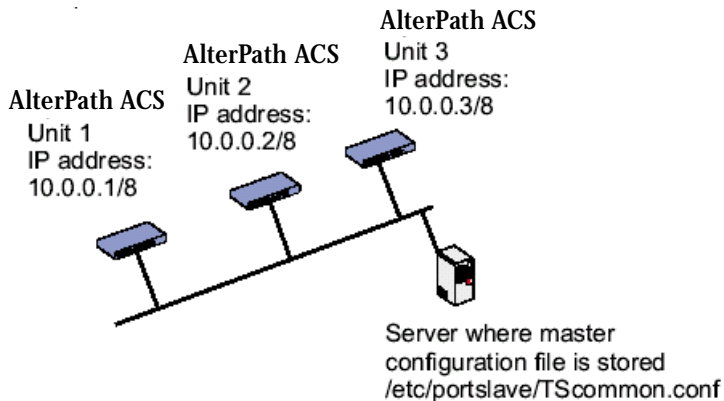


*Figure 21:  Example of Centralized Management*

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/TScommon.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:

```
all.authtype     none
all.protocol     socket_server
```

# Chapter 3 - Additional Features

```
conf.host_config unit1

all.socket_port   7001+

s1.tty      ttyS1

s2.tty      ttyS2

...

s16.tty     ttyS16

s17.tty     20.20.20.3:7033

s18.tty     20.20.20.3:7034

...

conf.host_config unit2

all.socket_port   7033+

s1.tty      ttyS1

s2.tty      ttyS2

...

sN.tty      ttySN

conf.host_config unit3

all.socket_port   7301+

s1.tty      ttyS1

s2.tty      ttyS2

...

sN.tty      ttySN

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 AlterPath Console Server.

**Step 2:** 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 3:** Execute the command signal_ras hup on each unit.

**Step 4:** Test each unit.

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

**Step 5:** Save the file and close it.

**Step 6:** 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 ACS> + N(+) is done same way as serial port.

---

# Chapter 3 - Additional Features

## Clustering

**Clustering is available for the AlterPath Console Server with firmware versions 2.1.0 and up. It allows the stringing of Terminal Servers so that one Master AlterPath Console Server can be used to access all AlterPath Console Servers on a LAN. The Master AlterPath Console Server can manage up to 1024 serial ports, so that the following can be clustered:**

- **1 Master ACS48 + 10 Slave ACS48s + 1 Slave ACS32, or**

- **1 Master ACS48 + 3 ACS8s + 1 ACS4 + 4 ACS1s, or**

- **1 Master ACS16 + 31 Slave ACS16s, or**

- **1 Master ACS32 + 15 Slave ACS32s**

**An example with one Master ACS32 and two Slave ACS16s is shown in the following figure.**
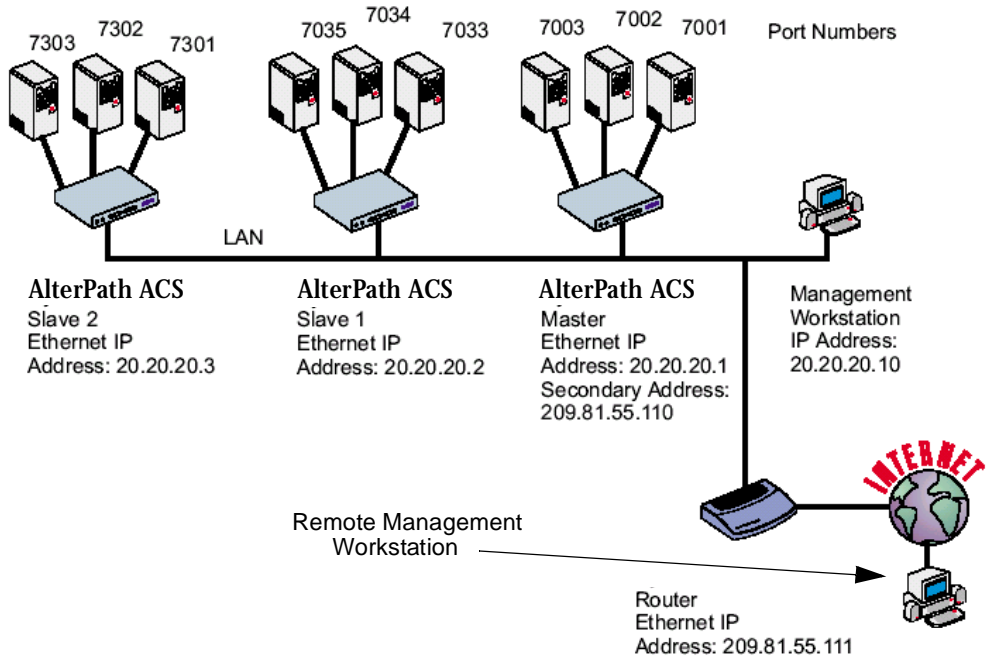


*Figure 22: An example using the Clustering feature*

## Parameters Involved and Passed Values

The Master AlterPath Console Server 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 Cyclades 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). | 209.81.55.110 |
| 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 AlterPath Console Server. | 7001+ |
| all.protocol | Depends on the application. | Socket_ssh or socket_server |
| all.authtype | Depends on the application. | Radius, local, none, remote, TacacsPlus, Ldap, kerberos, local/Radius, radius/local, local/TacacsPlus, TacacsPlus, local, RadiusDownLocal, ldapDownLocal, NIS |

# Chapter 3 - Additional Features

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

| Parameter | Description | Value for this example |
|---|---|---|
| s33.tty | This parameter must be created in the Master ACS 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. (This is an optional parameter) | Server_on_slave1_serial_s1 |
| s33.ipno | This parameter must be created in the Master ACS 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 |

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

| Parameter | Description | Value for this example |
|-----------|-------------|------------------------|
| 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 |
| 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 AlterPath Console Servers do not need to know they are being accessed through the Master AlterPath Console Server. (You are creating virtual terminals: virtual serial ports.) Their port numbers, however, must agree with those assigned by the Master.

Table 8: AlterPath Console Server 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+ |

# Chapter 3 - Additional Features

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

| Parameter | Value for this example |
|---|---|
| all.protocol | socket_server |
| all.authtype | none |
| conf.eth_ip | 20.20.20.3 |
| 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 AlterPath Console Server.

```
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.

> ⚠ **Important!**  There is one restriction in the Enhanced Clustering "session": all slaves boxes must have the same host key.

## Enhanced Clustering

With Enhanced Clustering, the CAS ports in the slave box can be configured as ssh or telnet and can have any type of authentication available. Authentication is performed in the Slave and not in the Master anymore. Additionally, the Master no longer needs to be the default gateway for all Slave boxes.

Enhanced clustering is available on implementations running Linux 2.4.x versions or newer. This new implementation is based on "iptables/nat" which is only available in these higher versions of Linux.

Enhanced Clustering has improved performance and security. Performance is greatly increased because only the NAT translation is performed on the Master box. The Master doesn't open an intermediary TCP connection with the Slave box. Also if ssh encryption and decryption is desired, it is performed on the Slave.

### New Parameters and Commands

A new parameter, conf.nat_clustering_ip allows you to enable or disable the clustering via the NAT table. This parameter should be configured with the IP address used to access the serial ports. The NAT clustering will work regardless of the interface where this IP address is assigned to. Additionally, there are two chains (post_nat_cluster and pre_nat_cluster) that holds all rules to perform NAT for clustering.

### Abbreviation List

| | |
|---|---|
| *clustering_ip* | IP address of any ACS interface (Master box). It is a public IP address and is the one that must be used to connect with the Slave's serial ports. |
| *master_ip* | Primary or secondary ethernet IP address of the Master box (usually a public IP address). |
| *slave_ip* | Primary or secondary ethernet IP address of the Slave box (usually a non public IP address) |
| *master_port* | Remote serial port parameter "socket_port" (configured in the Master box). |
| *slave_port* | Local serial port parameter "socket_port" (configured in the Slave box). |

# Chapter 3 - Additional Features

The Master box will issue a series of iptables commands to populate the nat table with the necessary rules to perform NAT translation for remote ports. Two chains will be created:

- *post_nat_cluster* (to change the source IP address), and

- *pre_nat_cluster* (to change the destination IP address)

The administrator must enable clustering via NAT in pslave.conf (conf.nat_clustering_ip <clustering_ip>).

```
iptables -D PREROUTING -t nat -p tcp -j pre_nat_cluster
iptables -D POSTROUTING -t nat -p tcp -j post_nat_cluster
iptables -t nat -F post_nat_cluster
iptables -t nat -F pre_nat_cluster
iptables -t nat -X pre_nat_cluster
iptables -t nat -X post_nat_cluster
iptables -t nat -N pre_nat_cluster
iptables -t nat -N post_nat_cluster
iptables -A PREROUTING -t nat -p tcp -j pre_nat_cluster
iptables -A POSTROUTING -t nat -p tcp -j post_nat_cluster
iptables -A pre_nat_cluster -t nat -p tcp -d <master_ip> --dport
<master_port> -j DNAT --to <slave_ip>:<slave_port>
.....
iptables -A post_nat_cluster -t nat -p tcp -d <slave_ip> --dport
<slave_port> -j SNAT --to <master_ip>

.....
```

At any time the administrator can issue an iptables command to view, change (at his own risk), or delete the rules in the nat table. If the administrator issues a "fwset restore" command he must also execute the command "signal_ras hup" to recover the nat table.

 clustering was primarily designed to allow a large number of serial ports (in more than one box) to be accessed using just one single public IP address. It only works for ports configured with the CAS profile. With iptables you can extend the access to the clustering.

Examples:

1. Accessing a Slave box with the WebUI from anywhere:

```
iptables -A PREROUTING -t nat -p tcp -d 192.168.47.79 --dport 8081
-j DNAT --to 192.168.51.2:80
```

**2.  Accessing a public DNS from any Slave box:**

```
iptables -A PREROUTING -t nat -p udp -d 64.186.161.2 --dport 53 -j
SNAT --to 64.186.161.79:53
```

How it works

The Master box () will perform two translation for each packet. The destination IP address is translated in the PREROUTING stage. The source IP address is translated in the POSTROUT-ING stage.

The command to start a telnet client session has not changed. As before, it looks like this:

```
telnet <clustering_ip> <master_port>
```

And it will have the same result as the command below issued from a local workstation:

```
telnet <slave_ip> <slave_port>
```

The command to start an ssh client session must have the following command line option:

```
-p <master_port>
```

The <master_port> will define at least the Slave box with which a connection is desired.

For example, you may use the following commands:

```
ssh -l <username1>:<server1> -p 7101 <master_ip>
ssh -l <username2>:<server2> -p 7101 <master_ip>
```

The above commands will respectively have the same result as the following commands issued from a local workstation:

```
ssh -l <username1>:<server1> <slave1_ip>
ssh -l <username2>:<server2> <slave1_ip>
```

If the parameter <master_port> defines the local IP address assigned to the serial port, the command can be simplified:

# Chapter 3 - Additional Features

```
ssh -l <username1> -p 7101 <master_ip>
ssh -l <username2> -p 7102 <master_ip>
```

**And it will have respectively the same result as the commands below issued from a local workstation:**

```
ssh -l <username1> <slave1_port1_ip>
ssh -l <username2> <slave2_port1_ip>
```

> **Note:** In the old clustering implementation <username?> and <server?> must be valid in the Master box. In the new clustering they must be valid in the Slave. In the Master box there is no meaning anymore for remote port's serverfarm and authtype parameters.
>
> If you wish to access all clustering ports with the ssh command option *-p port*, you must assign an IP address to the serial port. Do not omit the parameter socket_port in the Master box,

## General Configuration

The configuration of clustering ports is pretty much the same as before. There is only one new parameter in the Master box (conf.nat_clustering_ip) that enables or disables the clustering via NAT. The parameters *usernames* (if authentication is local) and *serverfarm* for remote ports must be configured now in the related Slave box.

In the following configuration examples, looking like "s[1-32].tty ttyS[1-32]" must be seen as 32 lines. For example:

```
s1.tty ttyS1
s2.tty ttyS2
...
s32.tty ttyS32
```

## Master box Configuration

```
#
# Enable Clustering via NAT
#
conf.nat_clustering_ip 64.186.161.108

#

#  Primary ethernet IP address (must be the public IP).

#

conf.eth_ip   64.186.161.108

conf.eth_mask   255.255.255.0

conf.eth_mtu   1500

#

# Secondary ethernet IP address

#

conf.eth_ip_alias      192.168.170.1

conf.eth_mask_alias 255.255.255.0

#

# Local CAS serial ports (32 socket_ssh ports)

#

all.protocol socket_ssh

all.authtype local

all.socket_port 7001+


s[1-32].tty ttyS[1-32]

#
```

# Chapter 3 - Additional Features

```
# Remote CAS serial ports, slave-1 (32 socket_ssh ports). This kind
of configuration can be used for ssh only; just one entry is neces-
sary.

#

s33.tty  192.168.170.2

s33.socket_port  7000

#

# Remote CAS serial ports, slave-2 (32 socket_server ports)

#

s65.tty 192.168.170.3:7101
s66.tty 192.168.170.3:7102
....
s96.tty 192.168.170.3:7132

s65.socket_port 8001
s66.socket_port 8002
...
s96.socket_port 8032

#

# Remote CAS serial ports, slave-3 (32 socket_ssh ports)

#

s[97-128].tty  192.168.170.[101-132]
```

Slave-1 box Configuration

```
#
# Primary ethernet IP address
#
conf.eth_ip 192.168.170.2
conf.eth_mask 255.255.255.0
conf.eth_mtu 1500
```

```
#
# Local CAS serial ports (32 socket_ssh ports)
#
all.protocol socket_ssh
all.authtype local

s[1-32].tty ttyS[1-32]
s[1-32].serverfarm slave-1-port[1-32]
```

## Slave-2 box Configuration

```
#
# Primary ethernet IP address
#
conf.eth_ip 192.168.170.3
conf.eth_mask 255.255.255.0
conf.eth_mtu 1500

#
# Local CAS serial ports (32 socket_server ports)
#
all.protocol socket_server
all.authtype local
all.socket_port 7101+

s[1-32].tty ttyS[1-32]
```

## Slave-3 box Configuration

```
#
# Primary ethernet IP address
#
conf.eth_ip 192.168.170.4
conf.eth_mask 255.255.255.0
conf.eth_mtu 1500
```

# Chapter 3 - Additional Features

```
#
# Local CAS serial ports (32 socket_ssh ports)
#
all.protocol socket_ssh
all.authtype local
all.ipno 192.168.170.101+

s[1-32].tty ttyS[1-32]
```

### Example of starting CAS session commands

The *serverfarm*, *socket_port*, or *tty* must be provided to select which serial port is to be connected to in the Slave box 1.

```
ssh -l <username>:<slave-1-port[1-32] -p 7000 64.186.161.108
```

The master_port (socket_port in the Master) will select which serial port is to be connected to in the Slave boxes 1 and 2.

```
telnet 64.186.161.108 80[01-32]
```

```
ssh -l -p [7097-7128] 64.186.161.108
```

## CronD

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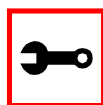
**Note:** In /etc/crontab, you can only have one active entry per user. For instance, from the example above, you cannot add another active entry for root because it already has an entry. If you want to add more scripts, you can just add them to the source file (/etc/crontab_tst).

## 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:**

**AlterPath Console Server**

# Chapter 3 - Additional Features

**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 AlterPath Console Server.**

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.
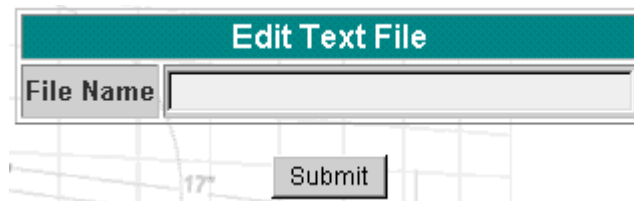


*Figure 23:  Edit Text File page*

## 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 ACS. 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 AlterPath Console Server; 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. |

# Chapter 3 - Additional Features

| | |
|---|---|
| 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*

# Chapter 3 - Additional Features

**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.



*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] :

# Chapter 3 - Additional Features

*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

# Chapter 3 - Additional Features

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 :
```

# Chapter 3 - Additional Features

**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>
```

# Chapter 3 - Additional Features
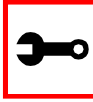
> **Tip. You can configure all the parameters for a serial port in one line:**
>
> ```
> config configure line <serial port number> tty <string>
> conf nfsdb <string> db <number> dbmode <string> dbmenu
> <number> dbtimestamp <number> syslogdb <number>
> ```

**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 AlterPath Console Server 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 AlterPath Console Server 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 AlterPath Console Server or the default gateway are changed, the AlterPath Console Server 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. |

# Chapter 3 - Additional Features

/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.

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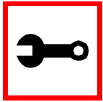
**Note.** Do not modify the -c /bin/handle_dhcp option.

---

## 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

## Dual Power Management

The AlterPath Console Server comes with two power supplies which it can self-monitor. If either of them fails, two actions are performed: sounding a buzzer and generating a syslog message. This automanagement can be disabled (no actions are taken) or enabled (default), any time by issuing the commands:

```
signal_ras buzzer off

signal_ras buzzer on
```

To disable the buzzer in boot time, edit the shell script */bin/ex_wdt_led.sh* and remove the keyword "buzzer." The buzzer won't sound if there is a power failure in any power supply. This parameter does not affect the behavior of the command "signal_ras buzzer on/off." To make this change effective even after future reboots, create a line with "/bin/ex_wdt_led.sh" in /etc/config_files, save and quit that file and run saveconf.
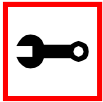
**Note:** This section applies only to the dual power supply model of the AlterPath Console Server.

### Parameters Involved and Passed Values

There are no parameters to be configured. However, if you want to generate alarms in case of a power failure, the s*yslog-ng.conf* file must be changed. See the section Generating Alarms.

### Configuration for CAS

vi Method
**Files to be changed:**

```
/etc/syslog-ng/syslog-ng.conf
```

Browser Method
**Follow the steps described in the section "Generating Alarms."**

## Configuration for TS

vi Method
**Same as for CAS.**

## Configuration for Dial-in Access

vi Method
**Same as for CAS.**

# Chapter 3 - Additional Features

## Filters and Network Address Translation

The Filter feature is available for firmware version 2.1.0 and above; the Network Address Translation (NAT) feature is available for firmware version 2.1.1 and above.

### Description

IP filtering consists of blocking or not the passage of IP packets, based on rules which describe the characteristics of the packet, such as the contents of the IP header, the input/output interface, or the protocol. This feature is used mainly in firewall applications, which filter the packets which could crack the network system or generate unnecessary traffic in the network.

Network Address Translation (NAT) allows the IP packets to be translated from local network to global network, and vice-versa. This feature is particularly useful when there is demand for more IP addresses in the local network than available as global IP addresses. In the ACS, this feature will be used mainly for clustering (one "Master" Console server works as the interface between the global network and the "slave" Console servers).

The ACS uses the Linux utility *iptables* to set up, maintain and inspect both the filter and the NAT tables of IP packet rules in the Linux kernel. Besides filtering or translating packets, the iptables utility is able to count the packets which match a rule, and to create logs for specific rules.

### Structure of the iptables

The iptables are structured in three levels: table, chain, and rule. A table can contain several chains, and each chain can contain several rules.

#### Table

The table indicates how the iptables will work. There are currently three independent tables supported by the iptables, but only two will be used:

| | |
|---|---|
| *filter* | This is the default table. |
| *nat* | This table is consulted when a packet that creates a new connection is encountered. |

## Chain

Each table contains a number of built-in chains and may also contain user-defined chains. The built-in chains will be called according to the type of packet. User-defined chains will be called when a rule which is matched by the packet points to the chain. Each table has a particular set of built-in chains:

for the *filter table*:

| | |
|---|---|
| *INPUT* | For packets coming into the box itself. |
| *FORWARD* | For packets being routed through the box. |
| *OUTPUT* | For locally-generated packets. |

for the *nat* table:

| | |
|---|---|
| *PREROUTING* | For altering packets as soon as they come in. |
| *OUTPUT* | For altering locally-generated packets as soon as they come in. |
| *POSTROUTING* | For altering packets as they are about to go out. |

## Rule

Each chain has a sequence of rules. These rules contain:

| | |
|---|---|
| *How the packet should appear in order to match the rule.* | Some information about the packet is checked according to the rule, for example, the IP header, the input and output interfaces, the TCP flags and the protocol. |
| *What to do when the packet matches the rule.* | The packet can be accepted, blocked, logged or jumped to a user-defined chain. For the nat table, the packet can also have its source IP address and source port altered (for the POSTROUTING chain) or have the destination IP address and destination port altered (for the PREROUTING and OUTPUT chain). |

AlterPath Console Server

# Chapter 3 - Additional Features

When a chain is analyzed, the rules of this chain are reviewed one-by-one until the packet matches one rule. If no rule is found, the default action for that chain will be taken.

## Syntax

An iptables tutorial is beyond the scope of this manual. For more information on iptables, see the iptables man page (not included with the ACS) or the how-to: http://www.netfilter.org or http://www.iptables.org

The syntax of the iptables command is:

```
iptables –command chain rule-specification [-t table] [options]

iptables -E old-chain-name new-chain-name
```

where:

| | |
|---|---|
| *table* | Can be filter or nat. If the option -t is not specified, the filter table will be assumed. |
| *chain* | Is one of the following: |

- for filter table: INPUT, OUTPUT, FORWARD or a user-created chain.

- for nat table: PREROUTING, OUTPUT, POSTROUTING or a user-created chain.

## 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 iptables can differentiate it from all other options.

| | |
|---|---|
| *-A*<br>- - *append* | Append one or more rules to the end of the selected chain. When the source and/or destination names resolve to more than one address, a rule will be added for each possible address combination. |
| *-D*<br>- - *delete* | Delete one or more rules from the selected chain. There are two versions of this command. The rule can be specified as a number in the chain (starting at 1 for the first rule) or as a rule to match. |

| | |
|---|---|
| *-R*<br>*- - replace* | Replace a rule in the selected chain. If the source and/or destination names resolve to multiple addresses, the command will fail. Rules are numbered starting at 1. |
| *-I*<br>*- - insert* | Insert one or more rules in the selected chain as the given rule number. Thus if the rule number is 1, the rule or rules are inserted at the head of the chain. This is also the default if no rule number is specified. |
| *-L*<br>*- - list* | List all rules in the selected chain. If no chain is selected, all chains are listed. It is legal to specify the -Z (zero) option as well, in which case the chain(s) will be atomically listed and zeroed. The exact output is affected by the other arguments given. |
| *-F*<br>*- - flush* | Flush the selected chain. This is equivalent to deleting all the rules one-by-one. |
| *-Z*<br>*- - zero* | Zero the packet and byte counters in all chains. It is legal to specify the  -L, --list (list) option as well, to see the counters immediately before they are cleared. (See above.) |
| *-N*<br>*- - new-chain* | New chain. Create a new user-defined chain by the given name. There must be no target of that name already. |
| *-X*<br>*- - delete-chain* | Delete the specified user-defined chain. There must be no references to the chain. If there are, you must delete or replace the referring rules before the chain can be deleted. If no argument is given, it will attempt to delete every non-built-in chain in the table. |
| *-P*<br>*- - policy* | Set the policy for the chain to the given target. Only non-user-defined chains can have policies, and neither built-in nor user-defined chains can be policy targets. |
| *-E*<br>*- - rename-chain* | Rename the user-specified chain to the user-supplied name. This is cosmetic, and has no effect on the structure of the table. |
| *-h*<br>*- - help* | Help. Gives a (currently very brief) description of the command syntax. |

# Chapter 3 - Additional Features

Rule Specification

-*p*        - -protocol[!]protocol
The protocol of the rule or of the packet to check. The specified protocol can be one of tcp, udp, icmp, or all, or it can be a numeric value, representing one of these protocols or a different one. A protocol name from */etc/protocols* is also allowed. A "!" argument before the protocol inverts the test. The number zero is equivalent to *all*. Protocol *all* will match with all protocols and is taken as default when this option is omitted.

-*s*        - -source[!]address[/mask]
Source specification. Address can be either a hostname, a network name, or a plain IP address. The mask can be either a network mask or a plain number, specifying the number of 1's at the left side of the network mask. Thus, a mask of 24 is equivalent to 255.255.255.0. A "!" argument before the address specification inverts the sense of the address. The flag - -src is a convenient alias for this option.

-*d*        - -destination[!]address[/mask]
Destination specification. See the description of the -s (source) flag for a detailed description of the syntax. The flag - -dst is an alias for this option.

-*j*        - - jump target
This specifies the target of the rule; i.e., what to do if the packet matches it. The target can be a user-defined chain (other than the one this rule is in), one of the special built-in targets which decide the fate of the packet immediately, or an extension (see EXTENSIONS below). If this option is omitted in a rule, then matching the rule will have no effect on the packet's fate, but the counters on the rule will be incremented. The special built-in targets are :

- ACCEPT means to let the packet through.

- DROP means to drop the packet on the floor.

- QUEUE means to pass the packet to userspace (if supported by the kernel).

- RETURN means stop traversing this chain and resume at the next rule in the previous (calling) chain. If the end of a built-in chain is reached or a rule in a built-in chain with target RETURN is matched, the target specified by the chain policy determines the fate of the packet.

## Options

The following additional options can be specified:

| | |
|---|---|
| *-i* | - -in-interface[!][name]<br>Optional name of an interface via which a packet is received (for packets entering the INPUT and FORWARD chains). When the "!" argument is used before the interface name, the sense is inverted. If the interface name ends in a "+" then any interface which begins with this name will match. If this option is omitted, the string "+" is assumed, which will match with any interface name. |
| *-o* | - -out-interface[!][name]<br>Optional name of an interface via which a packet is going to be sent (for packets entering the FORWARD and OUTPUT chains). When the "!" argument is used before the interface name, the sense is inverted. If the interface name ends in a "+" then any interface which begins with this name will match. If this option is omitted, the string "+" is assumed, which will match with any interface name. |
| [!] | -f - -fragment<br>This means that the rule only refers to second and further fragments of fragmented packets. Since there is no way to tell the source or destination ports of such a packet (or ICMP type), such a packet will not match any rules which specify them. When the "!" argument precedes the "-f" flag, the rule will only match head fragments, or unfragmented packets. |
| *-c* | - -set-counters PKTS BYTES<br>This enables the administrater to initialize the packet and byte counters of a rule (during INSERT, APPEND, REPLACE operations). |
| *-v* | - -verbose<br>Verbose output. This option makes the list command show the interface address, the rule options (if any), and the TOS masks. The packet and byte counters are also listed, with the suffix 'K', 'M' or 'G' for 1000, 1,000,000 and 1,000,000,000 multipliers respectively (but see the -x flag to change this). For appending, insertion, deletion and replacement, this causes detailed information on the rule or rules to be printed. |

# Chapter 3 - Additional Features

***-n***            - -numeric
Numeric output. IP addresses and port numbers will be printed in numeric
format. By default, the program will try to display them as host names, net-
work names, or services (whenever applicable).

***-x***            - -exact
Expand numbers. Display the exact value of the packet and byte counters,
instead of only the rounded number in K's (multiples of 1000) M's (multiples
of 1000K) or G's (multiples of 1000M). This option is only relevant for the -L
command.

- -line-        When listing rules, add line numbers to the beginning of each rule, corre-
numbers      sponding to that rule's position in the chain.

## Match Extensions

Iptables can use extended packet matching modules. These are loaded in two ways:
implicitly, when -p or - -protocol is specified, or with the -m or - -match option, followed by
the matching module name; after these, various extra command line options become
available, depending on the specific module.

## TCP Extension

These extensions are loaded if the protocol specified is tcp or "-m tcp" is specified. It
provides the following options:

| | |
|---|---|
| - -source-port [!] [port[:port]] | Source port or port range specification. This can either be a service name or a port number. Inclusive range can also be specified, using the format port:port. If the first port is omitted, "0" is assumed; if the last is omitted, "65535" is assumed. If the second port is greater then the first they will be swapped. The flag - -sport is an alias for this option. |
| - -destination-port [!] [port[:port]] | Destination port or port range specification. The flag - -dport is an alias for this option. |

| | |
|---|---|
| - -tcp-flags [!] mask comp | Match when the TCP flags are as specified. The first argument is the flags which we should examine, written as a comma-separated list, and the second argument is a comma-separated list of flags which must be set. Flags are: SYN ACK FIN RST URG PSH ALL NONE. Hence the command iptables<br>-A FORWARD -p tcp - -tcp-flags SYN,ACK,FIN,RST SYN will only match packets with the SYN flag set, and the ACK, FIN and RST flags unset. |
| [!] - -syn | Only match TCP packets with the SYN bit set and the ACK and FIN bits cleared. Such packets are used to request TCP connection initiation; for example, blocking such packets coming in an interface will prevent incoming TCP connections, but outgoing TCP connections will be unaffected. It is equivalent to<br>- -tcp-flags SYN,RST,ACK SYN.<br>If the "!" flag precedes the "- -syn," the sense of the option is inverted. |
| - -tcp-option [!] number | Match if TCP option set. |

## UDP Extension

These extensions are loaded if the protocol udp is specified or "-m udp" is specified. It provides the following options:

| | |
|---|---|
| - -source-port [!] [port[:port]] | Source port or port range specification. See the description of the - -source-port option of the TCP extension for details. |
| - -destination-port [!] [port[:port]] | Destination port or port range specification. See the description of the - -destination-port option of the TCP extension for details. |

# Chapter 3 - Additional Features

## ICMP Extension

This extension is loaded if the protocol icmp is specified or "-m icmp" is specified. It provides the following option:

| | |
|---|---|
| *- -icmp-type [!] typename* | This allows specification of the ICMP type, which can be a numeric ICMP type, or one of the ICMP type names shown by the command *iptables -p icmp -h* |

## Multiport Extension

This module matches a set of source or destination ports. Up to 15 ports can be specified. It can only be used in conjunction with -m tcp or -m udp.

| | |
|---|---|
| - -source-port [port[,port]] | Match if the source port is one of the given ports. |
| - -destination-port [port[,port]] | Match if the destination port is one of the given ports. |
| - -port [port[,port]] | Match if the both the source and destination port are equal to each other and to one of the given ports. |

## Target Extensions

Iptables can use extended target modules. The following are included in the standard distribution.

## LOG

Turn on kernel logging of matching packets. When this option is set for a rule, the Linux kernel will print some information on all matching packets (like most IP header fields) via the kernel log (where it can be read with syslog-ng).

| | |
|---|---|
| - -log-level level | Level of logging (numeric or see syslog.conf(5)). |
| - -log-prefix prefix | Prefix log messages with the specified prefix; up to 29 letters long, and useful for distinguishing messages in the logs. |
| - -log-tcp-sequence | Log TCP sequence numbers. This is a security risk if the log is readable by users. |

- -log-tcp-options          Log options from the TCP packet header.

- -log-ip-options           Log options from the IP packet header.

## REJECT (filter table only)

This is used to send back an error packet in response to the matched packet: otherwise it is equivalent to DROP. This target is only valid in the INPUT, FORWARD and OUTPUT chains, and user-defined chains which are only called from those chains. Several options control the nature of the error packet returned:

- -reject-with type     The type given can be icmp-net-unreachable, icmp-host-unreachable, icmp-port-unreachable, icmp-proto-unreachable, icmp-net-prohibited or icmp-host-prohibited, which return the appropriate ICMP error message (port-unreachable is the default). The option echo-reply is also allowed; it can only be used for rules which specify an ICMP ping packet, and generates a ping reply. Finally, the option tcp-reset can be used on rules which only match the TCP protocol: this causes a TCP RST packet to be sent back. This is mainly useful for blocking ident probes which frequently occur when sending mail to broken mail hosts (which won't accept your mail otherwise).

## SNAT (nat table only)

This target is only valid in the nat table, in the POSTROUTING chain. It specifies that the source address of the packet should be modified (and all future packets in this connection will also be mangled), and rules should cease being examined. It takes one option:

# Chapter 3 - Additional Features

- -to-source <ipaddr>[-<ipaddr>][:port-port]

This can specify a single new source IP address, an inclusive range of IP addresses, and optionally, a port range (which is only valid if the rule also specifies -p tcp or -p udp). If no port range is specified, then source ports below 1024 will be mapped to other ports below 1024: those between 1024 and 1023 inclusive will be mapped to ports below 1024, and other ports will be mapped to 1024 or above. Where possible, no port alteration will occur.

## DNAT (nat table only)

This target is only valid in the nat table, in the PREROUTING and OUTPUT chains, and user-defined chains which are only called from those chains. It specifies that the destination address of the packet should be modified (and all future packets in this connection will also be mangled), and rules should cease being examined. It takes one option:

- -to-destination <ipaddr>[-<ipaddr>][:port-port]

This can specify a single new destination IP address, an inclusive range of IP addresses, and optionally, a port range (which is only valid if the rule also specifies -p tcp or -p udp). If no port range is specified, then the destination port will never be modified.

## MASQUERADE (nat table only)

This target is only valid in the nat table, in the POSTROUTING chain. It should only be used with dynamically assigned IP (dialup) connections: if you have a static IP address, you should use the SNAT target. Masquerading is equivalent to specifying a mapping to the IP address of the interface the packet is going out on, but also has the effect that connections are forgotten when the interface goes down. This is the correct behavior when the next dialup is unlikely to have the same interface address (and hence any established connections are lost anyway). It takes one option:

- -to-ports <port>[-<port>]  This specifies a range of source ports to use, overriding the default SNAT source port-selection heuristics (see above). This is only valid with if the rule also specifies -p tcp or -p udp).

## REDIRECT (nat table only)

This target is only valid in the nat table, in the PREROUTING and OUTPUT chains, and user-defined chains which are only called from those chains. It alters the destination IP address to send the packet to the machine itself (locally-generated packets are mapped to the 127.0.0.1 address). It takes one option:

- -to-ports <port>[-<port>]  This specifies a destination port or range or ports to use: without this, the destination port is never altered. This is only valid with if the rule also specifies -p tcp or -p udp).

## Parameters Involved and Passed Values

The file with the iptables rules is */etc/network/firewall*. The *fwset* script saves the iptables rules in the file */etc/network/firewall* (command *iptales-save > /etc/network/firewall*) and then save the file in the flash memory. The *fwset restore* restores the iptables rules previously saved in */etc/network/firewall* file (command *iptables-restore </etc/network/firewall*). This command is executed at boot to invoke the last configuration saved.

## Configuration for CAS, TS, and Dial-in Access

vi method

**Step 1:  Execute fwset restore.**
This script will restore the IP Tables chains and rules configured in the /etc/network/ firewall file. This script can be called in the process, whenever the user wants to restore the original configuration.

**Step 2:  Add the chains and rules using the command line.**
See details of the iptables syntax earlier in this chapter.

# Chapter 3 - Additional Features

**Step 3:** **Execute iptables-save > /etc/network/firewall.**

This program will save all the rules and chains of all the tables in the /etc/network/ firewall file.

**Step 4:** **Execute updatefiles /etc/network/firewall.**

This program will save the configuration to the flash memory.

Browser method

**Step 1:** **Point the browser to the Console Server.**
In the Address or Location field of your browser type the IP Address or the alias of your console server.

**Step 2:** **Log in.**

Log in as root, and type the password configured for the root user. This will take you to the Configuration and Administration page.

**Step 3:** **Select the IPTables link.**

On the Configuration section of this page, select the IPTables link. The following page will appear.
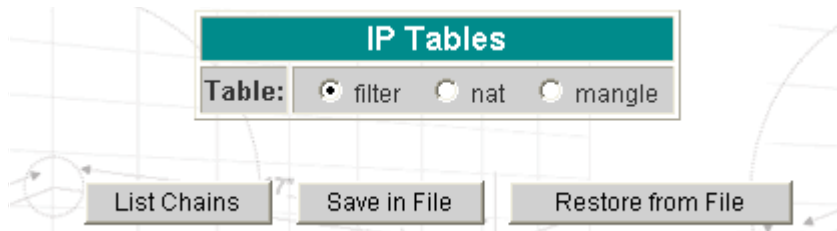


*Figure 27: First IP Tables page*

The options in this page are:

| | |
|---|---|
| *List Chains* | List all the chains of the table selected. |
| *Save in File* | Save the all the IP tables rules, chains and tables to the file /etc/network/firewall. |
| *Restore from File* | Reads the file /etc/network/firewall and make the IP Tables configuration from that file effective. |

**Step 4:** Select the filter table. Click the List Table button.

A table with all the chains of the table, and the number of bytes/packets which used each chain will appear. The available options are:

| | |
|---|---|
| *List Rules* | This option shows all the rules related to the chain selected. |
| *Edit Chain* | Will be valid only for built-in chains. The user will be able to edit the default policy. |
| *Delete Chain* | Only user-defined chains can be deleted. |
| *Insert Chain* | If the chain name is entered, this option will cause the program to try to create a chain with this name. |
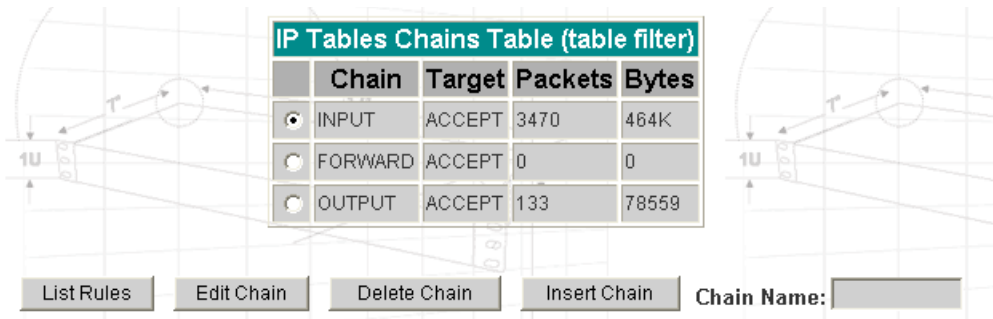


*Figure 28: IP Tables Chains Table (table filter)*

**Step 5:** Edit the chain list

If the user needs to define new chains, write in the Chain Name text input and click the Insert Chain button. If the default policy for a chain needs to be changed, select the chain and click the Edit Chain button. Select the new policy and click Submit.

**Step 6:** Choose one of the chains and click the List Rules button.

A table with all the rules related to the chain selected will appear in the page, containing the rule configuration and the accounting (number of bytes and packets which used the rule). In the beginning, there are no rules in the chain: in this case, the only option is to Append Rule.

# Chapter 3 - Additional Features

When there are rules in the chain, the page will appear like the picture below. The options are:
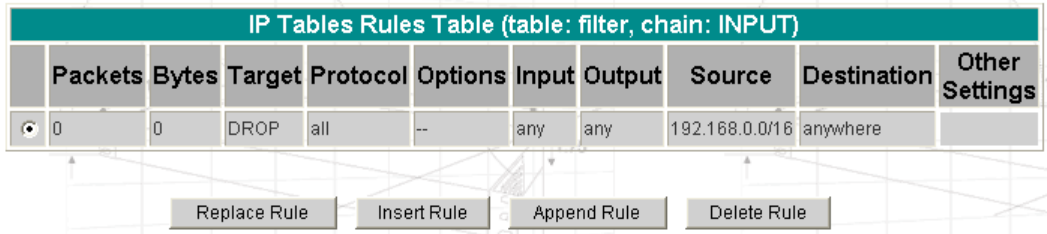
*Replace Rule*          Replace the rule selected with another rule, defined by the user.

*Insert Rule*           Insert a new rule in the position selected. After the rule is defined, all the rules from that position will move down.

*Append Rule*           Insert a rule in the bottom of the list.

*Delete Rule*           Remove the rule selected from the list.

[IP Tables] [IP Tables Chains Table]

| | Packets | Bytes | Target | Protocol | Options | Input | Output | Source | Destination | Other Settings |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | | IP Tables Rules Table (table: filter, chain: INPUT) | | | | | | |
| ⊙ | 0 | 0 | DROP | all | -- | any | any | 192.168.0.0/16 | anywhere | |

Replace Rule    Insert Rule    Append Rule    Delete Rule

*Figure 29:  IP Tables Rules Table (table: filter, chain: INPUT)*

Step 7:  Click the button Append Rule to start.

The page which follows is for configuring the rule. There are several parameters related to a rule:

Note:  For many parameters, there is a checkbox called *inverted*. Checking this box will invert the sense of the parameter.
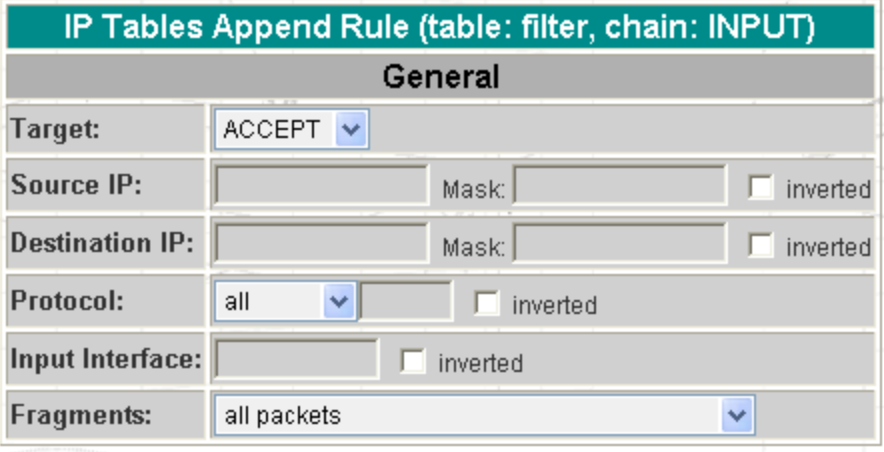
| | |
|---|---|
| *Target* | Indicates the action to be performed when the IP packet matches the rule. The kernel can ACCEPT the packet, DROP it, LOG it, REJECT it by sending a message, translating the source or the destination IP address/port (in the nat table) or send the packet to another user-defined chain. All the options are in the target list. |
| *Source/Destination IP* | Indicates how the source/destination IP address should be. When a network should be included in the rule, the network mask must be configured too. |
| *Input/Output interface* | Indicates the interface where the IP packet should pass. The Input Interface option will appear only for the chains INPUT, FORWARD and PREROUTING; the Output interface option will appear for the chains FORWARD, OUTPUT and POSTROUTING. |
| *Protocol* | Indicates the transport protocol to check. If the numeric value is available, select numeric and type the value in the text input; otherwise, select one of the other options. |
| *Fragments* | Indicates if the fragments will be checked. The IP Tables can either check for head fragments and unfragmented packets or for the subsequent fragments. |
| *TCP options* | This section will appear only when TCP protocol is selected. The source/destination ports can be configured in this section, as well as the TCP flags. |
| *UDP options* | This section will appear only when UDP protocol is selected. The source/destination ports can be configured in this section. |
| *ICMP options* | This section will appear only when ICMP protocol is selected. The ICMP type can be configured in this section. |
| *LOG options* | This section will appear only when the target selected is LOG. It contains parameters to set the way the logs will appear (syslog level, prefix, flags). |
| *REJECT options* | This section will appear only when the target selected is REJECT. It indicates what message the filter will send when the IP packet is rejected. |

# Chapter 3 - Additional Features



*Figure 30:  IP Tables Append Rule (table: filter, chain: INPUT)*

**Step 8:  Configure the rule and click the Submit button.**

If there is an error in the configuration, a red message will appear with the page; otherwise, the rule will be included in the chain rules list.

**Step 9:  Repeat steps 7 and 8 to add as many rules as necessary.**

**Step 10:  Click on the link [IP Tables Chains Table] if there are rules to be added in other chains.**

Repeat steps 6 to 8 to add rules for other chains.

**Step 11:** Click on the link [IP Tables] if the nat table must be edited.

Select the nat table and click on the List Chains button. Repeat steps 5 to 8 to edit the chains and rules in the nat table. The tables presented on the Web page are the same as in the filter table, with the difference that there are more options in the Append/Insert/Replace Rule page:

| | |
|---|---|
| *DNAT/SNAT options* | This section will appear only when the target selected is DNAT and SNAT, respectively. The parameters of these sections will determine how the packets matched by the rule will be translated. DNAT translates the destination IP Address/Port, and SNAT translates the source IP Address/Port. |
| *MASQUERADE/REDIRECT options* | This section will appear only when the target selected is MASQUERADE or REDIRECT. The parameter of these sections configure the port or the port range used to masquerade the source or to redirect the destination, respectively. |

**Step 12:** Click on the link [IP Tables] and click on the Save to File button.

This will cause the rules and chains to be saved in the /etc/network/firewall file.

**Step 13:** Click on the link Administration > Load/Save Configuration and click the Save to Flash button.

This will save the rules and chains in the flash memory.

# Chapter 3 - Additional Features

## 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.

# Chapter 3 - Additional Features

## 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).**

# Chapter 3 - Additional Features

*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] :
```

# Chapter 3 - Additional Features

*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
```

# Chapter 3 - Additional Features

```
# 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  ACS 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 -v 1 -c public 10.0.0.1 \"\" \"\" 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 ACS 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.**

# Chapter 3 - Additional Features

```
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
```

# Chapter 3 - Additional Features

## 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. |

# Chapter 3 - Additional Features

*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 mes-
sage:

```
sendsms -d 123 -m "Hi. This is a test message send from ACS 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:<br>"-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 mal-formed. 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 -c  public 10.0.0.1 "" 2 0 "" .1.3.6.1.2.1.2.2.1.2.1
s " ACS: 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 (''). |

# Chapter 3 - Additional Features

## 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.**

# Chapter 3 - Additional Features

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 |

# Chapter 3 - Additional Features

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.)**

# Chapter 3 - Additional Features

**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"
```
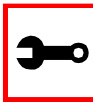
**Tip.** You can specify the range or list of serial ports if you wish to configure the same parameters for several ports. For instance, to configure parameters for ports 2 through 4, you can type this command: config configure line 2-4 [options]. Or to configure parameters for just ports 4, 6, and 9, 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> ?
```

# NTP

The ntpclient is a *Network Time Protocol* (RFC-1305) client for UNIX- and Linux-based computers. In order for the AlterPath Console Server 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 updated from the NTP 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). |

# Chapter 3 - Additional Features

## 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 Figure 16: Configuration and Administration page.) You can then pull up the appropriate file and edit it.

# PCMCIA

> **Note:** This section applies only to the model of the ACS that has a dual power supply.

## Supported Cards

The following cards are supported by the ACS:

• 16-bit PC Cards

• PCMCIA Compact Flash Cards

• The Sierra Aircard 750 GSM

• MerlinG201 GSM PCMCIA card

The 32-bit CardBus PC Cards are not supported. For an updated list of supported cards, please check the Cyclades Web site.

## Tools for Configuring and Monitoring PCMCIA Devices

During the ACS boot, the */etc/init.d/pcmcia* script loads the PCMCIA core drivers and the *cardmgr* daemon. The *cardmgr* daemon is responsible for monitoring PCMCIA sockets, loading client drivers when needed, and running user-level scripts in response to card insertions and removals.

*lsmod*          This command shows the modules loaded for the PCMCIA devices.

# Chapter 3 - Additional Features

*cardctl*     This command can be used to check the status of a socket, or to see how it is configured. Just type *cardctl* to see the syntax of the command. *cardctl config* displays the card configuration. *cardctl ident* can be used to get card identification information. *cardctl eject* stops the application and unloads the client driver, and *cardctl insert* re-loads the driver and re-starts the application.

> **Note:** *cardctl suspend*, *cardctl resume* and *cardctl reset* are not supported.

## Ejecting Cards

You can insert the card anytime, and the drivers should be loaded automatically. But you will need to run *cardctl eject* before ejecting the card to stop the application using the card. Otherwise the ACS may hang during the card removal. You must specify the slot number when using the *cardctl* command. For example:

```
cardctl eject 0 for the lower slot
```

and

```
cardclt eject 1 for the upper slot
```

> **Note:** Due to a known problem in the current release, the I/O ports used by the card cannot be re-used after card re-insertion. In each card insertion, the card gets a different I/O port. This limits the number of times the card can be ejected and inserted. When all the I/O ports known by the card are used, the *RequestIO: No more items* message is displayed, and the only way to reset the I/O port usage is to reboot the system.

## PCMCIA Network Configuration

The onboard Ethernet device has the *eth0* name. The first PCMCIA Ethernet card or wireless LAN card detected will receive the *eth1* name, the second card will be *eth2*.

*cardmgr* will read the network settings from the */etc/network/interface*s and assign an IP to *eth1*.

> **Note:** Before changing the */etc/network/interfaces* file, unload the network client driver using *cardctl eject*.

The factory default */etc/network/interfaces* has the following lines:

```
# auto eth1

# iface eth1 inet static

#     address 192.168.0.42

#     network 192.168.0.0

#     netmask 255.255.255.0

#     broadcast 192.168.0.255

#     gateway 192.168.0.1
```

Remove the # in the beginning of the line, and change the IPs to suit your network configuration. For instance, you may want the following configuration:

```
 auto eth1

 iface eth1 inet static

     address 192.168.162.10

     network 192.168.162.0

     netmask 255.255.255.0
```

# Chapter 3 - Additional Features

```
broadcast 192.168.162.255

gateway 192.168.162.1
```

Don't forget to run *saveconf* to save this configuration in the flash, so that it can be restored in the next boot. Run *cardctl insert* to load the network drivers with the new configuration.
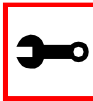
> 🔧 **Note:** Do not use *ifconfig* to change the network settings for the PCMCIA device. Otherwise, you may be unable to unload the network driver during *cardctl* eject and the ACS may hang. The correct way is to change the */etc/network/interfaces* file.

## Wireless LAN PC Cards

First do the appropriate PCMCIA network configuration. Additionally, the configuration of the wireless driver is done in the following file:

```
/etc/pcmcia/wireless.opts
```

For instance, to configure the network name as *MyPrivateNet*, and the WEP encryption key as *secu1*, the following settings could be added to the default "`*,*,*,*)`" entry:

```
*,*,*,*)

    INFO="This is a test"

    ESSID="MyPrivateNet"

    KEY="s:secu1"
```
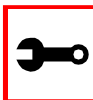
> 🔧 **Note:** The "s:" prefix in the KEY line indicates that the key is an ASCII string, as opposed to hex digits. Five characters or ten digits could be entered for WEP 40-bit and 13 characters or 26 digits could be entered for WEP 128-bit.

There is a generic sample in the end of the *wireless.opts* file that explains all possible settings. For more details in wireless configuration, search for *manpage iwconfig* on the Internet. The parameters in *wireless.opts* are used by the *iwconfig* utility. After changing any of the param-

eters, run *cardctl eject* followed by *cardctl inser*t to load the new settings. Also, run *saveconf* to save the new settings to flash. *iwconfig eth1*shows the basic wireless parameters set in *eth1*. *iwlist* allows to list frequencies, bit-rates, encryption, etc. The usage is:

```
iwlist eth1 frequency

iwlist eth1 channel

iwlist eth1 ap

iwlist eth1 accesspoints

iwlist eth1 bitrate

iwlist eth1 rate

iwlist eth1 encryption

iwlist eth1 key

iwlist eth1 power

iwlist eth1 txpower

iwlist eth1 retry
```

## Modem PC Cards

The modem device gets the */dev/ttySn* name, where *n* is the number of embedded serial devices plus 1. For instance, if the ACS has 32 onboard serial devices, the modem card becomes the */dev/ttyS33*.

When a modem card is detected, *cardmgr* starts a script which loads mgetty for the modem device automatically. mgetty provides the login screen to the remote user. mgetty may also be configured to start PPP (pppd) and let PPP login the caller. The steps to allow PPP connections are:

Step 1:  **Enable login and PAP authentication in /etc/mgetty/login.config.**
Enable the desired authentication in /etc/mgetty/login.config. For instance, you may want the following authentication in /etc/mgetty/login.config to enable PAP and system password database authentication:

# Chapter 3 - Additional Features

```
/AutoPPP/ - a_ppp /usr/local/sbin/pppd auth -chap +pap login
nobsdcomp nodeflate
```

**Step 2: Create a user name in /etc/ppp/pap-secrets.**

If +pap authentication was selected, create a user name in
/etc/ppp/pap-secrets. For instance, you may add the following line:

```
"mary" * "marypassword" *
```

**Step 3: Create 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 4: Copy the /etc/ppp/options.ttyXX as /etc/ppp/options.ttyS33 (the modem port).**

Copy the /etc/ppp/options.ttyXX to have the device name assigned to the pcmcia
modem. For instance, if the modem is the ttyS33, /etc/ppp/options.ttyXX should be
copied as /etc/ppp/options.ttyS33. If you are not sure which ttySxx is the modem
device, do a "ls -al /dev/modem" with the modem inserted.

**Step 5: Uncomment local and remote IPs in /etc/ppp/options.ttyS33.**

Uncomment the line that assigns the local and remote IPs in
/etc/ppp/options.ttyS33 (or whatever is the tty name in your system). For
instance, you may want to assign 192.168.0.1 for local ip, and 192.168.0.2
for the remote ip.

**Step 6: Save /etc/ppp/options.ttyS33 in flash.**

**Step 7: Create an entry in /etc/config_files.**

It should have the name of the file you created, so that the new file can be saved to
the flash. For instance, you will have to add a line with /etc/ppp/options.ttyS33 in
/etc/config_files.

**Step 8: Run saveconf to save the files listed in /etc/config_files to the flash.**

**Step 9: Insert the pcmcia modem if not inserted yet.**

**Step 10: Run ps to see that mgetty is running.**

The ACS is ready to receive dial in calls.

**Step 11: Establish PPP connection with the ACS.**

From the remote system, use pppd to dial and establish a PPP connection with the ACS. The remote system should have the login user name set in their /etc/ppp/pap-secrets to have a successful login in the ACS.

## Establishing a Callback with your Modem PC Card

Setting up a callback system serves two purposes:

1. Cost savings: reversing line charges - allows your company to call you back.

2. Security: makes sure users are who they pretend to be by calling a well-known or precon-figured number back.

The steps to allow callback are divided into two parts. Part One is the configuration for the Advanced Secure Console Port ServerACS (Server Side ACS Setup). Part Two is the configuration for the client side.

### Server Side ACS Setup

**Step 1: Enable authentication.**

Enable the desired authentication in /etc/mgetty/login.config. For instance, you may want the following authentication in /etc/mgetty/login.config to enable PAP and system password database authentication:
/AutoPPP/ - a_ppp /usr/local/sbin/pppd auth -chap +pap login nobsdcomp nodeflate

**Step 2: Configure a pseudo callback user.**

Add the following line to /etc/mgetty/login.config with the appropriate values. Do this before the line /*    -    -    /bin/login @/ at the end of the file.

```
<pseudo callback name>    -    -    /sbin/callback -S <phone
number of the client>/
```

**ie:**

```
call    -    -    /sbin/callback -S 12345
```

# Chapter 3 - Additional Features

'call' is the pseudo callback name. '123456' is the number to dial back.

---
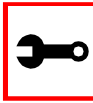
Note:

1.The order of configuration in /etc/mgetty/login.config matters. By default, it has the line *     -     -     /bin/login @ at the end of the file. This line allows any users to log in and be verified by the *login* program.  If you were to add the callback line *after* this line, the callback program will *not* be initiated when you try logging in. Instead, the *login* program will be used to verify you since it was encountered first. List the callback users first if you want the option of having some users access the callback program and the others the *login* program.

```
call    -    -    /sbin/callback -S 12345

call2   -    -    /sbin/callback -S 77777

*       -    -    /bin/login @
```

The example above will allow you to have the option whether or not you want to use the callback functionality. If you log in with *call* or *call2*, then callback starts immediately. If you log in as anybody else other than *call* or *call2*, callback will not start and you'll be verified by the *login* program.

2.Don't use * instead of some callback user name. Mgetty will fall to infinite callback.

3. If you don't specify a telephone number, callback will ask for a number after you log in as the pseudo callback user.

---

Step 3:  (If you plan to login through PPP with PAP authentication) create pap user name in /etc/ppp/pap-secrets.

Add a line similar to the following: (include the quotes and the two asterisks).

```
"myUserName"    *    "myUserNamePassword"*
```

Step 4:  (If you plan to login through PPP) follow steps 4 - 9 in the section above on Modem PC Cards.

**Step 5: Create users.**

**Step A: Create a new user with the command** *adduser myUserName.*
This will create an entry in /etc/passwd that resembles this:

```
myUserName:$1$/3Qc1pGe$./h3hzkaJQJ/:503:503:Embedix
User,,,:/home/myUserName:/bin/sh
```

**Step B: If you want to limit myUserName to getting ONLY PPP access and NOT shell access to the server, edit the entry for myUserName in /etc/passwd..**

Do this by replacing /bin/sh with a pathname to a script that you will be creating later. In the following example, the script is: */usr/ppp/ppplogin*

myUserName:$1$/3Qc1pGe$./h3hzkaJQJ/:503:503:Embedix
User,,,:/home/myUserName:*/usr/ppp/ppplogin*

**Step 6: If you executed Step 5b, create the ppp login script.**

**Step A: Create a script called /etc/ppp/ppplogin following this format:**

```
#!/bin/sh
exec /usr/local/sbin/pppd <ppp options>
```

**Step B: Make script executable.**

Type *chmod 755 /etc/ppp/ppplogin*.

**Step C: Save this file to flash.**
Save this file to flash so the next time the ACS gets rebooted, you won't lose the new file. Add /etc/ppp/ppplogin into /etc/config_files. Now execute *saveconf.*

**Step 7: Change permission of pppd.**

Type *chmod u+s /usr/local/sbin/pppd*

# Chapter 3 - Additional Features

---

> **Tip**. To prevent from always having to manually change permission every time your ACS reboots:
> 1. Edit /etc/users_scripts by uncommenting the following line:
>
> ```
> /bin/chmod_pppd
> ```
>
> 2. Add /etc/users_scripts into /etc/config_files.
> 3. Execute *saveconf*. The next time the ACS reboots, this change will be in effect. You should not need to manually change the pppd permission.

**Step 8:  Your ACS is ready to establish a callback connection.**

See Client Side Setup to start the callback connection.


Client Side Setup

**Step 1:  Activate Show Terminal Window option.**

(From Win2000) Go to your Connection window (the window to dial the ACS ) -> Properties -> Security -> look for Interactive Logon and Scripting -> click on Show Terminal Window.

**Step 2:  Disable/enable encryption protocols.**

If you are going to be using PPP connection with PAP authentication, make sure you disable all other encryption protocols.
(from Win2000) go to your Connection window (the window to dial the ACS)
-> Properties -> Security -> click on Advanced (custom settings) ->
click on Settings -> click on Allow these protocols -> disable all protocols except the PAP one.

**Step 3:  Set up modem init string.**

It is *very* important that before callback hangs the call, the modem in the Windows box does not tell Windows that the call has been dropped. Otherwise, Windows Dial-up Networking will abort everything (because it thinks the call was dropped with no reason).

(From Win2000) Go to Windows' control panel -> Phone and Modem  -> Modems -> choose your modem -> Properties -> Advanced -> add &c0s0=1 to Extra Settings.

Step 4:  Call your ACS.

> Step A:  Dial to the ACS modem using either the normal username or the ppp user-
> name that you created in Step 5 when configuring the server side.
>
> Step B:  Once a connection is made, you get a login prompt.
>
> Step C:  Login with the pseudo callback name to start the callback.
>
> Step D:  Your connection gets dropped. The ACS is now calling you back.
>
> Step E:  After reconnection to you, you get a login prompt again.
>
> Step F:  Now you can:
>
> • Log in through character mode: Log in with username and password. You will get
> the ACS shell prompt.
>
> • Log in through ppp: Click on Done on the Terminal Window.

## GSM Card Configuration

This works for Firmware 2.1.3 and up.

In /etc/mgetty/mgetty.config, add this entry:

```
port ttyS2

  data-only y

  init-chat "" \d\d\d+++\d\d\dATZ OK
```

where ttyS2 may have to be changed to the serial port that will be assigned to the GSM card,
eg. replace ttyS2 by ttyS9 for an ACS8.

Add '/etc/mgetty/mgetty.config' to /etc/config_files and call saveconf:

```
echo /etc/mgetty/mgetty.config >> /etc/config_files

saveconf
```

If the SIM card needs a PIN, edit /etc/pcmcia/serial.opts. Uncomment the line

# Chapter 3 - Additional Features

```
INITCHAT="- \d\d\d+++\d\d\datz OK at+cpin=1111 OK"
```

and replace '1111' by the PIN.

Insert the card. The card should flash red first. After the PIN is sent, the LED stays red, until the card found network. It then flashes green.

## ISDN PC Cards

You can establish synchronous PPP connections with ISDN cards. The ipppd is the daemon that handles the synchronous PPP connections.

### How to configure dial in

**Step 1: Create a user.**

Create a user in /etc/ppp/pap-secrets or in /etc/ppp/chap-secrets, depending if you want PAP or CHAP authentication. You will also have to create a user in /etc/ppp/pap-secrets if you want radius or local authentication. In case you don't want to repeat all the user database from the radius server an option is to use '*' as the user in /etc/ppp/pap-secrets:

```
*       *       " "       *
```

**Step 2: Change the options in /etc/pcmcia/isdn.opts to fit your environment.**

Make sure that $DIALIN is set to "yes." Set the desired authentication in DIALIN_AUTHENTICATION. For instance,
"+pap" for PAP, "+chap" for CHAP, "login auth" or "login +pap" for radius,
"login auth" or "login +pap" for local. When "login auth" or "login +pap"
are used, PAM libraries are used so /etc/pam.conf should be also configured.

**Step 3: Run saveconf to save your changes to the flash.**

**Step 4: If the ISDN card is not inserted, it is time to insert the card.**

ipppd is started automatically. Go to step 6.

**Step 5: Restart script.**

If the card was already inserted, you will need to restart the isdn script to re-load any changed configuration. To restart the script, issue:

```
/etc/pcmcia/isdn stop ipppp0
```

```
/etc/pcmcia/isdn start ipppp0
```

**Step 6:  You can dial from the remote system to the ACS, and get a PPP connection.**

**Step 7:  To hang up the connection from the ACS side, just issue:**

```
isdnctrl hangup ipppp0
```

How to configure dial out

**Step 1:  Create a user.**
Create a user in /etc/ppp/pap-secrets or in /etc/ppp/chap-secrets,depending if you want PAP or CHAP authentication.

**Step 2:  Change options.**
Change the options in /etc/pcmcia/isdn.opts to fit your environment. Make sure that $DIALIN is set to "no". Set $USERNAME to the user name provided by your ISP.

**Step 3:  Run saveconf to save your changes to the flash.**

**Step 4:  If the ISDN card is not inserted, it is time to insert the card.**
ipppd is started automatically. Go to step 6.

**Step 5:  Restart script.**
If the card was already inserted, you will need to restart the isdn script to re-load any changed configuration. To restart the script, issue:

```
/etc/pcmcia/isdn stop ipppp0
```

```
/etc/pcmcia/isdn start ipppp0
```

**Step 6:  To dial out, issue the command:**

```
isdnctrl dial ipppp0
```

**Step 7:  To hangup the connection from the ACS side, just issue:**

```
isdnctrl hangup ipppp0
```

# Chapter 3 - Additional Features

## Establishing a Callback with your ISDN PC Card

For the same cost saving reasons explained in <u>**Establishing a Callback with your Modem PC Card**</u>, the ISDN card in the ACS can be configured to callback client machines after receiving dial in calls.

The steps to allow callback are divided into two parts. Part One is the configuration for the ACS (ACS Setup) as callback server. Part Two is the configuration of a Windows 2000 Professional computer as callback client.

### ACS setup (Callback Server)

**Step 1:** Change the parameters in /etc/pcmcia/isdn.opts to fit your environment.

**Step 2:** Set the callback number in DIALOUT_REMOTENUMBER:

```
DIALOUT_REMOTENUMBER="8358662" # Remote phone that you want to
                               # dial to
```

**Step 3:** If your isdn line supports caller id, it is recommended that you also configure the DIALIN_REMOTENUMBER and enable secure calls. Otherwise skip to step 4.

```
DIALIN_REMOTENUMBER="8358662" # Remote phone from which you will
                              # receive calls

SECURE="on"      # "on" = incoming calls accepted only if remote
                 # phone matches DIALIN_REMOTENUMBER; "off" =
                 # accepts calls from any phone. "on" will work
                 # only if your line has the caller id info.
```

**Step 4:** Make sure the CALLBACK is set to "in" in /etc/pcmcia/isdn.opts.
       CALLBACK="in"   # "in" will enable callback for incoming calls.

**Step 5:** Uncomment line with user "mary" in /etc/ppp/pap-secrets.

**Step 6: Save changes to flash.**

```
saveconf
```

**Step 7: Activate the changes by stopping and starting the isdn script:**

```
/etc/pcmcia/isdn stop ippp0

/etc/pcmcia/isdn start ippp0
```

The ACS side is done.

Windows 2000 Professional configuration (Callback Client)

**Step 1: Create user "mary" with password "marypasswd" in Control Panel -> "User and Passwords".**

**Step 2: Create a dial-up connection that uses "Modem - AVM ISDN Internet (PPP over ISDN) (AVMISDN1)".**

(To create a dial-up connection, go to Start->Settings->Network and Dial-up Connections->Make New Connection, select "I want to set up my Internet connection manually, or I want to connect through a local area network", select "I connect through a phone line and a modem", select the "AVM ISDN Internet (PPP over ISDN)" modem, type the phone number you dial to connect to the ACS, and enter mary as User name and marypasswd as password.). After creating this dial-up, click on the Properties of this dial-up, select the "Options" panel, and change the Redial attempts to 0.

**Step 3: Accept incoming connections.**

To accept incoming connections, go to Start->Settings->Network and Dial-up Connections->Make New Connection, select "Accept incoming connections" (the words are slightly different in XP), select AVM ISDN Internet (PPP over ISDN), select "Do not allow virtual private connections", click the user "mary", then click on Properties of TCP/IP to specify the IP addresses for the calling computers. Also in "mary" Properties, select the Callback tab and make sure the option "Do not allow callback" is selected. After any change in the Incoming Connection Properties, it is recommended that the Windows is rebooted to apply the changes.

The Windows side is done.

# Chapter 3 - Additional Features

Now you can dial from Windows to the ACS. Go to Start-> Settings-> "Network and Dial-up Connections" and select the dial-up that you created. After the "Dialing" message, you will see a window with a warning message:

```
Opening port....
Error 676: The phone line is busy.
```

Just click Cancel. In a few seconds, the ACS will call you back, and you will see the connection icon in the taskbar.

## Establishing a Callback with your ISDN PC Card (2nd way)

The previous section explained how to do callback at D-Channel level. The advantages of having callback at D-Channel level is that it works independent of the Operating System on the client side. But a big disadvantage is that the callback call happens before the authentication phase in PPP. The only security is  by that only calls from predefined phone numbers are accepted.

To fix that drawback, this section explains another way to have callback with the ACS. The steps described here will work when the remote side is a UNIX machine, not Windows. The callback call will happen after the PPP authentication is successful.

### ACS Setup (Callback Server)

**Step 1:  Change the parameters in /etc/pcmcia/isdn.opts to fit your environment.**
**Step 1.1: Set the callback number in DIALOUT_REMOTENUMBER.**

```
DIALOUT_REMOTENUMBER="8358662" # Remote phone that you want to

                               # dial to
```

**Step 1.2: Configure the DIALIN_REMOTENUMBER.**
     If your ISDN line supports caller id, it is recommended that you also configure the DIALIN_REMOTENUMBER and enable secure calls. Otherwise skip to Step 1.3.

```
DIALIN_REMOTENUMBER="8358662" # Remote phone from which you will

                              # receive calls

SECURE="on"        # "on" = incoming calls accepted only if remote
```

```
                        # phone matches DIALIN_REMOTENUMBER; "off" =

                        # accepts calls from any phone. "on" will work

                        # only if your line has the caller id info.
```

**Step 1.3: Set the desired IPs for local and remote machines.**
**Step 1.4: Set DIALIN to "yes".**

```
DIALIN="yes" # "yes" if you want dial in, "no" if you want dial out
```

**Step 1.5: Make sure the CALLBACK parameter is disabled.**

```
CALLBACK="off"    # "off" = callback disabled.
```

**Step 1.6: Add the user that will callback the client in DIALIN_AUTHENTICATION.**

```
DIALIN_AUTHENTICATION="auth login user mary"
```

**Step 2:  Make sure /etc/pam.conf has the configuration you want (e.g., radius).**
**This step is only required if you are using "auth login" in DIALIN_AUTHENTICATION. When using "auth login," /etc/pam.conf is what defines which authentication will be used.**

**Step 3:  Add the user "mary" in /etc/ppp/pap-secrets.**

**Step 4:  Uncomment lines in /etc/ppp/auth-up.**

**Step 5:  Save changes to flash:**

```
saveconf
```

**Step 6:  Activate the changes by stopping and starting the isdn script:**

```
/etc/pcmcia/isdn stop ippp0
```

```
/etc/pcmcia/isdn start ippp0
```

# Chapter 3 - Additional Features

Linux (Callback Client)

**Step 1: Configure the ipppd to have user mary and pap authentication.**

**Step 2: Dial to the ACS:**

```
isdnctrl dial ippp0
```

**Step 3: As soon the ACS authenticates the user mary, the ACS will disconnect and callback.**

## Compact Flash Cards

Introduction

Compact Flash cards are small memory cards with a capacity up to 1GB. They can be used like a normal hard disk drive using IDE. Using an adapter, CF cards can be used in PCMCIA slots. Such an adapter is very cheap, because the PCMCIA and CF card standard are the same, only the pin layout and the socket is different. On the market, there are also small PCMCIA hard drives available, eg. a drive from Toshiba with a capacity of 5GB (Toshiba MK5002MPL). CF card support can be used in the ACS for storing files. This would be especially useful for example to save the configuration. CF cards cannot be rewritten indefinetly. For this reason, CF should not be used for logging.

For data buffering, a PCMCIA hard drive is ideal:

• data will not be lost on power loss / crash / reboot of the CAS.

• no dependency on an NFS server that may fail.

Description of the Feature

When inserting an adapter with a CF card or a PCMCIA hard drive, an *ide* device appears. This can be mounted, eg. by:

```
mkdir /mnt/ide
```

```
mount /dev/hda1 /mnt/ide
```

Apart from the ext2 filesystem, the VFAT filesystem will be supported. This makes it easy to exchange data with a Windows system. To create a vfat filesystem, the utility *mkdosfs* will be ported.

In addition, a utility to create partition the CF has been added. For this, the program *sfdisk* will be used. *sfdisk* can be easily used for scripting, so it can be called from the Web interface.

To check a filesystem, the utility *fsck.ext* has been added.

## How it works

When the card is inserted, *cardmgr* loads the *ide-cs* module, which depends on *ide-mod.o*. This in turn loads *ide-probe-mod.o*, which recognizes the CF as a *disk*, and *ide-disk.o* will be loaded. From this point on, the partitions (usually one) can be mounted using *mount*. If the filesystem is vfat, the modules *fat.o* and *vfat.o* will be loaded.

## Configuration

**Step 1:  Insert the card.**

**Step 2:  Type the following at the command prompt:**

```
mount /dev/hda1 /mnt/ide
```

Using /etc/pcmcia/ide.opts, mounting can be done automatically when the card is inserted:

```
# ATA/IDE drive adapter configuration

#

# The address format is "scheme,socket,serial_no[,part]".

#

# For multi-partition devices, first return list of parti-
tions in

# $PARTS.  Then, we'll get called for each partition.

#

case "$ADDRESS" in

*,*,*,1)

    #INFO="Sample IDE setup"

    DO_FSTAB="y" ; DO_FSCK="y" ; DO_MOUNT="y"
```

```
            FSTYPE="vfat"

            #OPTS=""

            MOUNTPT="/mnt/ide"

            ;;

     *,*,*)

            PARTS="1"

            # Card eject policy options

            NO_CHECK=n

            NO_FUSER=n

            ;;

     esac
```

These parameters can be changed:

| | |
|---|---|
| DO_FSTAB | If set to 'y', an entry in /etc/fstab will be created. |
| DO_FSCK | A boolean (y/n) setting: specifies if the filesystem should be checked before being mounted, with *fsck -Ta.* |
| DO_MOUNT | If set to 'y', the card will be mounted automatically upon insertion. |
| FS_TYPE | can be either 'vfat' or 'ext2'. Determines the filesystem type. |
| MOUNTPT | the mount point where the partition will be mounted. |
| NO_CHECK, NO_FUSER | Boolean (y/n) settings for card eject policy. If NO_CHECK is true, then ``cardctl eject'' will shut down a device even if it is busy. If NO_FUSER is true, then the script will not try to kill processes using an ejected device. |

PARTS
A list of partitions to be mounted. The conf file will be called again for each partition. In the example above, there is an entry only for partition '1', but you can eg. set PARTS="1 3 4" and add entries for the case statement like:

```
*,*,*,3)
 # settings for partition 3
 ;;
*,*,*,4)
 # settings for partition 4
 ;;
```

To give different configuration for slot 0 and 1, the second parameter in the case statement can be used. For example:

```
*,0,*,1)
 # settings for slot 0
 ;;
*,1,*,1)
 # settings for slot 1
 ;;
```

## Sierra Aircard 750 GSM PCMCIA card

### Introduction

The Sierra Aircard 750 is a GSM card. It allows wireless connections over the GSM network. It works very similar to a modem card.

### Description of the Feature

This feature implements support for the Sierra Aircard 750 card. When inserted, it will be detected automatically, and the modules loaded, just like all other PCMCIA cards that are supported.

# Chapter 3 - Additional Features

### How it works

When the card is inserted, *cardmgr* automatically loads the *serial_cs* module. The script */etc/pcmcia/serial* is started, reading options from */etc/pcmcia/serial.opts*. It starts an *mgetty* process on the serial device. Because the CIS information given by the card is wrong, a CIS file is supplied with correct parameters.

### How to configure it

The configuration is the same as for a PCMCIA modem card. Additionally, if the PIN is not disabled on the SIM card (some providers do not allow this), it must be configured in */etc/pcmcia/serial.opts* with an entry like

```
# one time init chat (example: PIN number for GSM card):

# important: give '-' to expect nothing, instead of '' or ""

INITCHAT="- \d\d\d+++\d\d\datz OK at+cpin=1111 OK"
```

This will ensure that the PIN number will be given when the card is inserted.

## Merlin G201 GSM PCMCIA card

### Introduction

The Merlin G201 is a GSM card. It allows wireless connections over the GSM network. It works very similar to a modem card.

### Description of the Feature

This feature implements support for the Merlin G201 card. When inserted, it will be detected automatically, and the modules loaded, just like all other PCMCIA cards that are supported.

### How it works

When the card is inserted, *cardmgr* automatically loads the *serial_cs* module. The *script /etc/pcmcia/serial* is started, reading options from */etc/pcmcia/serial.opts*. It starts an *mgetty* process on the serial device.

How to configure it

The configuration is the same as for a PCMCIA modem card. Additionally, if the PIN is not disabled on the SIM card (some providers do not allow this), it must be configured in */etc/pcmcia/serial.opts* with an entry like

```
# one time init chat (example: PIN number for GSM card):

# important: give '-' to expect nothing, instead of '' or ""

INITCHAT="- \d\d\d+++\d\d\datz OK at+cpin=1111 OK"
```

This will ensure that the PIN number will be given when the card is inserted.

# Chapter 3 - Additional Features

## Ports Configured as Terminal Servers

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] :
```

> **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 :
```

> **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]:
```

# Chapter 3 - Additional Features

*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>
```

> **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>
> ```

**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.)

**AlterPath Console Server**

# 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 : <br> *-igncr* <br> This tells the terminal not to ignore the carriage-return on input, <br> *-onlcr* <br> Do not map newline character to a carriage return or newline character sequence on output, <br> *opost* <br> Post-process output, <br> *-icrnl* <br> 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.

# Chapter 3 - Additional Features

**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
```

# Chapter 3 - Additional Features

```
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] :
```

# Chapter 3 - Additional Features

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.

# Chapter 3 - Additional Features

*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>
```

# Chapter 3 - Additional Features

> **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> par-
> ity <string> flow <string> dcd <number> dtr_reset <num-
> ber> 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.*

# Chapter 3 - Additional Features

*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>
```

# Chapter 3 - Additional Features

**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 conf igure datasize:**

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

**To conf igure stopbits:**

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

**AlterPath Console Server**

# Chapter 3 - Additional Features

## Session Sniffing

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 AlterPath Console Server 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 AlterPath Console Server 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 AlterPath Console Server 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 ACS
```

To inform the AlterPath Console Server 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 AlterPath Console Server will show him/her a list of the pairs PID/user_name, and s/he will be able to select one session

# Chapter 3 - Additional Features

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 ACS 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, whereas no means sniffing is not permitted.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.

## 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.

# Chapter 3 - Additional Features

**Step 5:** **Scroll down to the Sniff Session section.**

You can configure the appropriate values here.



*Figure 31:  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] :
```

# Chapter 3 - Additional Features

*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] :
```

# Chapter 3 - Additional Features

*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] :
```

# Chapter 3 - Additional Features

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
```

## 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 ACS uses the net-snmp package (http://www.net-snmp.org).

> ⚠️ **Important!** Check the SNMP configuration before gathering information about ACS 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 ACS cannot permit the public community to read SNMP information.

The net-snmp supports snmp version 1, 2 and 3. To use SNMP version 1 or 2 (community), you need to configure the communities in the snmp config file (/etc/snmp/snmpd.conf). For example, to include the communities cyclades and public, you need add the following lines in

```
/etc/snmp/snmpd.conf :

        # cyclades is read-write community

        rwcommunity cyclades

        # public is a read-only community

        rocommunity public
```

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
```
For example :
```
createUser usersnmp MD5 senhadousersnmp DES
```

# Chapter 3 - Additional Features

**Step 2:** **Edit the /etc/snmp/snmpd.conf file.**

If the user has permission to read only, to add the line :

```
rouser <username> (ex.: rouser usersnmp).
```

If the user has permission to read and write, to add the line :

```
rwuser <username> (ex.: rwuser usersnmp).
```

**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 Cyclades Vendor MIB ( enterprises.2925 )

- Cyclades ACSxx Remote Management Object Tree (cyclades.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 cyclades.mib file.)

## 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.

Step 1: Map the community name *public* into a *security name*.

```
#         sec.name   source            community

#com2sec notConfigUser  default         public
```

Step 2: Map the security name into a group name.

```
#         groupName        securityModel securityName

#group    notConfigGroup v1              notConfigUser

#group    notConfigGroup v2c             notConfigUser
```

# Chapter 3 - Additional Features

**Step 3: Create a view to which the group has rights.**

```
#       name            incl/excl    subtree       mask(optional)
view    all      included      .1
```

**Step 4: Grant the group read-only access to the all view.**

```
#       group           context sec.model sec.level prefix read    write
notif
#access  notConfigGroup ""       any       noauth    exact  all none none
```

**Example:**

```
# Here is a commented out example configuration that allows less
# restrictive access.


# YOU SHOULD CHANGE THE "COMMUNITY" TOKEN BELOW TO A NEW KEYWORD
# ONLY KNOWN AT YOUR SITE.  YOU *MUST* CHANGE THE NETWORK TOKEN
# BELOW TO SOMETHING REFLECTING YOUR LOCAL NETWORK ADDRESS SPACE.


##       sec.name  source          community
#com2sec local     localhost       COMMUNITY
#com2sec mynetwork NETWORK/24      COMMUNITY


##      group.name sec.model  sec.name
#group MyRWGroup  any         local
#group MyROGroup  any         mynetwork
#
#group MyRWGroup  any         otherv3user
```

```
#...


##           incl/excl subtree                        mask
#view all    included  .1                             80


## -or just the mib2 tree-


#view mib2   included  .iso.org.dod.internet.mgmt.mib-2 fc


##               context sec.model sec.level prefix read   write notif
#access MyROGroup ""      any       noauth    0      all    none  none
#access MyRWGroup ""      any       noauth    0      all    all   all


####################################################################
```

# Chapter 3 - Additional Features

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 .

# Chapter 3 - Additional Features

## 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:



*Figure 32: 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
```

# Chapter 3 - Additional Features

**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.

# Chapter 3 - Additional Features

*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>
```

AlterPath Console Server

# Chapter 3 - Additional Features

---

> **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 AlterPath Console Server family. 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. |

# Chapter 3 - Additional Features

**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])*<br><br>*and*<br><br>*unix-dgram (filename [options])* | They open the given AF_UNIX socket, and start listening for messages.<br>Options: owner(name), group(name), perm(mask) are equal global options<br><br>*keep-alive(yes/no)* - Selects whether to keep connections opened when syslog-ng is restarted. Can be used only with unix_stream. Default: yes<br>*max-connections(n)* - Limits the number of simultaneously opened connections. Can be used only with unix_stream. Default: 10. |

| | |
|---|---|
| *c) tcp([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. |
| *and* | 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 |
| *udp([options])* | listen on all available interfaces.<br>**Options:**<br>*ip(<ip address>)* - The IP address to bind to. Default: 0.0.0.0.<br>*port(<number>)* - UDP/TCP port used to listen messages. Default: 514.<br>*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));};
```

**AlterPath Console Server**

# Chapter 3 - Additional Features

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'); };
```

# Chapter 3 - Additional Features

**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.

# Chapter 3 - Additional Features

**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 " ACS-ALARM". The message will carry the current date, the hostname of this  ACS 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 \'ACS-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 ACS and the message that was received from the source:

```
destination d_pager {
```

# Chapter 3 - Additional Features

```
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>

-c 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 ACS and the message that was received from the source:**

```
destination d_trap {

pipe("/dev/cyc_alarm"

template("snmptrap -v 1 -c public 10.0.0.1 \"\" \"\" 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);...

};
```

# Chapter 3 - Additional Features

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"); };
```

# Chapter 3 - Additional Features

```
# 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); };
```

## TCP Keepalive

The objective of this feature is to allow the TS and ACS to recognize when the socket client (ssh or telnet) goes down without closing the connection properly. Currently, if this happens in a serial port the system administrator must close the connection manually or nobody else can access that port anymore.

### How it works

The TCP engine of TS or ACS will send a tcp keepalive message (ACK) to the client. If the maximum retry number is reached without an answer from the client, the connection is closed.

## How to Configure it

**The configuration is done in the file /bin/init_proc_fs using the linux proc filesystem.**

```
#
# Enable TCP keepalive timer  in ACS (six retries with ten seconds
of interval from each other).
#
# keepalive interval when the client is answering

echo 20 > /proc/sys/net/ipv4/tcp_keepalive_time

# keepalive interval when the client is not answering (ACS only).

echo 10 > /proc/sys/net/ipv4/tcp_keepalive_intvl

# number of retries

echo  6 > /proc/sys/net/ipv4/tcp_keepalive_probes


#
# Enable TCP keepalive timer in TS (six retries with twenty seconds
of interval from each other).
#
echo 20 > /proc/sys/net/ipv4/tcp_keepalive_time

echo  6 > /proc/sys/net/ipv4/tcp_keepalive_probes
```

# Chapter 3 - Additional Features

## 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 AlterPath Console Server. \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.

# Chapter 3 - Additional Features

**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 represents 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:] :
```

# Chapter 3 - Additional Features

*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.***

# Chapter 3 - Additional Features

*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>
```

# Chapter 3 - Additional Features

> **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.)

## 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.

**AlterPath Console Server**

# Chapter 3 - Additional Features

**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.**

This page has been left intentionally blank.

# Chapter 4 - Server Management

## Windows 2003 Server Management

### Introduction

Emergency Management Services (EMS) is a new feature in the Windows 2003 Server that allows out-of-band remote management and system recovery tasks. All Emergency Management Services output is accessible using a terminal emulator connected to the server serial port. Besides the normal character mode output sent to the serial console, Windows also sends xml tags. Those tags can be captured and processed by the ACS so that the administrator can automate the actions to be taken.

As the Administrator, you can manage the Emergency Management Services in the Windows Server 2003 from the Web interface. Instead of managing the server through the Special Administration Console (SAC), which is the console when connected directly to the Windows Server through telnet or ssh session, root users can interact with the SAC from the Web (via buttons or directly typing into the terminal window).

### How it works

When users enter the page to manage their Windows Server, corresponding commands are sent to the server when the users click on the buttons. The server message or result (successful or not) is displayed. If a user is defined for a specific port (sX.users), then that port will be displayed in the Web interface only when that user logs in. Even if are a root user, you won't be able to see that port unless you are defined in all.admin_users.

The Web interface for Windows Server 2003 is done through the java applet. Each button on the java applet is created in a configuration file (winbuttonbar.conf). When users click on one of the buttons, commands are sent to the remote server. Users can either use the buttons or they can actually click on the terminal window and enter in the commands themselves. To change to other channels once they exist, users can type <Esc> <tab> <enter> in the terminal window. To change to Channel 0, type <Esc> <tab> <0> <enter>.

Windows sends xml tags in the following situations:

- During Windows installation, it sends <channel-switch> with the setup logs.

- During boot, it sends the <machine-info> information.

- When switching channels, it sends the <channel-switch> information.

# Chapter 4 - Server Management

- During system crash, it sends the <BP> to indicate BreakPoint.

The <machine-info> tag is emitted once by Windows Server during its system boot sequence. This tag is also emitted as part of the <BP> tag. The following elements are included in <machine-info> tag:

| | |
|---|---|
| <guid> | It is the GUID that uniquely identifies the server platform. Normally, this is an SMBIOS provided identification. If no such value is available, all 0's GUID string is used (see sample encoding below). |
| <name> | Is the system name. |
| <os-build-number> | Is a numeric string that identifies a successive Windows Build. |
| <os-product> | Is the name of the Windows Server 2003 product currently running on this server. It is one of the following: |

- Windows Server 2003 Datacenter Edition

- Windows Server 2003 Embedded

- Windows Server 2003 Enterprise Edition

- Windows Server 2003

| | |
|---|---|
| <os-service-pack> | Is an alphanumeric string that identifies the most up-to-date service pack installed. If none installed, the string is None. |
| <os-version> | Is the numeric identification of the Windows version currently running. |
| <processor-architecture> | Is either x86 or IA64, designating the two processor architectures currently supported by Windows Server 2003. |

A sample encoding of this tag follows:

```
<?xml>

<machine-info>

<name>NTHEAD-800I-1</name>
```

# Chapter 4 - Server Management

`<guid>00000000-0000-0000-0000-000000000000</guid>`

`<processor-architecture>x86</processor-architecture>`

`<os-version>5.2</os-version>`

`<os-build-number>3735</os-build-number>`

`<os-product>Windows Server 2003 Enterprise Edition</os-product>`

`<os-service-pack>None</os-service-pack>`

`</machine-info>`

The console environment provided by the serial port is called Special Administration Console (SAC). In the SAC command line, each time we enter the "cmd" command we create a channel. A channel is the "Command Prompt" environment, where you can enter the Command Prompt commands (dir, cd, edit, del, copy, etc). We can switch back and forth between channel(s) and SAC by pressing Esc Tab keys. We can create up to 9 channels, i.e., up to 9 Command Prompt sessions. Whenever we switch channels, the <channel-switch> tag is sent. The following elements are included in the <channel-switch> tag:

    <application-type>       Is a hexadecimal GUID signifying the application or tool that is running on the Windows Server platform and communicating via this active channel. It is to be used to discern the different interaction modes. During the Windows GUI-mode Setup phase, the following GUIDs identify the specific types of data being emitted:

                    1.   Debug Log (5ED3BAC7-A2F9-4E45-9875-B259EA3F291F)

                    2.   Error Log (773D2759-19B8-4D6E-8045-26BF38402252)

                    3.   Action Log (D37C67BA-89E7-44BA-AE5A-112C6806B0DD)

                During nominal Windows Server operations, the following GUIDs can be expected:

                    1.   SAC (63D02270-8AA4-11D5-BCCF-806D6172696F)

                    2.   CMD (63D02271-8AA4-11D5-BCCF-00B0D014A2D0)

                The above are constant GUIDs and should not be confused with those provided via the <guid> tag below.

# Chapter 4 - Server Management

<description>      Is the user-friendly name of the active channel. For the GUI-Mode Setup tool they are:

1. Debug Log (Setup tracing log)

2. Error Log (Setup errors log)

3. Action Log (Setup actions log)

For the Windows Server, they are:

1. SAC (Special Administration Console)

2. CMD (Command Prompt)

<guid>      Is a hexadecimal GUID that identifies a specific instance of a channel. During a life-span of a Windows Server (between any two system boots), there is a total of 10 channels being allocated. Of those, one can be expected a GUID for each of the following channel types:

1. GUI-Mode Setup Debug Log

2. GUI-Mode Setup Error Log

3. GUI-Mode Setup Action Log

4. SAC

The remaining GUIDs are of the CMD channel type. For example, during Windows setup, there are 3 GUIDs assigned to Setup, 1 to SAC and the remaining 6 to CMD. However, during normal Windows operations, there is 1 GUID assigned to SAC and the remaining 9 to CMD.

These GUIDs are created anew for each instance of channels, and should not be confused with the constant GUIDs provided via the <application-type> tag above.

          AlterPath Console Server

# Chapter 4 - Server Management

| | |
|---|---|
| \<name\> | Is the system name of the active channel. For the GUI-mode Setup tool, they are the file names where the data is written: |

1. Debug Log (setuplog.txt)

2. Error Log (setuperr.log)

3. Action Log (setupact.log)

For Windows Server, they are:

1. SAC (SAC)

2. CMD (Cmdnnnn), where nnnn indicates the corresponding channel number

| | |
|---|---|
| \<type\> | Is the type of data being emitted on the active channel. Currently, there are two types of data supported: |

1. Raw for the 3 GUI-Mode Setup channels

2. VT-UTF8 for the SAC and CMD channels

A sample encoding of the SAC channel tag follows:

```
<channel-switch>

<name>SAC</name>

<description>Special Administration Console</description>

<type>VT-UTF8</type>

<guid>1aee4cc0-cff3-11d6-9a3d-806e6f6e6963</guid>

<application-type>63d02270-8aa4-11d5-bccf-806d6172696f</application-type>

</channel-switch>
```

A sample encoding of the CMD channel tag follows:

```
<channel-switch>

<name>Cmd0001</name>

<description>Command Prompt</description>
```

# Chapter 4 - Server Management

```
<type>VT-UTF8</type>

<guid>970438d1-12bb-11d7-8a92-505054503030</guid>

<application-type>63d02271-8aa4-11d5-bccf-00b0d014a2d0</application-type>

</channel-switch>
```

**A sample encoding of the GUI-Mode Setup Debug Log channel tag follows:**

```
<channel-switch>

<name>setuplog.txt</name>

<description>Setup tracing log</description>

<type>Raw</type>

<guid>6f28e904-1298-11d7-b54e-806e6f6e6963</guid>

<application-type>5ed3bac7-a2f9-4e45-9875-b259ea3f291f</application-type>

</channel-switch>
```

**The <BP> tag is emitted when the Windows Server system halts such that only elements of the kernel are the most recently operating logic.**

| | |
|---|---|
| <INSTANCE CLASSNAME=> | Is the type of break point. Currently, there is only one type emitted, i.e. "Blue Screen" which indicates the system was halted prematurely. It is represented by the CLASSNAME="BLUESCREEN" value. |
| <machine-info> | Is described above. |
| <PROPERTY NAME=> | Provides additional details, such as error code of the abnormal condition that caused the break point. |

**A sample encoding of the Break Point tag follows:**

```
<?xml>

<BP>

<INSTANCE CLASSNAME="BLUESCREEN">
```

**AlterPath Console Server**

# Chapter 4 - Server Management

```
<PROPERTY NAME="STOPCODE" TYPE="string"><VALUE>"0xE2"</VALUE>

</PROPERTY>

<machine-info>

<name>NTHEAD-800I-1</name>

<guid>00000000-0000-0000-0000-000000000000</guid>

<processor-architecture>x86</processor-architecture>

<os-version>5.2</os-version>

<os-build-number>3735</os-build-number>

<os-product>Windows Server 2003 Enterprise Edition</os-product>

<os-service-pack>None</os-service-pack>

</machine-info>

</INSTANCE>

</BP>
```

## How to Configure it

**There is a new parameter in /etc/portslave/pslave.conf to monitor for xml data. For instance, for ttyS1 we could configure:**

```
s1.xml_monitor          1
```

**When the xml_monitor is set, cy_buffering will search for xml packets coming from the serial port. When a complete xml packet is received, cy_buffering will send it to syslog-ng. In syslog-ng.conf, the following filters are available to filter the xml messages:**

```
filter f_windows_bluescreen { facility(local[0+<conf.DB_facility>]) and level(info)
and

            match("XML_MONITOR") and match("BLUESCREEN"); } ;
filter f_windows_boot { facility(local[0+<conf.DB_facility>]) and level(info) and
            match("XML_MONITOR") and not match("BLUESCREEN") and
            match("machine-info"); } ;
```

# Chapter 4 - Server Management

Once the desired message is filtered, we have to define which actions we would like to take. Syslog-ng will create macros that can give easy access for the administrators to access the xml information. If the administrator uses these macros, syslog-ng replaces the macros by the data received in the xml packet. For instance, the following table shows the macros that are available when filter f_windows_bluescreen is successful and the examples of values that can replace the macros:

Table 14: Windows 2003 Macros

| Macro | Description | Value to replace macro |
|-------|-------------|------------------------|
| $<INSTANCE CLASSNAME=> | Reason for the break point. Currently there is only one type, BLUESCREEN. | BLUESCREEN |
| $<PROPERTY NAME=> | Additional details about break point. | STOPCODE |
| $<VALUE> | Additional details about break point. | 0xE2 |
| $<name> | Machine name | MY_WIN_SERVER |
| $<guid> | GUID that uniquely identifies this server. If no such value is available, all 0's GUID string is used. | 4c4c4544-8e00-4410-8045-80c04f4c4c20 |
| $<processor-architecture> | Processor architecture. It can be either x86 or IA64. | x86 |
| $<os-version> | Windows version. | 5.2 |
| $<os-build-number> | Numeric string that identifies a successive Windows Build. | 3763 |
| $<os-product> | Which Windows Server product. It can be: Windows Server 2003 Datacenter Edition, Windows Server 2003 Embedded, Windows Server 2003 Enterprise Edition or Windows Server 2003. | Windows Server 2003 |

AlterPath Console Server

# Chapter 4 - Server Management

Table 14: Windows 2003 Macros

| Macro | Description | Value to replace macro |
|---|---|---|
| $<os-service-pack> | Alphanumeric string that identifies the most up-to-date service packet installed. If none installed, the string is None. | None |
| $<tty> | ACS serial port tty or serverfarm name. | S1.ttyS1 |

For the f_windows_boot, the following macros are available:

Table 15: f_windows_boot Macros

| Macro | Description | Value to replace macro |
|---|---|---|
| $<name> | Machine name | MY_WIN_SERVER |
| $<guid> | GUID that uniquely identifies this server. If no such value is available, all 0's GUID string is used. | 4c4c4544-8e00-4410-8045-80c04f4c4c20 |
| $<processor-architecture> | Processor architecture. It can be either x86 or IA64. | x86 |
| $<os-version> | Windows version. | 5.2 |
| $<os-build-number> | Numeric string that identifies a successive Windows Build. | 3763 |
| $<os-product> | Which Windows Server product. It can be: Windows Server 2003 Datacenter Edition, Windows Server 2003 Embedded, Windows Server 2003 Enterprise Edition or Windows Server 2003. | Windows Server 2003 |

Table 15: f_windows_boot Macros

| Macro | Description | Value to replace macro |
|-------|-------------|------------------------|
| $<os-service-pack> | Alphanumeric string that identifies the most up-to-date service packet installed. If none installed, the string is None. | None |
| $<tty | ACS serial port tty or serverfarm name. | S2.server_connected_ to_serial2 |

As an example on how we can use above macros, let's say we want the ACS to send an e-mail to the administrator whenever a crash happens. The e-mail should have the information about the reason of the crash, machine name and windows version information. So we just have to create the following entry in syslog-ng.conf:

```
destination win2003mail { pipe("/dev/cyc_alarm"

        template("sendmail -t administrator@cyclades.com -f acs -s \"\

        Server $<name> crashed\" -m \'\

        Break Point: $<INSTANCE CLASSNAME=>  $<PROPERTY NAME=> $<VALUE>\

        Server: $<name>\

        OS: $<os-product>\

        Build: $<os-build-number>  Version: $<os-version>\

        Service Pack: $<os-service-pack>\

        Processor: $<processor-architecture>\

        Server GUID: $<guid>\

        ACS port: $<tty>\

         \' -h mail.cyclades.com "));};
```

And the following entry will activate the win2003mail action when the f_windows_bluescreen filter is successful:

# Chapter 4 - Server Management

```
source src { unix-stream("/dev/log"); };

log { source(src); filter(f_windows_bluescreen); destination(win2003mail); };
```

## Server Commands

**The following are the different commands and their descriptions that can be sent to the server.**

Table 16: Server Commands

| Button Name | Command Set | Description |
|---|---|---|
| List Channels | ch | Channel management commands. |
| Close Channel | ch -ci <#> | Close a channel by its number. |
| Create a Channel | cmd | Create a Command Prompt channel. |
| Switch a Channel | ch -si <#> | Switch to another channel (from Channel 0). |
| Show Kernel Log | d | Dump the current kernel log. |
| Toggle T-list | f | Toggles the information output by the t-list command, which shows processes only, or shows processes and threads. |
| List IP Information | i | List all IP network numbers and their IP addresses. |
| Set IP Parameters | i <#> <ip> <subnet> <gateway> | Set network interface number, IP address, subnet and gateway. |
| Display Server ID | id | Display the computer identification information. |
| Kill a Process | k <pid> | Kill the given process. |
| Lower Priority of a Process | l <pid> | Lower the priority of a process to the lowest possible. |

# Chapter 4 - Server Management

Table 16: Server Commands

| Button Name | Command Set | Description |
| --- | --- | --- |
| Lock Channels | lock | Lock access to Command Prompt channels. You must provide valid logon credentials to unlock a channel. |
| Limit Memory Usage of a Process | m <pid> <MB-allow> | Limit the memory usage of a process to <MB-allow>. |
| Paging | p | Causes t-list command output to pause after displaying one full screen of information. |
| Raise Priority of a Process | r <pid> | Raise the priority of a process by one. |
| Show Time and Date | s | Display the current time and date (24 hour clock used). |
| Set Time and Date | mm/dd/yyyy  hh:mm | Set the current time and date (24 hour clock used). |
| Show Task List | t | Tlist. |
| Crash Dump | crashdump | Crash the system. Crash dump must be enabled. |
| Restart | restart | Restart the system immediately. |
| Shutdown | shutdown | Shut down the system immediately. |

AlterPath Console Server

# Chapter 4 - Server Management

How it Works

**Sample configuration in pslave.conf:**

```
s1.tty ttyS1

s1.web_WinEMS 1

s1.users jen

s1.admin_users john
```

**Step 1: Log into the web interface of your ACS/TS.**

**Step 2: Go to the Navigation menu -> Applications section -> Windows EMS.**
       **Start managing your server by pressing on actions desired.**

# Chapter 4 - Server Management

This page has been left intentionally blank.

# Appendix A - New User Background Information

## Users and Passwords

A username and password are necessary to log in to the AlterPath Console Server. 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. |

AlterPath Console Server

# 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*<br>a) cp text.txt /tmp<br>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.<br>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*<br>a) mkdir spot<br>b) mkdir /tmp/snuggles | Creates a directory named directory_name.<br>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]<br>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. |

# Appendix A - New User Background Information

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 17: 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:

# Appendix A - New User Background Information

Table 18: 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 19: 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 20: 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 AlterPath Console Server has a static routing table that can be seen using the commands:

```
route
```

**or**

```
netstat -rn
```

The file /etc/network/st_routes is the AlterPath Console Server's method for configuring static routes. Routes should be added to the file (which is a script run when the AlterPath Console Server 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_msk* 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 AlterPath Console Server 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:            cyclades

ACS16 IP address:     192.168.160.1

host name:           acs16

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 cyclades:ttyS1@acs16

ssh -t cyclades:7001@acs16

ssh -t cyclades:10.0.0.1@acs16

ssh -t cyclades:file_server@acs16
```

# Appendix A - New User Background Information

```
ssh -t -l cyclades:10.0.0.1acs16
```

```
ssh -t -l cyclades:7001 acs16
```

**For openssh clients, version 3.1p1 or later ssh2 is the default. In that case, the -1 flag is used for ssh1.**

```
ssh -t cyclades:7001@acs16
```

```
ssh -t -2 cyclades:7001@acs16
```

```
ssh -t cyclades:7001@acs16
```
**(openssh 3.1p1 or later - Cyclades-TS V_1.3.2 or later/AlterPath Console Server version 2.1.0 or later -> ssh2 will be used)**

```
ssh -t -1 cyclades:7001@acs16
```
**(openssh 3.1p1 or later - AlterPath Console Server version 2.1.0 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@acs16
```
**Note: In this case, the file sshd_config must be changed in the following way:**

```
PermitRootLogin Yes
```

```
PermitEmptyPassword Yes
```

AlterPath Console Server

# Appendix A - New User Background Information

## The Session Channel Break Extension

This is a new feature for the AlterPath Console Server version 2.1.3 and Cyclades-TS version 1.3.7. The ACS/TS provides new way to send a break signal during a SSH version 2 terminal session. This method is defined by "Session Channel Break Extension : draft-ietf-secsh-break-00.txt." In previous ACS and TS versions there is one break lenght in milliseconds (break duration). Now the ACS and TS have a new parameter <all/Sx>.break_interval, which is used with all.break_sequence (<all/Sxx>.break_sequence). (This improves the SSH-break Cyclades implementation).

The ssh2-client receives a command ("<ssh escape char>B") from the user and sends one "break request" to ssh-server. The ssh-server receives the "break request" and sends a break command to the serial port. The ssh client can send the break duration (break interval), so the user can configure this value by command line (" -B <break interval in miliseconds> ") or by ssh_config file ("breakinterval <break interval in miliseconds>").

### How it works in SSH Server (all.protocol is socket_ssh)

The serial driver accepts the parameter *break interval* in the break command. If the version is 2 (ssh-2), the server accepts and treats the "break request" sent by the client. The "break request" defines the break-length in miliseconds. The server sends a break command with the break-length to the serial driver to perform the break in the serial port. If the parameter all.break_sequence is configured and the server finds the sequence in the data received from client, the server sends a break command with all.break_interval to serial driver.

### How it works in SSH Client

The SSH client has a new option "-B <break_interval in miliseconds>" and accepts break_interval in ssh_config. When the user types "<ssh-escape>B" (where ssh-escape is "~") the client sends a "break request" to ssh-server. When the ACS/TS calls the ssh-client automatically, it uses the parameter all.break_interval to calls the ssh-2 client.

### 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 <ACS_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 ACS's database).**

- **Client start-up command: ssh -t -l <username> <ACS_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 ACS's database).**

> **Note: For security reasons, some ssh clients do not allow just this type of authentication. To access the serial port, the ACS 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 T/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:**
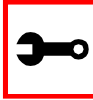
```
echo 'n 'client_hostname ' >> /etc/ssh/ssh_known_hosts or ~/.ssh/
known_hosts
```

```
cat /tmp/known_hosts >> /etc/ssh/ssh_known_hosts or ~/.ssh/
known_hosts
```

- **client start-up command: ssh -t <ACS_ip or Serial_port_ip>**

# Appendix A - New User Background Information

> **Note:** "client_hostname" should be the DNS name. To access the serial port, the ACS 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 ACS's *.equiv, ~/.?hosts, and *known_hosts files.

- Client identity created by ssh-keygen and its public part (~/.ssh/identity.pub) copied into  ACS's ~/.ssh/authorized_keys.

- Client start-up command: ssh -t <ACS_ip or Serial_port_ip>.

**Step 4:  Only PasswdAuthentication yes in sshd_config.**

- Removal of the ACS's *.equiv, ~/.?hosts, *known_hosts, and *authorized_keys files.

- Client startup command: ssh –t -l <username> <ACS_ip or Serial_port_ip> or ssh –t –l <username:alias><ACS_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 ACS's ~/.ssh/authorized_keys2 file.

- Password Authentication is performed if DSA key is not known to the ACS. Client start-up command: ssh -2 -t <TS_ip or Serial_port_ip>.
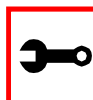
> **Note:** All files "~/*" or "~/.ssh/*" must be owned by the user and readable only by others. All files created or updated must have their full path and file name inside the file config_files and the command saveconf must be executed before rebooting the ACS.

# Appendix A - New User Background Information

Configuring the Session Channel Break Extension in SSH Server

**Step 1: Configure the parameter break_interval in pslave.conf.**
This can be done by the admin using the Web, snmpset, the Wizard or CLI.

**Step 2: Configure the parameter ssh_interval in ssh_config.**
This can be done using the vi editor.

## The Process Table

The process table shows which processes are running. Type *ps -a* to see a table similar to that below.

**Table 21: 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.

AlterPath Console Server

# 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 AlterPath Console Server. 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.Cyclades.com

3 az84.Cycladess.com

4 64.186.190.85

5 az85.Cyclades.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 AlterPath Console Server. If the user selects 2 through 5, remote serial ports will be accessed. This is

# Appendix A - New User Background Information

used when there is clustering (one AlterPath Console Server master box and one or more AlterPath Console Server 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.

```
AlterPath Console Server: 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.

```
AlterPath Console Server: 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
```

# Appendix A - New User Background Information

```
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 AlterPath Console Server.**

-**h : Lists script options.**

# Appendix A - New User Background Information

This page has been left intentionally blank.

# Appendix B - Cabling, Hardware, & Electrical

## General Hardware Specifications

The power requirements, environmental conditions and physical specifications of the AlterPath Console Server are listed below.

Table 22: AlterPath Console Server power requirements

| Power Specifications | | | | | | |
|---|---|---|---|---|---|---|
| | ACS1 | ACS4 | ACS8 | ACS16 | ACS32 | ACS48 |
| 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) | Internal 100 - 240VAC autorange (-48VDC option available) | Internal 100 - 240VAC autorange (-48VDC option available) | Internal 100-240VAC autorange (-48VDC option available) | Internal 100-240VAC autorange (-48VDC option available) | Internal 100-240VAC autorange (-48VDC option available) |
| Input Frequency Range | 50/60H | 50/60H | 50/60H | 50/60H | 50/60H | 50/60H |
| Power @120VAC | 5 W max | 16 W max | 18 W max | 22 W max | 26 W max | 11 W max |
| Power @220 VAC | 6 W max | 25 W max | 28 W max | 28 W max | 37 W max | 17 W max |

# Appendix B - Cabling, Hardware, & Electrical

Table 25: AlterPath Console Server environmental conditions

| Environmental Information | | | | | | |
|---|---|---|---|---|---|---|
| | ACS1 | ACS4 | ACS8 | ACS16 | ACS32 | ACS48 |
| Operating Temperature | 50F to 122F (10$^o$C to 50$^o$C) | 50F to 112F (10$^o$C to 44$^o$C) | 50F to 112F (10$^o$C to 44$^o$C) | 50F to 112F (10$^o$C to 44$^o$C) | 50F to 112F (10$^o$C to 44$^o$C) | 50F to 112F (10$^o$C to 44$^o$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 26: AlterPath Console Server physical conditions

| Physical Information | | | | | | |
|---|---|---|---|---|---|---|
| | ACS1 | ACS4 | ACS8 | ACS16 | ACS32 | ACS48 |
| External Dimensions | 2.76 x 3.35 x 1.18 in. | 8.5 in. x 4.75 in. x 1 in. | 8.5 in. x 4.75 in. x 1 in. | 17 in. x 8.5 in. x 1.75 in. | 17 in. x 8.5 in. x 1.75 in. | 17 in. x 8.5 in. x 1.75 in. |
| Weight | 0.3 lb. | 1.5 lb. | 1.6 lb. | 6 lb. | 6.2 lb. | 8 lb. |

Table 29: AlterPath Console Server safety specifications

| Safety Information | | | | | | |
|---|---|---|---|---|---|---|
| | ACS1 | ACS4 | ACS8 | ACS16 | ACS32 | ACS48 |
| Approvals | FCC and CE, Class A | | | | | |

The following section has all the information you need to quickly and successfully purchase or build cables to the AlterPath Console Server. It focuses on information related to the RS-232 interface, which applies not only to the AlterPath Console Server but also to any RS-232 cabling.

# Appendix B - Cabling, Hardware, & Electrical

## Rear Panel LEDs

The ACS' rear panel has connectors (serial, console and Ethernet) with some LEDs that have the following functionalities:

### Ethernet Connector

| | |
|---|---|
| *Col* <br> *(collision)* | Shows *collision* on the LAN every time the unit tries to transmit an Ethernet packet. |
| *DT/LK* <br> *(data transaction* <br> */link state* | DT flashes when there's data transmitted to or received from the LAN. It's hardware-controlled. LK keeps steady if the LAN is active. The green LED is *Data Transaction* activity and the yellow one is *LinK state*. |
| *100* | If 100BT is detected the LED lights on. If 10BT is detected it turns off. |

### Console Connector

| | |
|---|---|
| *CP* | CPU activity. It flashes at roughly 1 second intervals. |
| *P1* | Power supply #1 ON. |
| *P2* | Power supply #2 ON. |

### Serial Connector

| | |
|---|---|
| *LK* | DTR. It's software-controlled. |
| *DT* | Data transmitted to or received from the serial line. It's hardware-controlled. |

# 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,

AlterPath Console Server

# Appendix B - Cabling, Hardware, & Electrical

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 31: Cables and their pin specifications

| RS-232 Signal | Name/Function (Input/Output) | DB-25 pins (Standard) | DB-9 pins (Standard) | RJ-45 pins (Cyclades) |
|---------------|------------------------------|-----------------------|----------------------|-----------------------|
| 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 |

AlterPath Console Server

# 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 Cyclades or a cable vendor.

Table 32: Which cable to use

| To Connect To | Use Cable |
|---|---|
| DCE DB-25 Female (standard)<br><br>• Analog Modems<br><br>• ISDN Terminal Adapters | Cable 1:<br>RJ-45 to DB-25 M straight-through (Custom). This custom cable can be ordered from Cyclades or other cable vendors. A sample is included with the product ("straight-through"). |

# Appendix B - Cabling, Hardware, & Electrical

Table 32: Which cable to use

| To Connect To | Use Cable |
|---|---|
| DTE RJ-45 Cyclades (custom)<br><br>• All Cyclades Console Ports | Cable 2:<br>RJ-45 to RJ-45 crossover (custom). A sample is included with the product ("straight-through") This custom cable can be ordered from Cyclades or other cable vendors using the provided wiring diagram. |

## 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: Cyclades RJ-45 to DB-25 Male, straight-through

**Application: This cable connects Cyclades products (serial ports) to modems and other DCE RS-232 devices. It is included in both Cable Package #1 and #2.**
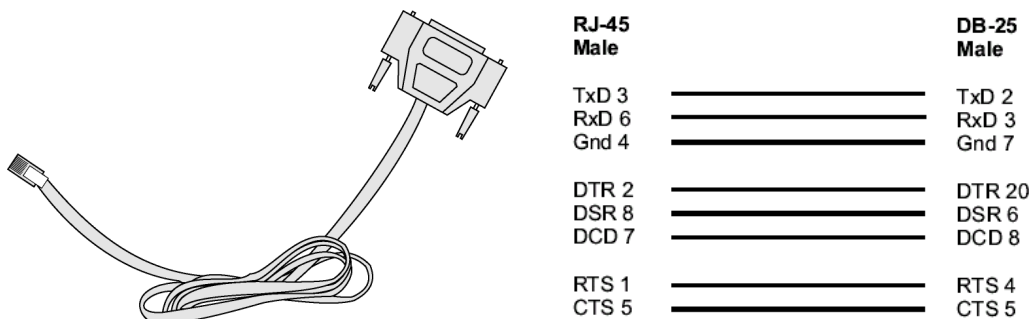


*Figure 33: Cable 1 - Cyclades RJ-45 to DB-25 Male, straight-through*

## Cable #2: Cyclades RJ-45 to DB-25 Female/Male, crossover

**This cable connects Cyclades 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.**
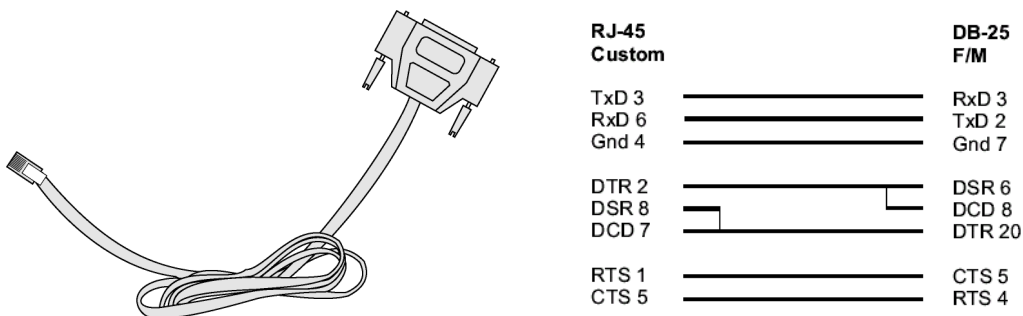


*Figure 34: Cable 2 - Cyclades RJ-45 to DB-25 Female/Male, crossover*

# Appendix B - Cabling, Hardware, & Electrical

## Cable #3: Cyclades RJ-45 to DB-9 Female, crossover

This cable connects Cyclades 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.



| RJ-45 Custom | | DB-9 Female |
|---|---|---|
| TxD 3 | —— | RxD 2 |
| RxD 6 | —— | TxD 3 |
| Gnd 4 | —— | Gnd 5 |
| DTR 2 | | DSR 6 |
| DSR 8 | | DCD 1 |
| DCD 7 | | DTR 4 |
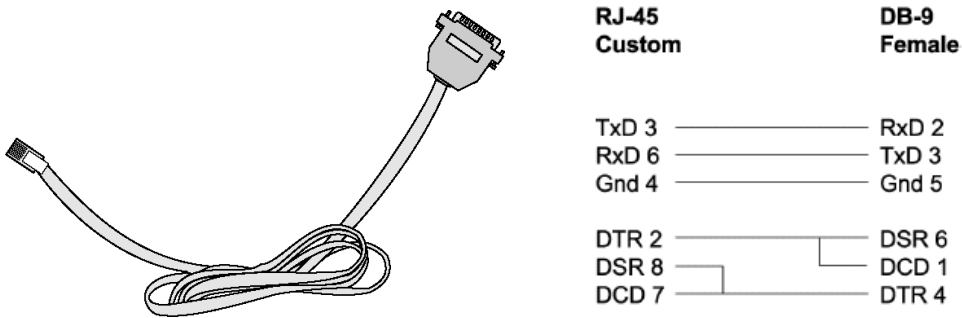
*Figure 35: Cable 3 - Cyclades RJ-45 to DB-9 Female, crossover*

## Cable #4: Cyclades RJ-45 to Cyclades 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.



| | RJ-45 Male | | RJ-45 Male | |
|---|---|---|---|---|
| TxD | 3 | —— | 3 | TxD |
| RxD | 6 | —— | 6 | RxD |
| GND | 4 | —— | 4 | GND |
| DTR | 2 | —— | 2 | DTR |
| DSR | 8 | —— | 8 | DSR |
| DCD | 7 | —— | 7 | DCD |
| RTS | 1 | —— | 1 | RTS |
| CTS | 5 | —— | 5 | CTS |
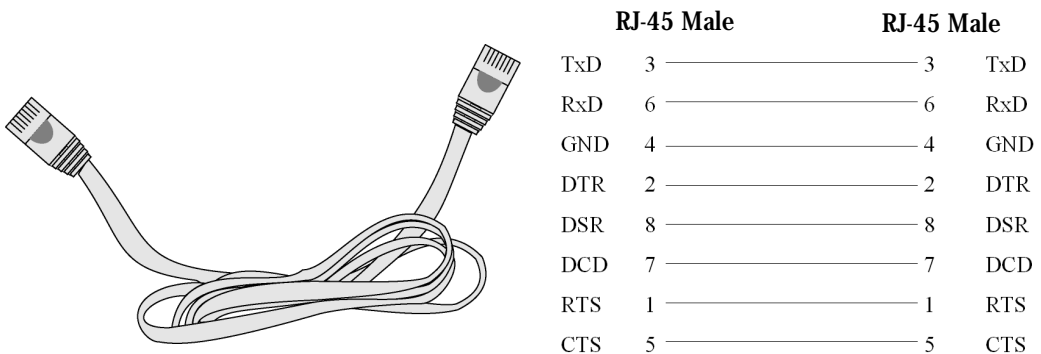
*Figure 36: Cable 4 - Cyclades RJ-45 to Cyclades RJ-45, straight-through*

AlterPath Console Server

# Appendix B - Cabling, Hardware, & Electrical

## Cable #5: Cyclades/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 Cyclades products to a Sun Netra server or to a Cisco product. This cable is included in Cable Package #2.



| PLUG | | | | JACK |
|------|---|---|---|------|
| (Sun Netra/Cisco) | | | | (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 |
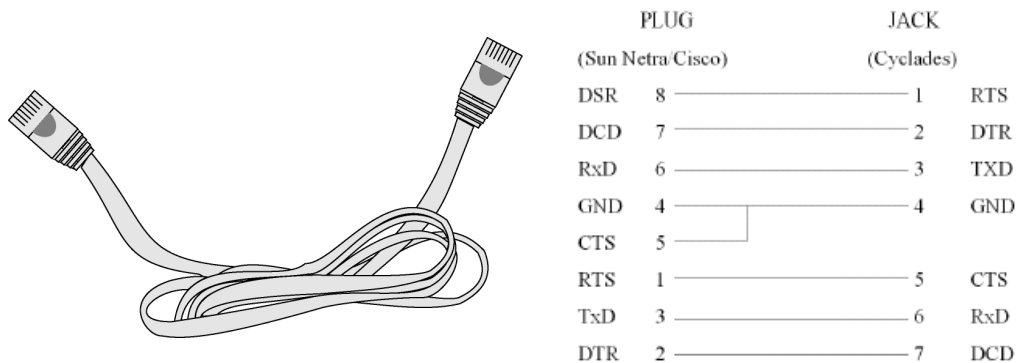
*Figure 37:  Cable 5 - Cyclades/Sun Netra Cable*

## 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.



| | |
|---|---|
| TxD | 2 |
| RxD | 3 |
| RTS | 4 |
| CTS | 5 |
| DSR | 6 |
| DCD | 8 |
| DTR | 20 |
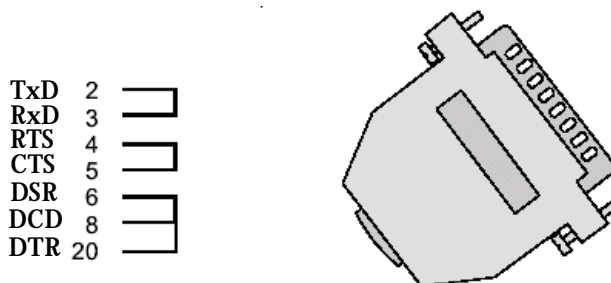
*Figure 38:  Loop-Back Connector*

# Appendix B - Cabling, Hardware, & Electrical

## Cyclades\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 Cyclades 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 | | | JACK | |
|---|---|---|---|---|
| (Sun Netra/Cisco) | | | (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 |
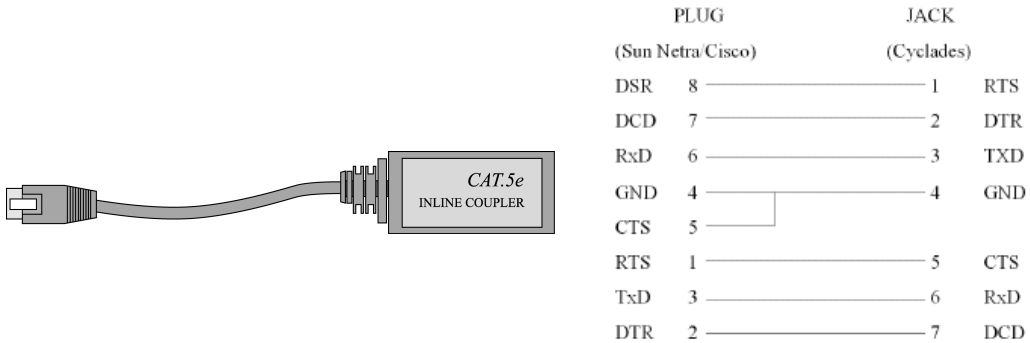
*Figure 39:  Cyclades\Sun Netra Adapter*

## 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 | | | |
| DSR | 8 | —— | 20 | DTR |
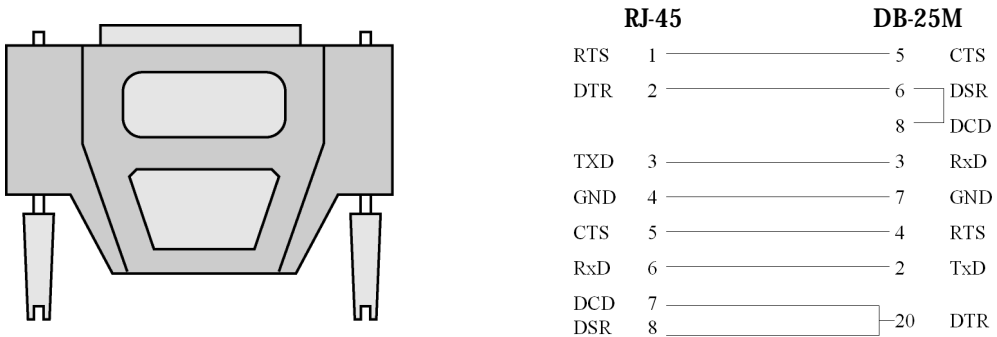
*Figure 40:  RJ-45 Female to DB-25 Male Adapter*

AlterPath Console Server

# 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.**
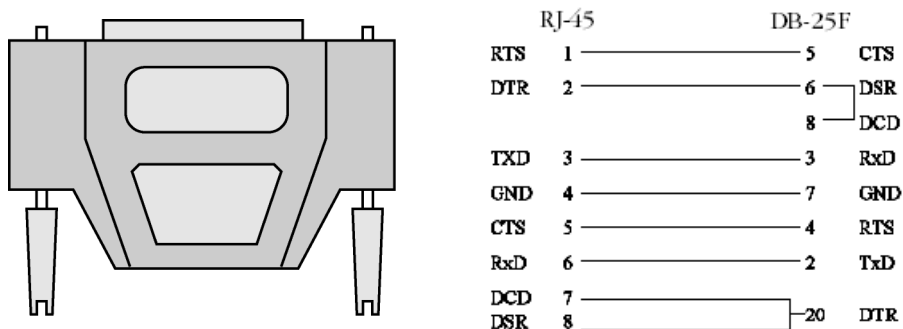


*Figure 41:  RJ-45 Female to DB-25 Female Adapter*

RJ-45 Female to DB-9 Female Adapter

**The following adapter may be necessary. This is included in Cable Package #1.**
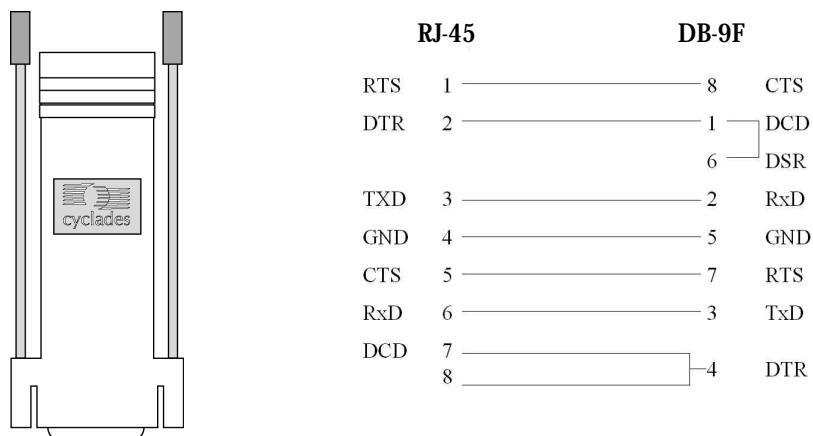


*Figure 42:  RJ-45 Female to DB-9 Female Adapter*

# Appendix B - Cabling, Hardware, & Electrical

## ACS1-only Cabling Information

### ACS1 Connectors

**Table 33: RS-485 Pinout for the ACS1 - Connector pin assignment**

| RS-485 Signal | Name/Function | Terminal Block pins |
|---|---|---|
| Chassis | Not in use | 1 |
| TXA- | Transmit Data - (A) | 2 |
| TXB+ | Transmit Data + (B) | 3 |
| RXA- | Receive Data - (A) | 4 |
| RXB- | Receive Data + (B) | 5 |
| Chassis | Not in use | 6 |
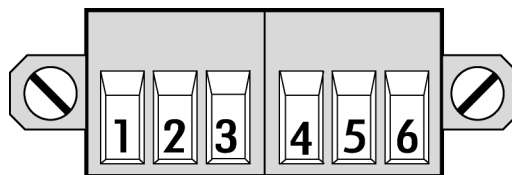


*Figure 43:  Terminal Block Pins*

# Appendix B - Cabling, Hardware, & Electrical

Cable #1: Terminal Block to Terminal Block, crossover half duplex

**Application: It connects the ACS1 (serial port) to DTE RS-485 devices with half duplex communication.**

Terminal Block

| Terminal Block | | Terminal Block |
| --- | --- | --- |
| RxD -5 | | RxD -5 |
| TxD -2 | | TxD -2 |
| RxD +4 | | RxD +4 |
| TxD +3 | | TxD +3 |
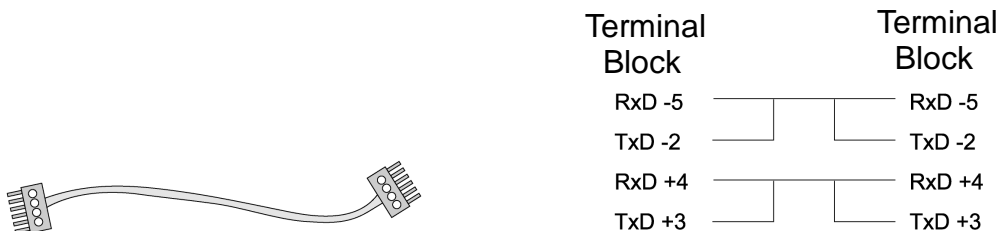
*Figure 44:  Cable 1 for the ACS1 - Terminal Block to Terminal Block, crossover half duplex*

Cable #2: Terminal Block to Terminal Block, crossover full duplex

**Application: It connects the ACS1 (serial port) to DTE RS-485 devices with full duplex communication.**

| Terminal Block | | Terminal Block |
| --- | --- | --- |
| RxD -5 | | TxD -2 |
| RxD +4 | | TxD +3 |
| TxD -2 | | RxD -5 |
| TxD +3 | | RxD +4 |
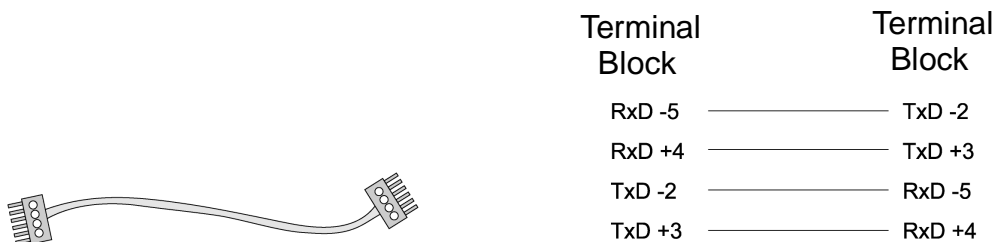
*Figure 45:  Cable 2 for the ACS1 - Terminal Block to Terminal Block, crossover full duplex*

# Appendix B - Cabling, Hardware, & Electrical

Cable #3: DB-9 Female to DB-25 Female, crossover

**This cable connects the ACS1 to console ports, terminals, printers and other DTE RS-232 devices. You will essentially have the cable shown in this picture:**
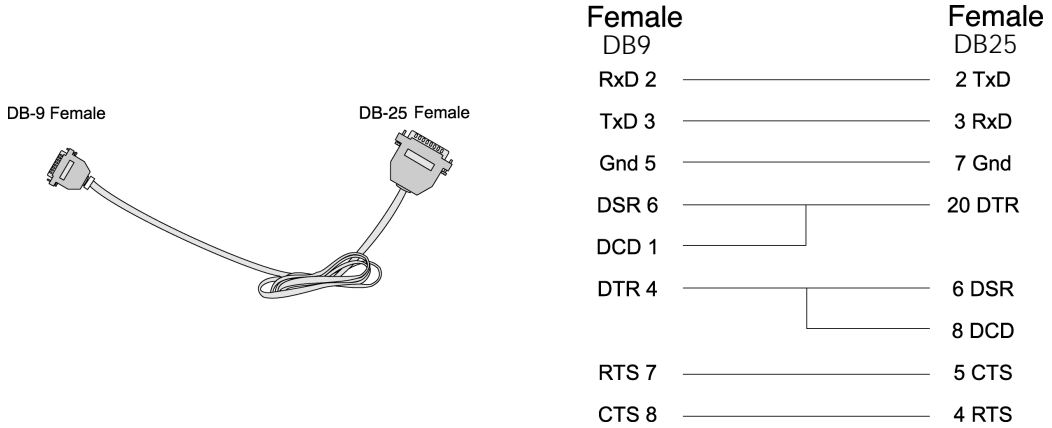


*Figure 46: Cable 3 for the ACS1 - DB-9 Female to DB-25 Female, crossover*

# 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 ACS box.

## Configuration Parameters

### CAS, TS, and Dial-in Common Parameters

The parameters on the following table are common to all three profiles:

Table 34: 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:<br>  0 - DHCP disabled<br>  1 - DHCP active<br>  2 - DHCP active and the unit saves in flash the last IP assigned by the DHCP server (default). | 1<br>Also see Description column. |
| conf.eth_ip_alias | Secondary IP address for the Ethernet inter-face (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<br>*Note: This is a parameter specific to TS profile.* | Ex: /bin/rlogin |

# Appendix C - The pslave Configuration File

Table 34: 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 AlterPath Console Server. It should not be changed unless the user decides to customize the operating system. | /var/lock |

# Appendix C - The pslave Configuration File

Table 34: 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 |

# Appendix C - The pslave Configuration File

Table 34: 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 AlterPath Console Server. \n represents a new line and \r represents a carriage return. Expansion characters can be used here. *Value for this Example:* <br><br> `\r\n\` <br> `Welcome to terminal server %h` <br> `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. <br> *Valid values for all products* : <br> rs232         - RS232 (default value). <br> rs232_half    - RS232 with RTS legcy half duplex <br> rs232_half_cts  - RS232 with RTS legacy half duplex and CTS control <br><br> *Valid values for the TS100/TS110/ACS1 only :* <br> rs485_half     - RS485 half duplex with out terminator <br> rs485_half_terminator - RS485 half duplex with terminator <br> rs485_full_terminator - RS485 full duplex with terminator <br> rs422         - alike rs485_full_terminator | See Description column |

# Appendix C - The pslave Configuration File

Table 34: 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 |

# Appendix C - The pslave Configuration File

Table 34: Parameters Common to CAS, TS, & Dial-in Access

| Parameter | Description | Value for this Example |
|-----------|-------------|------------------------|
| all.sttyCmd | The TTY is programmed to work as config-ured and this user-specific configuration is applied over that serial port. Parameters must be separated by a space. The following example sets : <br> *-igncr* <br> This tells the terminal not to ignore the car-riage-return on input, <br> *-onlcr* <br> Do not map newline character to a carriage return or newline character sequence on output, <br> *opost* <br> Post-process output, <br> *-icrnl* <br> Do not map carriage-return to a newline character on input. <br><br> `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. <br><br> Ex: "%g:%P.%3.%4" <br><br> %g : process id <br> %P : Protocol <br> %3 : Third nibble of remote IP <br> %J : Remote IP <br><br> Note: In the pslave.conf file there is a list of all expansion variables available. | See Description Column |

AlterPath Console Server

# Appendix C - The pslave Configuration File

Table 34: 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 |

# Appendix C - The pslave Configuration File

<p style="text-align:center">Table 34: Parameters Common to CAS, TS, & Dial-in Access</p>

| 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:<br><br>• *none* (no authentication)<br><br>• *local* (authentication is performed using the /etc/passwd file)<br><br>• *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.)<br><br>• *radius* (authentication is performed using a Radius authentication server)<br><br>• *TacacsPlus* (authentication is performed using a TacacsPlus authentication server)<br><br>• *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)<br><br>• *kerberos* (authentication is performed using a kerberos server. The IP address and other details of the kerberos server are defined in the file /etc/krb5.conf) | local |

# Appendix C - The pslave Configuration File

<p align="center">Table 34: Parameters Common to CAS, TS, & Dial-in Access</p>

| Parameter | Description | Value for this Example |
|---|---|---|
| | • *local/radius* (authentication is performed locally first, switching to Radius if unsuccessful)<br><br>• *radius/local* (the opposite of the previous option)<br><br>• *local/TacacsPlus* (authentication is performed locally first, switching to TacacsPlus if unsuccessful)<br><br>• *TacacsPlus/local* (the opposite of the previous option)<br><br>• *RadiusDownLocal* (local authentication is tried only when the Radius server is down)<br><br>• *TacacsPlusDownLocal* (local authentication is tried only when the TacacsPlus server is down)<br><br>• *kerberosDownLocal* (local authentication is tried only when the kerberos server is down)<br><br>• *ldapDownLocal* (local authentication is tried only when the ldap server is down) | |

# Appendix C - The pslave Configuration File

Table 34: Parameters Common to CAS, TS, & Dial-in Access

| Parameter | Description | Value for this Example |
|-----------|-------------|------------------------|
| all.authtype (cont.) | • *NIS* - All authentication types but NIS follow the format all.authtype <Authentication>DownLocal or <Authentication> (e.g. all.authtype radius or radiusDownLocal or ldap or ldapDownLocal, etc). NIS requires all.authtype to be set as local, regardless if it will be "nis" or its "Downlocal" equivalent. The service related to "nis" or its "Downlocal" equivalent would be configured in the /etc/nsswitch.conf file, not in the /etc/portslave/pslave.conf file.<br><br>Note that this parameter controls the authentication required by the AlterPath Console Server. The authentication required by the device to which the user is connecting is controlled separately. | |
| 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 milliseconds. It is valid if TTY protocol is socket_ssh,socket_server or ssh-2 (client). | |
| 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 |

AlterPath Console Server

# Appendix C - The pslave Configuration File

Table 34: 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 AlterPath Console Server 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 36, "TS Parameters," on page 377. The Dial-in configuration settings are in Table 37, "Dial-in configuration Parameters," on page 378. For Power Management, see the section "Appendix J - Power Management" on page 457. | socket_server |
| all.web_WinEMS | Defines whether or not management of Windows Emergency Management Service is allowed from the Web. | yes or 1, or no or 0 |

# Appendix C - The pslave Configuration File

Table 34: Parameters Common to CAS, TS, & Dial-in Access

| Parameter | Description | Value for this Example |
|---|---|---|
| all.xml_monitor | A non-zero value activates XML monitoring. All XML data received from the port is captured and sent to syslog-ng with facility LOCAL<DB_facility> and priority INFO. The format of the message is "XML_MONITOR (ttySx) [data]". XML tags are sent by Windows Server 2003 Emergency Management Services during boot or crash. You can read more on XML_MONITOR in: /etc/syslog-ng/syslog-ng.conf | 1 |
| all.translation | Defines whether or not to perform translation of Fn-keys (e.g. F8 key) from one terminal type to VT-UTF8. Currently only translation from xterm to VT-UTF8 is supported. | xterm |
| 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 |

AlterPath Console Server

# Appendix C - The pslave Configuration File

## CAS Parameters

You can configure additional CAS features with the parameters given on the following tables. (The  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 35: 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 |

# Appendix C - The pslave Configuration File

Table 35: Mostly CAS-specific Parameters

| Parameter | Description | Value for this Example |
|---|---|---|
| conf.nat_clustering_ip | IP address of any ACS interface (master box). It is a public IP address (e.g. Ethernet's interface IP address) and it is the one that must be used to connect the slave's serial ports. You can use the same value assigned to the Ethernet's IP address as that of the master box in the chain. | 64.186.161.108 |
| all.ipno | This is the default IP address of the AlterPath Console Server'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 milliseconds. It is valid if TTY protocol is socket_ssh, | socket_ server or ssh-2 (client) |

AlterPath Console Server

# Appendix C - The pslave Configuration File

Table 35: Mostly CAS-specific Parameters

| Parameter | Description | Value for this Example |
|---|---|---|
| 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 |
| 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 con-figure 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 uncom-mented and a string of bytes is set, matching occurs whenever there is not session estab-lished to the port. If this parameter is com-mented out, then no checking and matching occurs. (See more on the usage of this parame-ter in Terminal Appearance in Chapter 3.) | commented |

# Appendix C - The pslave Configuration File

Table 35: 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 config-ure 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 com-mented, then nothing will be sent back to the remote server even if all.auto_answer_input is uncommented. If this parameter is uncom-mented 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 AlterPath Console Server for this period of time, the AlterPath Console Server 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 |

AlterPath Console Server

# Appendix C - The pslave Configuration File

Table 35: 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 AlterPath Console Server 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.<br><br>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+ |

# Appendix C - The pslave Configuration File

Table 35: 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 AlterPath Console Server; 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 |

# Appendix C - The pslave Configuration File

Table 35: 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 |

# Appendix C - The pslave Configuration File

Table 35: 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 |

# Appendix C - The pslave Configuration File

Table 35: 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" |

# Appendix C - The pslave Configuration File

Table 35: 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, whereas no means sniffing is not permitted. 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 |

# Appendix C - The pslave Configuration File

Table 35: Mostly CAS-specific Parameters

| Parameter | Description | Value for this Example |
|---|---|---|
| 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 |
| 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 |
| s1.pool_ipno | This is the default IP of the AlterPath Console Server's pool of serial ports. Any host can access a port from the pool using its pool's IP address as long as a path to the address exists in the host's routing table. | 192.168.2.1 |

# Appendix C - The pslave Configuration File

Table 35: Mostly CAS-specific Parameters

| Parameter | Description | Value for this Example |
|---|---|---|
| s1.pool_socket_port | In the CAS profile, this defines an alternative labeling system for the AlterPath Console Server pool of ports. In this example, serial interface 1 is assigned to the pool identified by port value 3001. Using s<serial port #>.pool_socket_port one can assign each serial interface to a different pool of ports. One serial interface can belong to just one pool of ports. Each pool of ports can have any number of serial interfaces. | 3000 |
| s1.pool_serverfarm | Alias name given to the pool where this serial interface belong to. | pool_1 |
| 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 |

**AlterPath Console Server**

# Appendix C - The pslave Configuration File

## TS Parameters

The following parameters are unique to a TS setup except where indicated.

Table 36: TS Parameters

| Parameter | Description | Value for this Example |
|---|---|---|
| conf.telnet | Location of the telnet utility | /usr/bin/telnet |
| 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 AlterPath Console Server 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 ACS 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). | |

# Appendix C - The pslave Configuration File

Table 36: 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 37: 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. | |

**AlterPath Console Server**

# Appendix C - The pslave Configuration File

Table 37: 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 "" \\ "" ATM0 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 ACS, 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). | %i:%j novj \\ proxyarp modem asyncmap 000A0000 \\ noipx noccp login auth require-pap refuse-chap\\ mtu %t mru %t \\ cb-script /etc/portslave/cb_script \\ plugin /usr/lib/libpsr.so |

# Appendix C - The pslave Configuration File

Table 37: 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, ppp_only, slip, and cslip. | ppp |
| s32.tty | See the s1.tty entry in the CAS section. | ttyS32 |

# 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
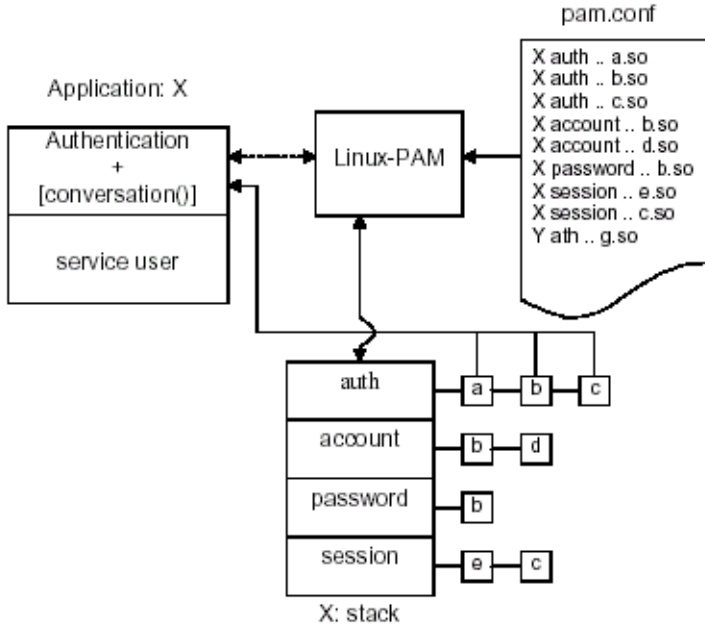


*Figure 47: Data flow diagram of Linux-PAM*

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.

# Appendix D - Linux-PAM

*Service-name*     The name of the service associated with this entry. Frequently the ser-
vice name is the conventional name of the given application. For exam-
ple, '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.

# Appendix D - Linux-PAM

*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.

# Appendix D - Linux-PAM

*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.

       AlterPath Console Server

# Appendix D - Linux-PAM

| | |
|---|---|
| *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 AlterPath Console Server 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. |

# Appendix D - Linux-PAM

*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_krb5*     The Kerberos module currently used is pam_krb5. This PAM module requires the MIT 1.1+ release of Kerberos, or the Cygnus CNS distribution. It has not been tested against heimdal or any other Kerberos distributions. Important file: */etc/krb5.conf.* The krb5.conf file contains Kerberos configuration information, including the locations of KDCs and admin servers for the Kerberos realms of interest, defaults for the current realm and for Kerberos applications, and mappings of hostnames onto Kerberos realms. Normally, you should install your krb5.conf file in the directory/etc. You can override the default location by setting the environment variable KRB5_CONFIG.

# Appendix D - Linux-PAM

*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
```

## 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).

# Appendix D - Linux-PAM

*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_*         This argument is not currently supported by any of the modules in
*pass*                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 configura-
tion file for the named service. The AlterPath Console Server 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
```

# Appendix D - Linux-PAM

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.

## LDAP server for Linux

**Step 1:  The RPMs required for the LDAP servers are:**

- **db (Sleepycat Berkeley Database)**

- **openssl (OpenSSL)**

- **openldap (OpenLDAP)**

It's possible also to load the source codes and compile them, but it is easier to load these RPMs from the RedHat CDs.

**Step 2:  Go to the directory /etc/openldap or /usr/local/etc/openldap.**

NOTE: the example uses /usr/local path. Change all references of /usr/local if the path is different, and check if the directory/file really exists.

```
cd /usr/local/etc/openldap
```

**Step 3: Create the certificates:**

```
ln -s /usr/local/bin/openssl .

ln -s /usr/local/ssl/misc/CA.pl .

PATH=$PATH:.

CA.pl -newca <-- anwer questions, you MUST fill in "commonName"

CA.pl -newreq <-- repeat

CA.pl -signreq

mv newreq.pem ldapkey.pem

chmod 0600 ldapkey.pem

mv newcert.pem ldapcert.pem
```

# Appendix D - Linux-PAM

**Step 4:  Edit slapd.conf. The basic configuration to make it work is:**

```
include /usr/local/etc/openldap/schema/core.schema

include /usr/local/etc/openldap/schema/cosine.schema


pidfile /usr/local/var/slapd.pid

argsfile /usr/local/var/slapd.args


TLSCipherSuite HIGH:MEDIUM:+SSLv2

TLSCertificateFile /usr/local/etc/openldap/ldapcert.pem

TLSCertificateKeyFile /usr/local/etc/openldap/ldapkey.pem

TLSCACertificateFile /usr/local/etc/openldap/demoCA/cacert.pem


database bdb

suffix "dc=cyclades,dc=com,dc=br"

rootdn "cn=admin,dc=cyclades,dc=com,dc=br"

rootpw bitadmin


directory /usr/local/var/openldap-data

index objectClass eq
```

**Step 5:  Start LDAP server. This is done by the command:**

```
/usr/local/libexec/slapd -h "ldap:/// ldaps:///"
```

**This will allow the LDAP server accept both secured mode and non-secure mode.**

# Appendix D - Linux-PAM

**Step 6: Add entries.**

       **Example:**

```
ldapadd -x -D "cn=admin,dc=cyclades,dc=com,dc=br" -w bitadmin

dn: uid=helio,dc=cyclades,dc=com,dc=br

objectClass: person

objectClass: uidobject

uid: helio

cn: Helio Fujimoto

sn: Fujimoto

userPassword: bithelio
```

       **To list the entries:**

```
ldapsearch -x -D "cn=admin,dc=cyclades,dc=com,dc=br" -w bitadmin

'(objectClass=*)'
```

**This is enough to set up a LDAP server with some users, for PAM authentication purposes. In order to configure the TS/ACS:**

**Step 1: Configure all.protocol as ldap, in /etc/portslave/pslave.conf**

**Step 2: Edit /etc/ldap.conf. Edit the following parameters:**

```
host 200.246.93.95 <== LDAP server IP address or name

base dc=cyclades,dc=com,dc=br <== distinguished name of the search
base

uri ldaps://200.246.93.95 <== to use secure LDAP
```

# Appendix D - Linux-PAM

For Active Directory

**A Windows 2000 or Windows 2003 Server edition is necessary. The installation, configuration and the users insertions were done with Djalma. It is not that difficult.**

**In the TS/ACS side, the /etc/ldap.conf must be configured.**

What needs to be set in the /etc/ldap.conf

```
# The Windows 2003 server IP address
host 200.246.93.118


# The Distinguished name (In our active directory, the format was set
# to CycladesCorporation.local)
base dc=CycladesCorporation,dc=local


# Here you can insert any user you had created, or the administrator
user. I set
# the administrator user. I could not make the AD accept anonymous users to
# bind.
binddn cn=Administrator,cn=Users,dc=CycladesCorporation,dc=local


# Password for that user
bindpw Teste123


# PAM login attribute
pam_login_attribute sAMAccountName


# Update Active Directory password, by
# creating Unicode password and updating
# unicodePwd attribute.
pam_password ad
```

# 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.

**AlterPath Console Server**

# 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. Cyclades 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
```

# Appendix D - Linux-PAM

```
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
```

# Appendix  D  -  Linux-PAM

```
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
#
```

# Appendix D - Linux-PAM

```
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
#
```

# Appendix D - Linux-PAM

```
# 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**

**AlterPath Console Server**

# Appendix E - Upgrades and Troubleshooting

## Upgrades

Users should upgrade the AlterPath Console Server whenever there is a bug fix or new features that they would like to have. Below are the six files added by Cyclades 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 allAlterPath Console Server configuration information is stored

### The Upgrade Process

To upgrade the AlterPath Console Server, follow these steps:

**Step 1:  Log in to the ACS 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 ACS 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 Cyclades 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 the following to see the Linux kernel version:

```
cat /proc/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 ACS 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 AlterPath Console Server 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 AlterPath Console Server 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 38: 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... |

# Appendix E - Upgrades and Troubleshooting

Table 38: 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, ACS 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 AlterPath Console Server administrator must execute the command *saveconf* before rebooting the AlterPath Console Server 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!** Cyclades 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 AlterPath Console Server version written to the screen. This will speed the resolution of most problems.

## Hardware Test

A hardware test called *tstest* is included with the AlterPath Console Server firmware. It is a menu-driven program, run by typing tstest at the command prompt. The various options are described below. Note that the AlterPath Console Server 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). When tstest senses the presence of the cable or connector, the test will be run automatically and the result shown on the screen.

Each 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.

# Appendix E - Upgrades and Troubleshooting

| | | <- 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*

# Appendix E - Upgrades and Troubleshooting

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.

## Single User Mode

The AlterPath Console Server 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 AlterPath Console Server unstable,

- After a configuration change which leaves the AlterPath Console Server 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
```

# Appendix E - Upgrades and Troubleshooting

```
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 AlterPath Console Server 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

# Appendix E - Upgrades and Troubleshooting

your system. If your ftp server is on the same network as the ACS, 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 AlterPath Console Server 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 AlterPath Console Server 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.

## 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]
```

# Appendix E - Upgrades and Troubleshooting

```
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 39: 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 40: 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 40: 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 Cyclades 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.**

# Appendix F - Certificate for HTTP Security

**Step E:** **Save the configuration in flash.**

```
#saveconf
```

**Step F:** **The certification will be effective in the next reboot.**

# Appendix F - Certificate for HTTP Security

This page has been left intentionally blank.

AlterPath Console Server

# Appendix G - IPSEC

## Introduction

The IPsec protocol provides encryption and authentication services at the IP level of the network protocol stack. Working at this level, IPsec can protect any traffic carried over IP, unlike other encryption which generally protects only a particular higher-level protocol (PGP for mail, SSH for login, SSL for Web work and so on). The implementation of IPsec used by the AlterPath Console Server is FreeSWAN (www.freeswan.org).

IPsec can be used on any machine which does IP networking. Dedicated IPsec gateway machines can be installed wherever required to protect traffic. IPsec can also run on routers, on firewall machines, on various application servers, and on end-user desktop or laptop machines.

IPsec is used mainly to construct a secure connection (tunnel) between two networks (ends) over a not-necessarily-secure third network. In our case, the IPsec will be used to connect the ACS securely to a host or to a whole network configurations frequently called host-to-network and host-to-host tunnel. Considering practical aspects, this is the same thing as a VPN, but here one or both sides have a degenerated subnet (only one machine).

## Applications of IPsec

Because IPsec operates at the network layer, it is remarkably flexible and can be used to secure nearly any type of Internet traffic. Two applications, however, are extremely widespread:

- A Virtual Private Network, or VPN, allows multiple sites to communicate with the Console Server securely over an insecure Internet by encrypting all communication between the sites and the Console Server.

- Road Warriors connect to the Console Server from home, or perhaps from a hotel somewhere.

A somewhat more detailed description of each of these applications is below. Our Quick Start section will show you how to build each of them.

# Appendix G - IPSEC

## Using secure tunnels to create a VPN

A VPN, or Virtual Private Network lets the Console Server and a whole network communicate securely when the only connection between them is over a third network which is not trustable. The method is to put a security gateway machine in the network and create a security tunnel between the Console Server and this gateway. The gateway machine and the Console Server encrypt packets entering the untrusted net and decrypt packets leaving it, creating a secure tunnel through it.

## Road Warriors

The prototypical Road Warrior is a traveler connecting to the Console Server from a laptop machine. For purposes of this document:

- Anyone with a dynamic IP address is a Road Warrior.

- Any machine doing IPsec processing is a gateway. Think of the single-user Road Warrior machine as a gateway with a degenerate subnet (one machine: itself) behind it.

These require a somewhat different setup than VPN gateways with static addresses and with client systems behind them, but are basically not problematic. There are some difficulties which appear for some Road Warrior connections:

- Road Warriors who get their addresses via DHCP may have a problem. FreeS/WAN can quite happily build and use a tunnel to such an address, but when the DHCP lease expires, FreeS/WAN does not know that. The tunnel fails, and the only recovery method is to tear it down and rebuild it.

- If Network Address Translation (NAT) is applied between the two IPsec Gateways, this breaks IPsec. IPsec authenticates packets on an end-to-end basis, to ensure they are not altered en route. NAT rewrites packets as they go by.

In most situations, however, FreeS/WAN supports Road Warrior connections just fine.

AlterPath Console Server

# Appendix G - IPSEC

## Quick Start Guide

This is a quick guide to set up two common configurations: VPN and Road Warrior. There are two examples: a Road Warrior using RSA signature and a VPN using RSA signature. When listing the configuration of the remote side (the equipment the AlterPath Console Server will create a tunnel with) these examples will assume the other end is also running the FreeS/Wan. If it is not your case, make the appropriate conversions for your IPsec software.

### Before You Start

#### Setup and test networking
Before trying to get FreeS/WAN working, you should configure and test IP networking on the Console Server and on the other end. IPsec cannot function without a working IP network beneath it.

Many reported FreeS/WAN problems turn out to actually be problems with routing or firewalling. If any actual IPsec problems turn up, you often cannot even recognize them (much less debug them) unless the underlying network is right.

#### Enabling IPsec on your AlterPath Console Server
The IPsec is disabled by default in the Console Server family. To enable it you must edit the file /etc/inittab and /etc/config_files and uncomment the lines regarding the IPsec. After performing these changes you must save the configuration using the saveconf tool and reboot the equipment.

### "Road Warrior" configuration
Think about the administrator that wants to access the ACS securely from wherever he is, from his office desk, from his house, or from the hotel room. His IP address will not be always the same, so, for IPsec purposes, he is a "Road Warrior." We refer to the remote machines as Road Warriors. For purposes of IPsec, anyone with a dynamic IP address is a Road Warrior.

# Appendix G - IPSEC

## Necessary Information

To set up a Road Warrior connection, you need some information about the system on the other end. Connection descriptions use left and right to designate the two ends. We adopt the convention that, from the Console Server's point of view, left=local and right =remote. The Console Server administrator needs to know some things about each Road Warrior:

- The system's public key (for RSA only).

- The ID that system uses in IPsec negotiation.

To get system's public key in a format suitable for insertion directly into the Console Server's ipsec.conf file, issue this command on the warrior machine:

```
/usr/local/sbin/ipsec showhostkey --right
```

The output should look like this (with the key shortened for easy reading):

```
rightrsasigkey=0s1LgR7/oUM...
```

The Road Warrior needs to know:

- The Console Server's public key or the secret, and

- The ID the Console Server uses in IPsec negotiation.

which can be generated by running:

```
/usr/local/sbin/ipsec showhostkey --left
```

on the Console Server. Each warrior must also know the IP address of the Console Server. This information should be provided in a convenient format, ready for insertion in the warrior's ipsec.conf file. For example:

```
left=1.2.3.4 leftid=@acs.example.com leftrsasigkey=0s1LgR7/oUM...
```

The Console Server administrator typically needs to generate this only once. The same file can be given to all warriors.

AlterPath Console Server

# Appendix G - IPSEC

Setup on the "Road Warrior" machine

**Simply add a connection description us-to-Console Server, with the left and right information you gathered above to the ipsec.conf file of the warrior system. This might look like:**

```
# pre-configured link to Console Server
conn us-to-acs
     # information obtained from Console Server admin
     left=1.2.3.4 # Console Server IP address
     leftid=@acs.example.com
     # real keys are much longer than shown here
     leftrsasigkey=0s1LgR7/oUM...
     # warrior stuff
     right=%defaultroute
     rightid=@xy.example.com
     rightrsasigkey=0s1LgR7/oUM
     # Start this connection when IPsec starts
     auto=start
```

> ⚠ **Important!** The connection name line: "conn us-to-acs" must start on the FIRST column of the line. All other lines after that line must be indented by 1 TAB. This is MANDATORY.

Setup on the AlterPath Console Server

**Adding Road Warrior support so people can connect remotely to your Console Server is straightforward. Just create the file /etc/warrior.connection and add the following lines to this file:**

```
conn gate-xy
     left=1.2.3.4
     leftid=@acs.example.com
     leftrsasigkey=0s1LgR7/oUM...
     # allow connection attempt from any address
     # attempt fails if caller cannot authenticate
     right=%any
     # authentication information
     rightid=@xy.example.com
```

# Appendix G - IPSEC

```
rightrsasigkey=0s1LgR7/oUM...
# Add this connection to the database when IPsec starts
auto=add
```

> ⚠️ **Important!**  The connection name line: "conn gate-xy" must start on the FIRST column of the line. All other lines after that line must be indented by 1 TAB. This is MANDATORY.

## VPN configuration

Often it may be useful to have explicitly configured IPsec tunnels between the Console Server and a gateway of an office with a fixed IP address (in this case every machine on the office network would have a secure connection with the Console Server), or between the Console Server and the Console Server administrator machine, which must, in this case, have a fixed IP address.

To do it just insert this connection description in your ipsec.conf file with the variables that fit your environment:

```
# sample tunnel
# The network here looks like:
# ACS ----acsnexthop......rightnexthop----right====rightsubnet
# If ACS and right are on the same Ethernet, omit leftnexthop and
rightnexthop.
conn sample
        # ACS
        left=10.0.0.1
        leftid=@acs.example.com
        # next hop to reach right
        leftnexthop=10.44.55.66
        # This line is only for RSA signature
        leftrsasigkey=0s1LgR7/oUM...
        # right s.g., subnet behind it, and next hop to reach left
        right=10.12.12.1
        rightid=@xy.example.com
        rightnexthop=10.88.77.66
```

# Appendix G - IPSEC

```
rightsubnet=192.168.0.0/24
# Start this connection when IPsec starts
auto=start
# This line is for RSA signature
rightrsasigkey=0s1LgR7/oUM...
```

> ⚠️ **Important!**  The connection name line: "conn sample" must start on the FIRST column of the line. All other lines after that line must be indented by 1 TAB. This is MANDATORY.

The good part is that this connection descriptor can be added to both the Console Server and the other end. This is the advantage of using left and right instead of using local remote parameters.

If you give an explicit IP address for left (and left and right are not directly connected), then you must specify leftnexthop (the router which Console Server sends packets to in order to get them delivered to right). Similarly, you may need to specify rightnexthop (vice versa). The nexthop parameters are needed because of an unfortunate interaction between FreeS/ WAN and the Linuxkernel routing code. They will be eliminated in a future release.

## Authentication Keys

To build a connection, the Console Server and the other end must be able to authenticate each other. For FreeS/WAN, the default is public key authentication based on the RSA algorithm. IPsec does allow several other authentication methods. On this chapter you will learn how to generate authentication keys and how to exchange keys between systems.

### Generating an RSA key pair

The Console Server doesn't have an RSA key pair by default. It will be generated on the first reboot after you have uncommented the IPsec lines in the file /etc/inittab. You also can generate your key pair by issuing the following commands as root:

# Appendix G - IPSEC

```
/usr/local/sbin/ipsec newhostkey --bits <key length> --output /
etc/ipsec.secrets
chmod 600 /etc/ipsec.secrets
```

Key generation may take some time. In addition, the Console Server needs a lot of random numbers and therefore needs and uses traffic on the Ethernet to generate them. It is also possible to use keys in other formats, not generated by FreeS/WAN. This may be necessary for interoperation with other IPsec implementations.

## Exchanging authentication keys

Once your ACS' key is in ipsec.secrets, the next step is to send your public key to everyone you need to set up connections with and collect their public keys. To extract the public part in a suitable format you can use the ipsec_showhostkey command. For VPN or Road Warrior applications, use one of the following:

```
# if your AlterPath Console Server is the left side of the tunnel
/usr/local/sbin/ipsec showhostkey --left

# if your AlterPath Console Server is the right side of the tunnel
/usr/local/sbin/ipsec showhostkey --right
```

These two produce the key formatted for insertion in an ipsec.conf file. Public keys need not be protected as fanatically as private keys. They are intended to be made public; the system is designed to work even if an enemy knows all the public keys used. You can safely make them publicly accessible. For example, put a gateway key on a Web page or make it available in DNS, or transmit it via an insecure method such as email.

# Appendix G - IPSEC

## IPsec Management

After you have all the configuration done you need to manage all tunnels and manage IPsec itself. This section will show you a few commands that have proven to be useful when managing IPsec and IPsec connections.

### The IPsec Daemon

The IPsec daemon (PLUTO) is the program that loads and negotiates the connections. To start the IPsec daemon use the following command:

```
/usr/local/sbin/ipsec setup --start
```

Similarly, this command accepts the usual daemon commands as stop and restart.

The ipsec daemon is not automatically initialized when you boot your Console Server equipment for the first time. If you want the IPsec to auto run on boot you must uncomment the lines regarding the IPsec on the /etc/rc.sysinit script.

### Adding and Removing a Connection

All the connections can be loaded to the IPsec database at boot time if these connections have the auto parameter set to *add*. However if a certain connection doesn't have this option set and you wish to add this connection manually you can use the following command:

```
/usr/local/sbin/ipsec auto --add <connection name>
```

Similarly, to take a connection out of the IPsec database you can use the command:

```
/usr/local/sbin/ipsec auto --delete <connection name>
```

Once a connection descriptor is in the IPsec internal database, IPsec will accept the other end to start the security connection negotiation. You can also start its negotiation as explained in the next section.

# Appendix G - IPSEC

## Starting and Stopping a Connection

All the connections can be negotiated at boot time if these connections have the auto parameter set to *start*. However if a certain connection doesn't have this option set, you can set it. Once a connection descriptor is in the IPsec internal database, you can start its negotiation using the command:

```
/usr/local/sbin/ipsec auto --up <connection name>
```

Similarly to close a tunnel you use the command:

```
/usr/local/sbin/ipsec auto --up <connection name>
```

Below you can see the output of a successful up operation:

```
[root@henrique root]# ipsec auto --up teste
104 "teste" #5: STATE_MAIN_I1: initiate
106 "teste" #5: STATE_MAIN_I2: sent MI2, expecting MR2
108 "teste" #5: STATE_MAIN_I3: sent MI3, expecting MR3
004 "teste" #5: STATE_MAIN_I4: ISAKMP SA established
112 "teste" #6: STATE_QUICK_I1: initiate
004 "teste" #6: STATE_QUICK_I2: sent QI2, IPsec SA established
```

## Debugging Commands

If you are having problems with IPsec you can use the following commands:

### IPsec look

It gives you a detailed information about the IPsec state:

```
[root@henrique root]# ipsec look
henrique Mon Oct 28 16:40:24 PST 2002
64.186.161.96/32 -> 64.186.161.128/32 => tun0x1006@64.186.161.128
esp0x4e1a10ce@64.186.161.128 (0)
ipsec0->eth0 mtu=16260(1443)->1500

esp0x4e1a10ce@64.186.161.128 ESP_3DES_HMAC_MD5: dir=out
src=64.186.161.96 iv_bits=64bits iv=0xd491678073a22185 ooowin=64
```

# Appendix G - IPSEC

```
alen=128 aklen=128 eklen=192 life(c,s,h)=addtime(4,0,0)

esp0xa99f2a63@64.186.161.96 ESP_3DES_HMAC_MD5: dir=in
src=64.186.161.128 iv_bits=64bits iv=0x46209cee5f952117 ooowin=64
alen=128 aklen=128 eklen=192 life(c,s,h)=addtime(4,0,0)

tun0x1005@64.186.161.96 IPIP: dir=in src=64.186.161.128
policy= 64.186.161.128/32->64.186.161.96/32 flags=0x8<>
life(c,s,h)=addtime(4,0,0)

tun0x1006@64.186.161.128 IPIP: dir=out src=64.186.161.96
life(c,s,h)=addtime(4,0,0)

Destination Gateway Genmask Flags MSS Window irtt Iface
0.0.0.0 64.186.161.1 0.0.0.0 UG 40 0 0 eth0
64.186.161.0 0.0.0.0 255.255.255.0 U 40 0 0 eth0
64.186.161.0 0.0.0.0 255.255.255.0 U 40 0 0 ipsec0
64.186.161.128 64.186.161.128 255.255.255.255 UGH 40 0 0 ipsec0
```

**In this output you can see that there is an activated tunnel between the networks 64.186.161.96/32 and 64.186.161.128/32. You can also see the routing table for this host after the encryption information .**

## IPsec whack

**The ipsec whack command show the status of the connections.**

```
[root@henrique root]# ipsec whack --status
000 interface ipsec0/eth0 64.186.161.96
000
000 "teste": 64.186.161.96[@micro]...64.186.161.128[@ACS ]
000 "teste": ike_life: 3600s; ipsec_life: 28800s; rekey_margin: 540s;
rekey_fuzz: 100%; keyingtries: 0
000 "teste": policy: RSASIG+ENCRYPT+TUNNEL+PFS; interface: eth0; erouted
000 "teste": newest ISAKMP SA: #5; newest IPsec SA: #6; eroute owner: #6
000
000 #6: "teste" STATE_QUICK_I2 (sent QI2, IPsec SA established);
EVENT_SA_REPLACE in 28245s; newest IPSEC; eroute owner
000 #6: "teste" esp.4e1a10ce@64.186.161.128 esp.a99f2a63@64.186.161.96
tun.1006@64.186.161.128 tun.1005@64.186.161.96
000 #5: "teste" STATE_MAIN_I4 (ISAKMP SA established); EVENT_SA_REPLACE in
```

# Appendix G - IPSEC

```
3019s; newest ISAKMP
```

As you can see, it shows almost the same information shown by the ipsec auto -up command. You can use this command if the up command doesn't show anything on the screen (it can happen depending on the ACS syslog configuration).

## The IPsec Configuration Files in Detail

This section will describe the file /etc/ipsec.conf in detail.

### Description

The ipsec.conf file specifies most configuration and control information for the FreeS/WAN IPsec subsystem. (The major exception is secrets for authentication; ipsec.secrets) Its contents are not security-sensitive unless manual keying is being done for more than just testing, in which case the encryption/authentication keys in the descriptions for the manually-keyed connections are very sensitive (and those connection descriptions are probably best kept in a separate file, via the include facility described below).

The file is a text file, consisting of one or more sections. White space followed by # followed by anything to the end of the line is a comment and is ignored, as are empty lines which are not within a section.

A line which contains include and a file name, separated by white space, is replaced by the contents of that file, preceded and followed by empty lines. If the file name is not a full pathname, it is considered to be relative to the directory containing the including file. Such inclusions can be nested. Only a single filename may be supplied, and it may not contain white space, but it may include shell wildcards for example:

```
include ipsec.*.conf
```

The intention of the include facility is mostly to permit keeping information on connections, or sets of connections, separate from the main configuration file. This permits such connec-

AlterPath Console Server

# Appendix G - IPSEC

tion descriptions to be changed, copied to the other security gateways involved, etc., without having to constantly extract them from the configuration file and then insert them back into it. Note the also parameter (described below) which permits splitting a single logical section (e.g., a connection description) into several actual sections.

A section begins with a line of the form:

```
type name
```

where type indicates what type of section follows, and name is an arbitrary name which distinguishes the section from others of the same type. (Names must start with a letter and may contain only letters, digits, periods, underscores, and hyphens.) All subsequent non-empty lines which begin with white space are part of the section; comments within a section must begin with white space too. There may be only one section of a given type with a given name.

Lines within the section are generally of the following form:

```
        parameter=value
```

(Note the mandatory preceding TAB.) There can be white space on either side of the =. Parameter names follow the same syntax as section names, and are specific to a section type. Unless otherwise explicitly specified, no parameter name may appear more than once in a section.

An empty value stands for the system default value (if any) of the parameter, i.e., it is roughly equivalent to omitting the parameter line entirely. A value may contain white space only if the entire value is enclosed in double quotes ("); a value cannot itself contain a double quote, nor may it be continued across more than one line.

Numeric values are specified to be either an integer (a sequence of digits) or a decimal number (sequence of digits optionally followed by . and another sequence of digits).

There is currently one parameter which is available in any type of section:

also

The value is a section name; the parameters of that section are appended to this section, as if they had been written as part of it. The specified section must exist, must follow the current

# Appendix G - IPSEC

one, and must have the same section type. (Nesting is permitted, and there may be more than one also in a single section, although it is forbidden to append the same section more than once.) This allows, for example, keeping the encryption keys for a connection in a separate file from the rest of the description, by using both an also parameter and an include line.

A section with name %default specifies defaults for sections of the same type. For each parameter in it, any section of that type which does not have a parameter of the same name gets a copy of the one from the %default section. There may be multiple %default sections of a given type, but only one default may be supplied for any specific parameter name, and all %default sections of a given type must precede all non-%default sections of that type. %default sections may not contain also parameters.

Currently there are two types of sections: a config section specifies general configuration information for IPsec, while a conn section specifies an IPsec connection.

## Conn Sections

A conn section contains a connection specification, defining a network connection to be made using IPsec. The name given is arbitrary, and is used to identify the connection to ipsec_auto and ipsec_manual. Here's a simple example:

```
conn snt
      left=10.11.11.1
      leftsubnet=10.0.1.0/24
      leftnexthop=172.16.55.66
      right=192.168.22.1
      rightsubnet=10.0.2.0/24
      rightnexthop=172.16.88.99
      keyingtries=0 # be very persistent
```

To avoid trivial editing of the configuration file to suit it to each system involved in a connection, connection specifications are written in terms of left and right participants, rather than in terms of local and remote. Which participant is considered left or right is arbitrary; IPsec figures out which one it is being run on based on internal information. This permits using identical connection specifications on both ends.

Many of the parameters relate to one participant or the other; only the ones for left are listed here, but every parameter whose name begins with left has a right counterpart, whose

# Appendix G - IPSEC

description is the same but with left and right reversed.

Parameters are optional unless marked required; a parameter required for manual keying need not be included for a connection which will use only automatic keying, and vice versa.

### Conn parameters: General

The following parameters are relevant to both automatic and manual keying. Unless otherwise noted, for a connection to work, in general it is necessary for the two ends to agree exactly on the values of these parameters.

- type: The type of the connection. Currently the accepted values are: tunnel (the default) signifying a host-to-host, host-to-subnet, or subnet-tosubnet tunnel; transport, signifying host-to-host transport mode; and passthrough (supported only for manual keying), signifying that no IPsec processing should be done at all.

- left: Required. The IP address of the left participant's public-network interface. If it is the magic value %defaultroute, and interfaces=%defaultroute is used in the config setup section, left will be filled in automatically with the local address of the default-route interface (as determined at IPsec startup time). This also overrides any value supplied for leftnexthop. (Either left or right may be %defaultroute, but not both.) The magic value %any signifies an address to be filled in (by automatic keying) during negotiation; the magic value %opportunistic signifies that both left and leftnexthop are to be filled in (by automatic keying) from DNS data for left's client.

- leftsubnet: Private subnet behind the left participant, expressed as network/ netmask. If omitted, essentially assumed to be left/32, signifying that the left end of the connection goes to the left participant only.

- leftnexthop: Next-hop gateway IP address for the left participant's connection to the public network. Defaults to %direct (meaning right).

- leftupdown: What updown script to run to adjust routing and/or firewalling when the status of the connection changes.

### Conn parameters: Automatic Keying

The following parameters are relevant only to automatic keying, and are ignored in manual keying. Unless otherwise noted, for a connection to work, in general it is necessary for the two ends to agree exactly on the values of these parameters.

- auto: What operation, if any, should be done automatically at IPsec startup; currently-accepted values are add (signifying an ipsec auto --add), route (signifying that plus an

# Appendix G - IPSEC

ipsec auto --route), start (signifying that plus an ipsec auto --up), and ignore (also the default) (signifying no automatic startup operation). This parameter is ignored unless the plutoload or plutostart configuration parameter is set suitably; see the config setup discussion below.

- auth: Whether authentication should be done as part of ESP encryption, or separately using the AH protocol, acceptable values are esp (the default) and ah.

- authby: How the two security gateways should authenticate each other. Acceptable values are secret for shared secrets (the default) and rsasig for RSA digital signatures.

- leftid: How the left participant should be identified for authentication. Defaults to left. Can be an IP address or a fully-qualified domain name preceded by @ (which is used as a literal string and not resolved).

- leftsrasigkey: The left participant's public key for RSA signature authentication, in RFC 2537 format. The magic value %none means the same as not specifying a value (useful to override a default). The value %dnsondemand means the key is to be fetched from DNS at the time it is needed. The value %dnsonload means the key is to be fetched from DNS at the time the connection description is read from ipsec.conf. Currently this is treated as %none if right=%any or right=%opportunistic. The value %dns is currently treated as %dnsonload but will change to %dnsondemand in the future. The identity used for the left participant must be a specific host, not %any or another magic value. Caution: if two connection descriptions specify different public keys for the same leftid, confusion and madness will ensue.

- pfs: Whether Perfect Forward Secrecy of keys is desired on the connection's keying channel. (With PFS, penetration of the key-exchange protocol does not compromise keys negotiated earlier.) Acceptable values are yes (the default) and no.

- keylife: How long a particular instance of a connection (a set of encryption/ authentication keys for user packets) should last, from successful negotiation to expiry. Acceptable values are an integer optionally followed by s (a time in seconds) or a decimal number followed by m, h, or d (a time in minutes, hours, or days respectively) (default 8.0h, maximum 24h).

- rekey: Whether a connection should be renegotiated when it is about to expire. Acceptable values are yes (the default) and no.

- rekeymargin: How long before connection expiry or keying-channel expiry should attempts to negotiate a replacement begin. Acceptable values as for keylife (default 9m).

# Appendix G - IPSEC

- redeyfuzz: Maximum percentage by which rekeymargin should be randomly increased to randomize rekeying intervals (important for hosts with many connections). Acceptable values are an integer, which may exceed 100, followed by a %.

- keyingtries: How many attempts (an integer) should be made to negotiate a connection, or a replacement for one, before giving up (default 3). The value 0 means never give up.

- ikelifetime: How long the keying channel of a connection (buzzphrase: ISAKMP SA) should last before being renegotiated. Acceptable values as for keylife.

- compress: Whether IPComp compression of content is desired on the connection. Acceptable values are yes and no (the default).

## Conn parameters: Manual Keying

The following parameters are relevant only to manual keying, and are ignored in automatic keying. Unless otherwise noted, for a connection to work, in general it is necessary for the two ends to agree exactly on the values of these parameters. A manually-keyed connection must specify at least one of AH or ESP.

- spi or spibase: spi or spibase Spi or spibase isrequired for manual keying. the SPI number to be used for the connection. Must be of the form 0xhex, where hex is one or more hexadecimal digits. (Note: it will generally be necessary to make spi at least 0x100 to be acceptable to KLIPS, and use of SPIs in the range 0x100-0xfff is recommended.)

- eps: ESP encryption/authentication algorithm to be used for the connection, e.g. 3des-md5-96.

- espenckey: ESP encryption key.

- espauthkey: ESP authentication key.

- espreplay_window: ESP replay-window setting. An integer from 0 to 64. Relevant only if ESP authentication is being used.

- leftespspi: SPI to be used for the leftward ESP SA, overriding automatic assignment using spi or spibase. Typically a hexadecimal number beginning with 0x.

- ah: AH authentication algorithm to be used for the connection, e.g. hmac-md5-96. Default is not to use AH.

- ahkey: Required if ah is present. AH authentication key

- ahreplay_window: AH replay-window setting. An integer from 0 to 64.

# Appendix G - IPSEC

- leftahspi: SPI to be used for the leftward AH SA, overriding automatic assignment using spi or spibase. Typically a hexadecimal number beginning with 0x.

## Config Section

At present, the only config section known to the IPsec software is the one named setup, which contains information used when the software is being started. Here's an example:

```
config setup
interfaces="ipsec0=eth1 ipsec1=ppp0"
klipsdebug=none
plutodebug=all
manualstart=
plutoload="snta sntb sntc sntd"
plutostart=
```

Parameters are optional unless marked required. The currently-accepted parameter names in a config setup section are:

- interfaces: Required. Virtual and physical interfaces for IPsec to use: a single virtual= physical pair, a quoted list of pairs separated by white space, or %defaultroute, which means to find the interface d that the default route points to, and then act as if the value was ipsec0=d.

- forwardcontrol: Whether setup should turn IP forwarding on (if it's not already on) as IPsec is started, and turn it off again (if it was off) as IPsec is stopped. Acceptable values are yes and (the default) no.

- klipsdebug: How much KLIPS debugging output should be logged. An empty value, or the magic value none, means no debugging output (the default). The magic value all means full output.

- plutodebug: How much Pluto debugging output should be logged. An empty value, or the magic value none, means no debugging output (the default). The magic value all means full output.

- dumpdir: In what directory should things started by setup (notably the Pluto daemon) be allowed to dump core? The empty value (the default) means they are not allowed to.

- manualstart: Which manually-keyed connections to set up at startup (can be empty, a name, or a quoted list of names separated by white space).

# Appendix G - IPSEC

- plutoload: Which connections (by name) to load into Pluto's internal database at startup (can be empty, a name, or a quoted list of names separated by white space); see ipsec_auto for details. Default is none. If the special value %search is used, all connections with auto=add, auto=route, or auto=start are loaded.

- plutostart: Which connections (by name) to attempt to negotiate at startup (can be empty, a name, or a quoted list of names separated by white space). Any such names which do not appear in plutoload are implicitly added to it. Default is none. If the special value %search is used, all connections with auto=route or auto=start are routed, and all connections with auto=start are started.

- plutowait: Should Pluto wait for each plutostart negotiation attempt to finish before proceeding with the next? Values are yes (the default) or no.

- prepluto: Shell command to run before starting Pluto. For example, to decrypt an encrypted copy of the ipsec.secrets file. It's run in a very simple way. Complexities like I/O redirection are best hidden within a script. Any output is redirected for logging, so running interactive commands is difficult unless they use /dev/tty or equivalent for their interaction. Default is none.

- postpluto: Shell command to run after starting Pluto (e.g., to remove a decrypted copy of the ipsec.secrets file).

- fragicmp: Whether a tunnel s need to fragment a packet should be reported back with an ICMP message, in an attempt to make the sender lower his PMTU estimate. Acceptable values are yes (the default) and no.

- packetdefault: What should be done with a packet which reaches KLIPS (via a route into a virtual interface) but does not match any eroute. Acceptable values are pass (insecure unless you really know what you're doing!!!), drop (the default), and reject (currently same as drop).

- hidetos: Whether a tunnel packet's TOS field should be set to 0 rather than copied from the user packet inside. Acceptable values are yes (the default) and no.

- uniqueids: Whether a particular participant ID should be kept unique, with any new (automatically keyed) connection using an ID from a different IP address deemed to replace all old ones using that ID. Acceptable values are yes and no (the default).

- overridemtu: Value that the MTU of the ipsecn interface(s) should be set to, overriding IPsec's (large) default. This parameter is needed only in special situations.

# Appendix G - IPSEC

This page has been left intentionally blank.
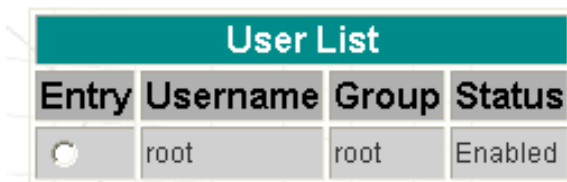
# Appendix H - Web User Management

## Introduction

In the ACS 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*.
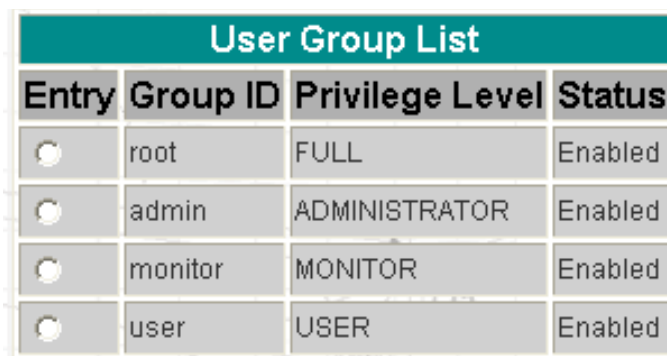
## Default Configuration for Web User Management

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*

# Appendix H - Web User Management

| 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 H - 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.

# Appendix H - Web User Management

### 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 tslinux).

**AlterPath Console Server**

# Appendix H - 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.

# Appendix H - Web User Management

**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.

**AlterPath Console Server**

# Appendix H - 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.

# Appendix H - Web User Management

**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.

AlterPath Console Server

# Appendix H - Web User Management

**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.**

# Appendix H - Web User Management

This page has been left intentionally blank.

# Appendix I - 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 41: 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 I - 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 I - 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.

# Appendix I - Connect to Serial Ports from Web
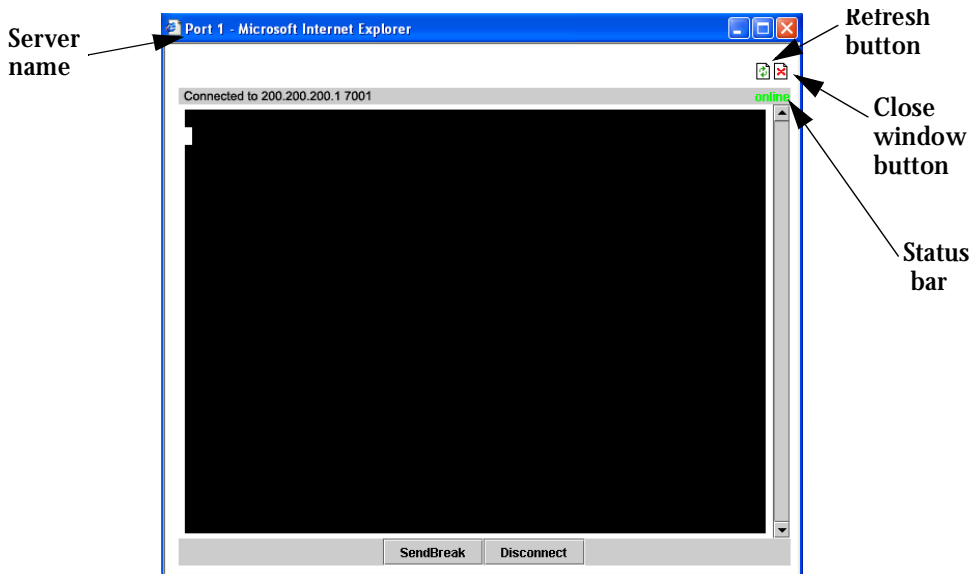


*Figure 54: Port Connection page*

At the upper right hand corner of the window, the left icon is a refresh button. Clicking on that button will reconnect to the server.



*Figure 55: The Refresh button*

The right icon closes the window. At the upper left corner, the server name is shown. In this case, the user didn't configure the serverfarm name, so "Port 1" appears.

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.

Enter in the username and the username's password if the servers were configured for authentication. If no authentication is configured, then just click Cancel.


SSH User Authentication
SSH Authorization required

User name root
Password **************

Cancel
Login
Java Applet Window

*Figure 56: 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.

# Appendix I - Connect to Serial Ports from Web

This page has been left intentionally blank.

# Appendix J - Power Management

## Introduction

The AlterPath PM is a family of intelligent power strips (IPDU - Integrated Power Distribution Units), which is used for power management. Through a serial port, the administrator can use the AlterPath PM to control all the equipment connected to its outlets, using operations like On, Off, Cycle, Lock, and Unlock.

Using the AlterPath PM and the AlterPath Console Server together, the administrator can have full control over his data center equipment. He can, for example, reboot the data center equipment when it crashes, without leaving his console session (telnet or ssh). To do that, he must simply press a configurable hotkey and select the appropriate option from the menu displayed in the session.

## Configuration

This section covers only the software configuration for the Console Server when used in conjunction with the AlterPath PM. For hardware and cabling installation instructions for the AlterPath PM, Please refer to the AlterPath PM User Guide included in the product.
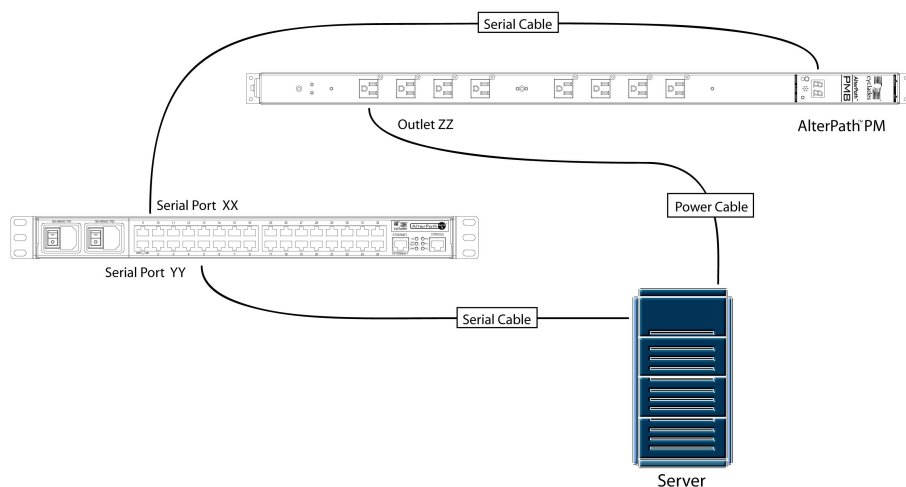


*Figure 57: Configuration diagram*

# Appendix J - Power Management

[Figure 57: Configuration diagram](#) shows a typical setup for the AlterPath PM and the Cyclades-TS/AlterPath ACS. The AlterPath PM's serial console is connected to port YY of the Console Server, the server's serial console is connected to port YY of the Console Server, and the server's power plug is connected to power outlet ZZ on the AlterPath PM. These port denominations will be used in the descriptions below.

## Port Slave Parameters Involved and Passed Values

There are two different types of parameters:

1. Parameters to the port XX where the AlterPath PM is connected:

- sXX.protocol IPDU: New protocol Integrated Power Distribution Unit. For example: ipdu.

- sXX.pmtype: The IPDU manufacturer. For example: cyclades.

- 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>]...
  ```

  The list of users 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.pmNumOfOutlets: the number of outlets of the AlterPath PM. Default: 8.

2. Parameters to the other ports where the servers are connected:

- all.protocol: Protocols for the CAS profile. For example: socket_server, socket_raw, socket_ssh.

- all.pmkey: The hot-key that starts a power management session. Default: ^p (Ctrl-p).

- sYY.pmoutlet: The outlet list where the server YY is plugged. The outlet is passed as a pair /PM_serial_port.outlet_number/. If the server has a dual power supply, the outlets are separated by space char. For example, one power supply is plugged in the second

**AlterPath Console Server**

# Appendix J - Power Management

outlet of the IPDU connected in serial port 1. The other is plugged in the third outlet of the IPDU connected in serial port 5. The value is 1.2 5.3".
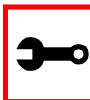
> **sXXpmusers notes:** The ellipses in the field format for sXX.pmusers means that you can add as many users as you need. The [] indicates that the parameter is optional, again indicating that you can configure more than one user. The separator is the semicolon, and spacing between the parameter and the variable matters in that a blank between names does not work.
>
> e.g. jane:1,2; john:3,4  does not work
>      jane:1,2;john:3,4   works.
>
> The users described in this parameter (sXX.pmusers) are related to the users logged in a console session. These users will not be able to do power management from any other means, unless they are root users.

## How to change the IPDU Password

**Step 1:** Change password using *pm* or *pmCommand*.

**Step 2:** Save the configuration in the IPDU.

**Step 3:** Edit the appropriate */etc/pm.\** config file.

**Step 4:** Restart pmd to re-read the config file.

## vi Method

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

## Browser Method

To configure Power Management to control IPDUs through the Web interface:

# Appendix J - Power Management

**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 the serial port where the IPDU is connected.**

After selecting the port, click the Connect button.

**Step 5: Click the PM8 profile button in the Wizard Section.**

This will automatically set the protocol to IPDU, the IPDU type to cyclades, and the number of outlets to 8.

**Step 6: Scroll down to the IPDU Section.**

Change the number of outlets and the user permissions in this section, if necessary.

**Step 7: Click the Submit button.**

If there are more IPDUs to be configured, repeat steps 4 to 7.

**Step 8: Select the port whose server has the power supply plugged into one or more IPDU outlets.**

After selecting the port, click the Submit button.

**Step 9: Configure the port as a Console Access Server.**

Read the Access Method section in Chapter 3 for details.

**Step 10: Scroll down to the Power Management Section**

Set the hotkey to access the power management menu and the outlet(s) the server is plugged into.

# Appendix J - Power Management

**Step 11: Click on the Submit button.**

**Step 12: If there are more servers to be configured, repeat steps 8 to 11.**

**Step 13: 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 14: Click on the link Administration > Load/Save Configuration.**

**Step 15: 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 Power Management custom wizard:

```
wiz --pm
```

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.
```

# Appendix J - Power Management

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 : ipdu
all.pmtype : cyclades
all.pmusers : #
all.pmoutlet : #
all.pmkey : ^p
all.pmNumOfOutlets : 8

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.PROTOCOL - The possible protocols are telnet (socket_
server), ssh1/ssh2 (socket_ssh), raw data (raw_data),
or integrated power distributed unit (ipdu).

all.protocol[ipdu] :


ALL.PMTYPE - Name of the IPDU manufacturer.

all.pmtype[cyclades] :
```

# Appendix J - Power Management

*Screen 4:*
```
*********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
*********************************************************

ALL.PMUSERS - List of the outlets each user can access.
(e.g. Joe: 1-3; Jane: 4,5,6)

all.pmusers[#] :


ALL.PMOUTLET - The number of the outlet where the server
is plugged.

all.pmoutlet[#] :
```

*Screen 5:*
```
*********************************************************
********* C O N F I G U R A T I O N   W I Z A R D *********
*********************************************************

ALL.PMKEY - The hot-key that identifies the power
management command.

all.pmkey[^p] :


ALL.PMNUMOFOUTLETS - The number of outlets you have on the
AlterPath PM.

all.pmNumOfOutlets[8] :
```

# Appendix J - Power Management

*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.protocol : ipdu
all.pmtype : cyclades
all.pmusers : #
all.pmoutlet : #
all.pmkey : ^p
all.pmNumOfOutlets : 8


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

## How to Access the AlterPath PM regular menu from the Console Session

Step 1:  **Open a console session.**
**Open a telnet or ssh session for the serial port.**

Step 2:  **Access the IPDU regular menu.**
**This should be done, for example, when the server crashes and it necessary to change the power status. Type the preconfigured hot-key.**

**If the user does not have permission to access any outlet, the following message will appear, and you will return to the Console Session:**

```
It was impossible to start a Power Management Session
You can't access any Power Management functionality.
Please contact your Console Server Administrator.
```

**If the user does not have permission to access the outlet(s) of this server, but can access another outlet, the following message will appear:**

# Appendix J - Power Management

```
You cannot manage the outlet(s) of this server.
Please enter the outlet(s) (or 'h' for help):
```

The user should type the outlet(s) he wants to manage, before reaching the main menu. The main menu will appear only if the user has permission for this/these outlet(s). Typing 'h' will cause the session to show text explaining what to type, and 'l' will cause the PM session to be logged out, and the user to return to the Console Session. If the user has permission to access the outlet(s) of this server, these outlets will be managed by the PM session.

**Step 3: Regular Menu.**

This is the AlterPath PM regular menu:

```
-------------------------------------------------------
Cyclades Corporation - Power Management Utility
-------------------------------------------------------
1 - Exit    2 - On     3 - Off
4 - Cycle   5 - Lock   6 - Unlock
7 - Status  8 - Help   9 - Other

Please choose an option:
```

**Table 42: AlterPath PM Regular Menu Options**

| Option | Description |
|--------|-------------|
| Exit | Exits the Power Management Session. |
| On | Turns the outlet on. |
| Off | Turns the outlet off. |
| Cycle | Turns the outlet off and back on. |
| Lock | Locks the current status of the outlet. |
| Unlock | Unlocks the current status of the outlet. |
| Status | Shows the current status of the outlet. |
| Other | Allows user to control other outlets. |

# Appendix J - Power Management

**Step 4: Check the status of the server's outlet or the outlet list.**

**Type '7' and wait for the answer. For example:**

```
Please choose an option: 7

IPDU 1 Outlet 8:
Outlet Status User
8 OFF NONE
--------------------------------------------------------
Cyclades Corporation - Power Management Utility
--------------------------------------------------------
1 - Exit       2 - On      3 - Off
4 - Cycle      5 - Lock    6 - Unlock
7 - Status     8 - Help    9 - Other

Please choose an option:
```

**Step 5: Reboot the server.**

**If the outlet(s) is/are locked, the user must unlock the outlet(s) first (option 6 - Unlock). The Cycle command turns the power off for some seconds and the turn it on again. Type '4' and wait for the answer. For example:**

```
Please choose an option: 4

IPDU 1 Outlet 8:
8: Outlet power cycled.

--------------------------------------------------------
Cyclades Corporation - Power Management Utility
--------------------------------------------------------
1 - Exit       2 - On      3 - Off
4 - Cycle      5 - Lock    6 - Unlock
7 - Status     8 - Help    9 - Other

Please choose an option:
```

# Appendix J - Power Management

**Step 6:  Change the outlet list.**

If the user needs to access another outlet(s) which can be managed by him, the option 9 - Other should be used. For example:

```
--------------------------------------------------------
Cyclades Corporation - Power Management Utility
--------------------------------------------------------
1 - Exit    2 - On     3 - Off
4 - Cycle   5 - Lock   6 - Unlock
7 - Status  8 - Help   9 - Other

Please choose an option: 9
Please enter the outlet(s) (or 'h' for help): 1.2

--------------------------------------------------------
Cyclades Corporation - Power Management Utility
--------------------------------------------------------
1 - Exit    2 - On     3 - Off
4 - Cycle   5 - Lock   6 - Unlock
7 - Status  8 - Help   9 - Other

Please choose an option:
```

From this point, all the commands will be related to the 2nd outlet of the IPDU in the port 1.


**Step 7:  Return to the Console Session.**

The user can exit from the PM session and return to the Console Session in three ways:

1. Type the hot-key again, any time.

2. If the session is waiting for a menu option, type the option 1 - Exit.

3. If the session is waiting for the outlet, type 'l'.

When the user leaves the PM session, the following message will appear:

```
Exit from PM session
```

# Appendix J - Power Management

## Power Management for the Administrator

The administrator, who can log onto the Console server itself, must have total control over all the IPDU outlets. An additional menu, with more options than the regular menu, is provided for the administrator to manage any IPDU.

There are two commands which can be used to manage the IPDU. The first one (*pm*) deals with menu options, while the second one (*pmCommand*) deals with the commands as they are sent to the IPDU, and requires more knowledge about the AlterPath-PM commands.

> **Note:** Only the root user can do power management by using pm or pmCommand.

### pm command

There are two ways to use this command: menu interface or command line. The menu is reached by typing the following command, from the prompt:

```
pm <IPDU port>
```

For example:

```
-------------------------------------------------------------------
         Cyclades Corporation - Power Management Utility
-------------------------------------------------------------------
       1 - Exit              2 - Help              3 - On
       4 - Off               5 - Cycle             6 - Lock
       7 - Unlock            8 - Status            9 - Password
      10 - Alarm            11 - Syslog           12 - Buzzer
      13 - Current          14 - Save             15 - Version



Please choose an option: 2

Exit              - Exits the Power Management session
```

# Appendix J - Power Management

```
Help                - Shows this message
On                  - Turn outlet(s) ON
Off                 - Turn outlet(s) OFF
Cycle               - Turn outlet(s) OFF and back ON
Lock                - Lock the current status of outlet(s)
Unlock              - Unlock the current status of outlet(s)
Status              - Show the current status of outlet(s)
Password            - Set a password for the specific user
Alarm               - Set alarm threshold for current
Syslog              - Turn syslog on or off
Buzzer              - Turn buzzer on or off
Current             - Show current consumption for the entire unit
Save                - Save configuration and status
Version             - Displays version information
```

Some of these options require the outlet number (On, Off, Cycle, Lock, Unlock, Status), and others don't. In the first case, when the option is selected, the number of the outlet will be asked. The user can enter one or more outlets (separated by commas or dashes), or "all," to apply the option to all the outlets.

Following are examples of some things which can be done through this command.

# Appendix J - Power Management

### Turning the outlet off

```
Please choose an option: 3
Please enter the outlets (or 'help' for help): 4

4: Outlet turned off.


--------------------------------------------------------
Cyclades Corporation - Power Management Utility
--------------------------------------------------------
1 - Exit        2 - On      3 - Off
4 - Cycle       5 - Lock    6 - Unlock
7 - Status      8 - Help    9 - List
10 - Current

Please choose an option:
```

### Locking the outlets

**When the outlet is locked, the previous status cannot be changed, until the outlet is unlocked. This means that if the outlet was on, it cannot be turned off and, if it was off, it cannot be turned on.**

```
Please choose an option: 5
Please enter the outlets (or 'help' for help): 1-3

1: Outlet locked.
2: Outlet locked.
3: Outlet locked.


--------------------------------------------------------
Cyclades Corporation - Power Management Utility
--------------------------------------------------------
1 - Exit    2 - On       3 - Off
4 - Cycle   5 - Lock    6 - Unlock
7 - Status  8 - Help     9 - List
10 - Current

Please choose an option:
```

# Appendix J - Power Management

## Retrieving the status of the outlets

```
Please choose an option: 7
Please enter the outlets (or 'help' for help): all

Outlet Status User
1 Locked OFF NONE
2 Locked OFF NONE
3 Locked OFF NONE
4 OFF NONE
5 OFF NONE
6 OFF NONE
7 OFF NONE
8 OFF NONE


------------------------------------------------------
Cyclades Corporation - Power Management Utility
------------------------------------------------------
1 - Exit        2 - On      3 - Off
4 - Cycle       5 - Lock    6 - Unlock
7 - Status      8 - Help    9 - List
10 - Current

Please choose an option:
```

**The second way to use the pm command is the command line. In this case, the syntax for the command is**

```
pm <IPDU port> <option> [<outlet(s)>],
```

**where:**

**<option> is the name of the option, as written in the menu.**
**<outlet(s)> is the number of the outlet(s) the option will be applied to, and is used only if the option requires the outlet.**

**In the examples above, the same result can the achieved by using the command line mode:**

# Appendix J - Power Management

### Turning the outlet off

```
[root@TSx000 /root]# pm 1 Off 4

4: Outlet turned off.
[root@TSx000 /root]#
```

### Locking the outlets

```
[root@TSx000 /root]# pm 1 Lock 1-3

1: Outlet locked.
2: Outlet locked.
3: Outlet locked.
[root@TSx000 /root]#
```

### Retrieving the status of the outlets

```
[root@TSx000 /root]# pm 1 Status all

Outlet Status User
1 Locked OFF NONE
2 Locked OFF NONE
3 Locked OFF NONE
4 OFF NONE
5 OFF NONE
6 OFF NONE
7 OFF NONE
8 OFF NONE
[root@TSx000 /root]#
```

## pmCommand command

**Through pmCommand command, the administrator has access to other options beyond the menu options, because he will be accessing the IPDU itself. The administrator must have a good knowledge of the AlterPath PM command set to use it.**

# Appendix J - Power Management

There are two ways to use this command. If only the IPDU port is passed as an argument, it will appear in a prompt where the administrator can write the command. Otherwise, the arguments after the IPDU port will be considered the PM command.

**Syntax:**

```
pmCommand <IPDU port> [<command>]
```

**For example:**
```
[root@CAS root]# pmCommand 1
You're entering the "Power Management Prompt".
To go back to the Console Server's command line type: exitPm

[Cyclades - Power Management Prompt]#
```

**The following are examples of some things which can be done through this command.**

## Listing the commands available for the AlterPath PM
```
[Cyclades - Power Management Prompt]# help

on <outlet><cr>          --Turn <outlet> ON
off <outlet><cr>         --Turn <outlet> OFF
cycle <outlet><cr>       --Turn <outlet> OFF and back ON
lock <outlet><cr>        --Lock the current status of <outlet>
unlock <outlet><cr>      --Unlock the current status of <outlet>
status <outlet><cr>      --Show the current status of <outlet>
list<cr>                 --List users created and eventual outlets
                           assigned
exit<cr>                 --Exit session
passwd <user><cr>        --Set a password for the specific user
help<cr>                 --Show supported commands
current<cr>              --Show the instantaneous current consump
                           tion for the entire unit
adduser <username><cr>   --Add user to the DB (8 maximum users
                           allowed)
deluser <username><cr>   --Delete user from the DB
assign <outlet> <username><cr> --Assign <outlet> to a specific
                                 user
name <outlet> <name><cr> --Name an outlet
[Cyclades - Power Management Prompt]#
```

# Appendix J - Power Management

### Cycling all the outlets

```
[Cyclades - Power Management Prompt]# cycle all

1: Outlet power cycled.
2: Outlet power cycled.
3: Outlet power cycled.
4: Outlet power cycled.
5: Outlet power cycled.
6: Outlet power cycled.
7: Outlet power cycled.
8: Outlet power cycled.
[Cyclades - Power Management Prompt]#
```

### Unlocking the outlets 1, 5 and 8

```
[Cyclades - Power Management Prompt]# unlock 1, 5, 8

1: Outlet unlocked.
5: Outlet unlocked.
8: Outlet unlocked.
[Cyclades - Power Management Prompt]#
```

**AlterPath Console Server**

# Appendix J - Power Management

## Power Management from a Browser

The Console Server Web server also supports power management. From a Web browser it is possible to check the status of all the IPDUs connected to the Console Server, as well as their outlets. If the user has Administration privileges, he can also perform the commands to turn on, turn off, cycle, lock and unlock the outlets.

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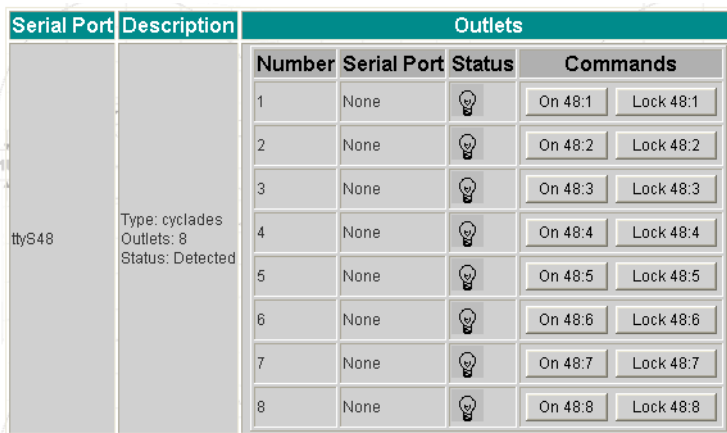
**Step 2:  Log in as root and type the Web root password configured by the Web server.**

This will take you to the Configuration and Administration page.

**Step 3:  Select the Power Management link.**

This link is in the Administration section. The following page will appear:

| Serial Port | Description | Outlets | | | |
|---|---|---|---|---|---|
| | | Number | Serial Port | Status | Commands |
| ttyS48 | Type: cyclades Outlets: 8 Status: Detected | 1 | None | 💡 | On 48:1   Lock 48:1 |
| | | 2 | None | 💡 | On 48:2   Lock 48:2 |
| | | 3 | None | 💡 | On 48:3   Lock 48:3 |
| | | 4 | None | 💡 | On 48:4   Lock 48:4 |
| | | 5 | None | 💡 | On 48:5   Lock 48:5 |
| | | 6 | None | 💡 | On 48:6   Lock 48:6 |
| | | 7 | None | 💡 | On 48:7   Lock 48:7 |
| | | 8 | None | 💡 | On 48:8   Lock 48:8 |

*Figure 58:  Power Management page*

# Appendix J - Power Management

In the figure above, all the outlets are off (the light bulbs are off) and unlocked. For this status, there are two possible commands: turn it on and lock it.

The following steps are examples of what can be done in this page.

**Step 4: Turn the outlet 1 on.**

Click on the "On 48:1" button. The following page will appear:

| Serial Port | Description | Outlets | | | |
|---|---|---|---|---|---|
| | | Number | Serial Port | Status | Commands |
| ttyS48 | Type: cyclades Outlets: 8 Status: Detected | 1 | None | 💡 | Off 48:1  Cycle 48:1  Lock 48:1 |
| | | 2 | None | 💡 | On 48:2  Lock 48:2 |
| | | 3 | None | 💡 | On 48:3  Lock 48:3 |
| | | 4 | None | 💡 | On 48:4  Lock 48:4 |
| | | 5 | None | 💡 | On 48:5  Lock 48:5 |
| | | 6 | None | 💡 | On 48:6  Lock 48:6 |
| | | 7 | None | 💡 | On 48:7  Lock 48:7 |
| | | 8 | None | 💡 | On 48:8  Lock 48:8 |

*Figure 59:  Power Management page after turning outlet 1 on*

After this operation, the outlet 1 was turned on (the light bulb is on), and now the administrator can turn it off and cycle this outlet.

**Step 5: Lock outlet 1.**

Click on the "Lock 48:1" button. The following page will appear:

# Appendix J - Power Management

| Serial Port | Description | Outlets | | | |
|---|---|---|---|---|---|
| | | Number | Serial Port | Status | Commands |
| ttyS48 | Type: cyclades Outlets: 8 Status: Detected | 1 | None | 💡🔒 | Off 48:1 \| Cycle 48:1 \| Unlock 48:1 |
| | | 2 | None | 💡 | On 48:2 \| Lock 48:2 |
| | | 3 | None | 💡 | On 48:3 \| Lock 48:3 |
| | | 4 | None | 💡 | On 48:4 \| Lock 48:4 |
| | | 5 | None | 💡 | On 48:5 \| Lock 48:5 |
| | | 6 | None | 💡 | On 48:6 \| Lock 48:6 |
| | | 7 | None | 💡 | On 48:7 \| Lock 48:7 |
| | | 8 | None | 💡 | On 48:8 \| Lock 48:8 |

*Figure 60:  Power Management page after locking outlet 1*

**The padlock indicates that outlet 1 was locked. From this point, nothing can change the outlet status, until the outlet is unlocked.**

# Appendix J - Power Management

This page has been left intentionally blank.

# Appendix K - Examples for Config Testing

## Introduction

The following three examples are just given to *test* a configuration. The steps should be followed *after* configuring the AlterPath Console Server.

## Console Access Server

With the AlterPath Console Server set up as a CAS you can access a server connected to the ACS 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 the following figure.Figure 61: Console Access Server diagram. This configuration example has local authentication, an Ethernet interface provided by a router, and serially-connected workstations.

# Appendix K - Examples for Config Testing



ACS16 Ethernet Interface
IP Address: 200.200.200.1

Authentication

Ethernet Interface
200.200.200.5

LAN

AlterPath ACS16

Router

Internet

Socket Port
7002
192.168.1.102

Socket Port
7008
192.168.1.108

Serial
Connections
Speed: 9.6K

Workstation

Socket Port
7001
192.168.1.101
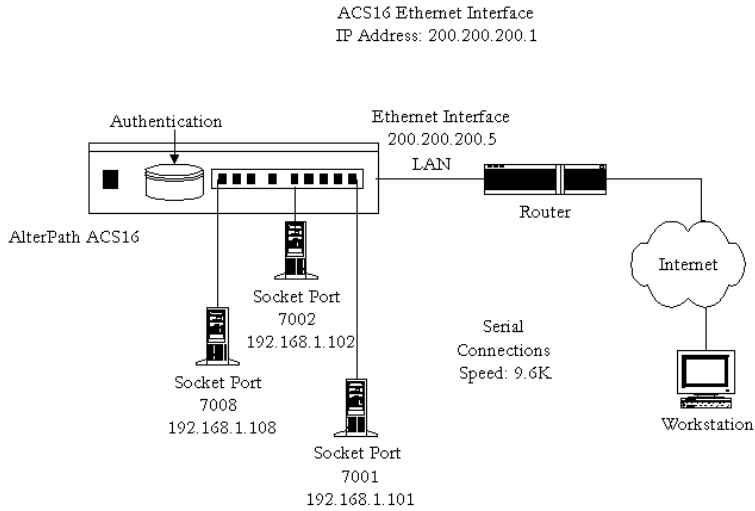
*Figure 61:  Console Access Server diagram*

The following diagram<u>Figure 62: CAS diagram with various authentication methods</u>, shows additional scenarios for the AlterPath Console Server: both remote and local authentication, data buffering, and remote access.

# Appendix K - Examples for Config Testing
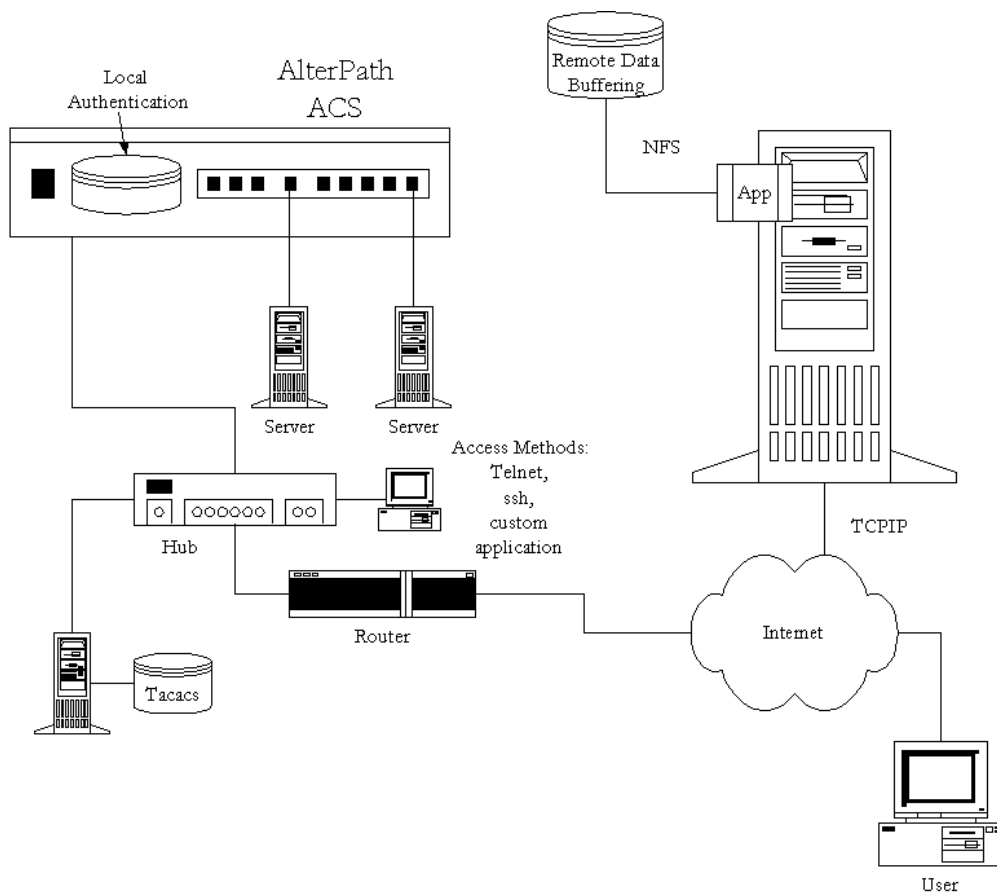


*Figure 62: CAS diagram with various authentication methods*

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 K - 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 AlterPath Console Server 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 ACS.**

The AlterPath Console Server 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 AlterPath Console Server 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 AlterPath Console Server 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.

AlterPath Console Server

# Appendix K - Examples for Config Testing

**Step 6: Activate the changes.**

Now continue on to [Task 5: Activate the changes](#) through   listed in [Chapter 2 - Installation, Configuration, and Usage](#).
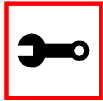
> Note: It is possible to access the serial ports from Microsoft stations using some off-the-shelf packages. Although Cyclades is not liable for those packages, successful tests were done using at least one of them. From the application's viewpoint running on a Microsoft station, the remote serial port works like a regular COM port. All the I/O with the serial device attached to the AlterPath Console Server is done through socket connections opened by these packages and a COM port is emulated to the application.

## Terminal Server

The AlterPath Console Server 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. The terminal can be either a dumb terminal or a terminal emulation program on a PC.
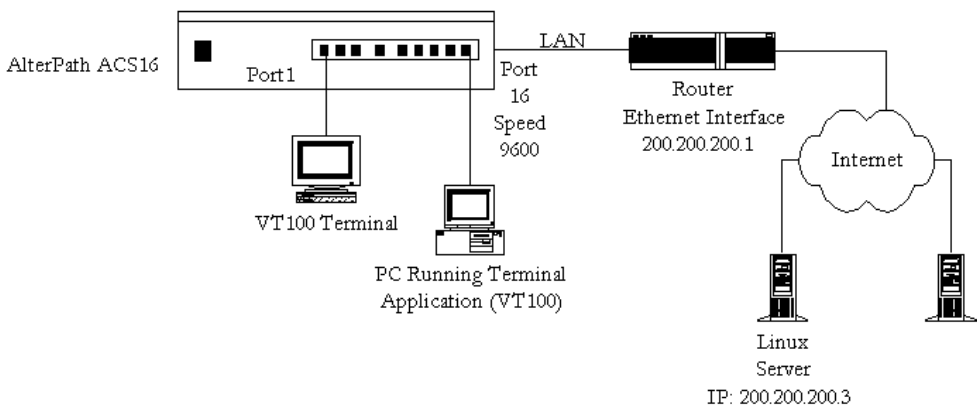


*Figure 63: Terminal Server diagram*

# Appendix K - Examples for Config Testing

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, the  AlterPath Console Server 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 AlterPath Console Server 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 ACS.**

The AlterPath Console Server has 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 AlterPath Console Server,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  listed in Chapter 2 - Installation, Configuration, and Usage.

AlterPath Console Server

# Appendix K - Examples for Config Testing

## Dial-in Access

The AlterPath Console Server can be configured to accommodate out-of-band management. Ports can be configured on the AlterPath Console Server 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. Cyclades recommends that a maximum of two ports be configured for this option.
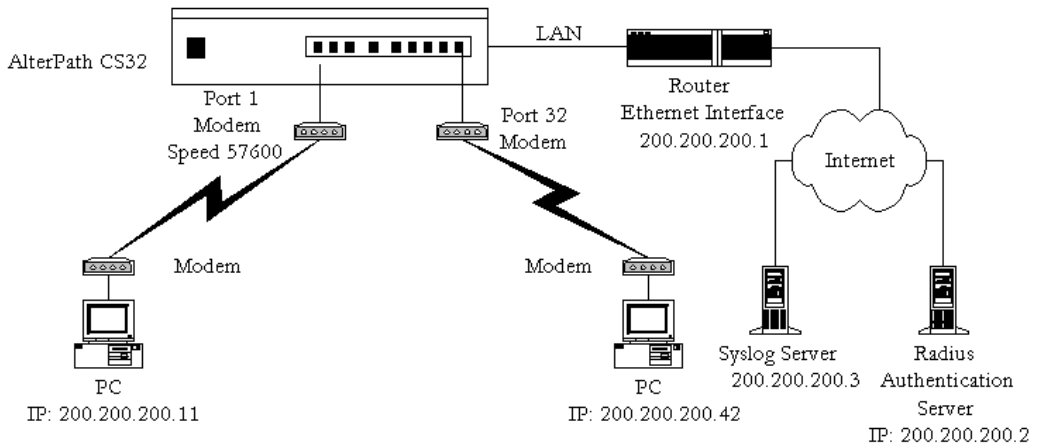


*Figure 64:  Ports configured for Dial-in Access*

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*.

# Appendix K - Examples for Config Testing

**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 AlterPath Console Server 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 AlterPath Console Server has 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 AlterPath Console Server 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 AlterPath Console Server 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 AlterPath Console Server listed in Chapter 2 - Installation, Configuration, and Usage.

AlterPath Console Server

# Appendix L - 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**
- **Users**
- **Poll_interval**
- **Tx_interval**
- **Idletimeout**
- **Conf.group**
- **<sN>.serverfarm**
- **pool_ipno**
- **pool_socket_port**
- **pool_serverfarm**

# Appendix L - Wiz Application Parameters

- **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**

# Appendix L - Wiz Application Parameters

- **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 L - 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**

**AlterPath Console Server**

# Appendix L - 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 L - Wiz Application Parameters

## Terminal Server Profile Other Parameters (wiz --tso)

- **Host**

- **Term**

- **Conf.locallogins**

**AlterPath Console Server**

# Appendix M - Copyrights

## References

The AlterPath Console Server 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 AlterPath Console Server and a reference to their maintainers. The copyrights notices required in some packets are placed in the /COPYRIGHTS directory of the AlterPath Console Server image.

### Bash

Bourne Again Shell version 2.0.5a. Extracted from the HardHat Linux distribution.
http://www.gnu.org/software/bash

### 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

# Appendix M - Copyrights

## 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**

## GNU

**The GNU project**
**http://www.gnu.org**

## HardHat Linux

**MontaVista Software - HardHat version 2.1**
**http://www.montavista.com**

## IPSec

**The Linux FreeS/WAN IPsec version 1.9.8**
**http://www.freeswan.org**
**COPYRIGHT: This product includes software developed by Eric Young (eay@cryptsoft.com)**

## IPtables

**Netfilter IPtables version 1.2.2. Extracted from the HardHat Linux distribution.**
**http://www.netfilter.org**

## Linux Kernel

**Linux Kernel version 2.4.18. Extracted from the HardHat Linux distribution**
**http://www.kernel.org**

## Net-SNMP

**SourceForge Net-SNMP project version 5.0.3**
**http://sourceforge.net/projects/net-snmp/**

# Appendix M - Copyrights

## 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)**

## 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/**

# Appendix M - Copyrights

## Tinylogin

**TinyLogin version 0.80**
**ftp://ftp.lineo.com/pub/tinylogin/**

## UCD-SNMP

**SourceForge Net-SNMP project version 4.2.4.pre1**
**http://sourceforge.net/projects/net-snmp/**

## 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/**

AlterPath Console Server

# Appendix N - AlterPath Bio

## Introduction

AlterPath Bio is a Cyclades solution that integrates biometric authentication while increasing security on the AlterPath Console Server. The AlterPath Bio device works in conjunction with public SSH keys and the AlterPath Console Server to require fingerprint authentication for Web-based users in order for them to remotely access the console ports on the AlterPath Console Server.

In order for the biometric device to integrate with the ACS' Web-based interface, there is an SSH Java applet that starts the utility which then reads the user's fingerprint and unlocks the AlterPath Bio device. Once unlocked, the Java applet can access the private key stored on the AlterPath Bio's memory and allow access to the requested console port. This Quick Start guide gives basic installation instructions.

To use the biometric authentication of AlterPath Bio with the AlterPath Bio system you need:

* Windows 2000 Pro (with Service Pack 4 or later), or

* Windows XP (with Service Pack 1 or later)

on a computer that has an available USB port, and

* Java Runtime version 1.4.2 or later with the Java plug-in configured for your browser of choice.

### Important Features

In order to take full advantage of AlterPath Bio, it is important to understand some of the features and limitations of the ClipDrive Bio software. These issues are outlined below.

1. When a fingerprint and an optional non-administrator password are given to the Clip-Drive Bio unlock utility to unlock the device, the utility will not allow user accounts to be changed.

2. When a fingerprint and an administrator password are given, the utility will expose the *User Accounts* button to manage accounts on the device.

3. Once unlocked, the device cannot be locked unless it is removed from the system. To safely remove a device from Windows, use the *Safely Remove Hardware* feature of Windows. (The Safely Remove Hardware feature usually displays an icon that appears in the taskbar once a removable device is connected to the system.)

# Appendix N - AlterPath Bio

4. Drive E, named *CLIP DRIVE* is always present when the ClipDrive Bio device is connected to the USB port. Drive F*, named *BIO SECURE* only appears after the device is unlocked. (Depending on your windows configuration, it may or may not be Drive E and F.)

## What's in the Box

Your package should contain the following items:

• **AlterPath Bio Software and Documentation CD**

• **This Quick Start Guide**

• **AlterPath Bio Device**

• **USB Cable**

• **Leather Carrying Case**
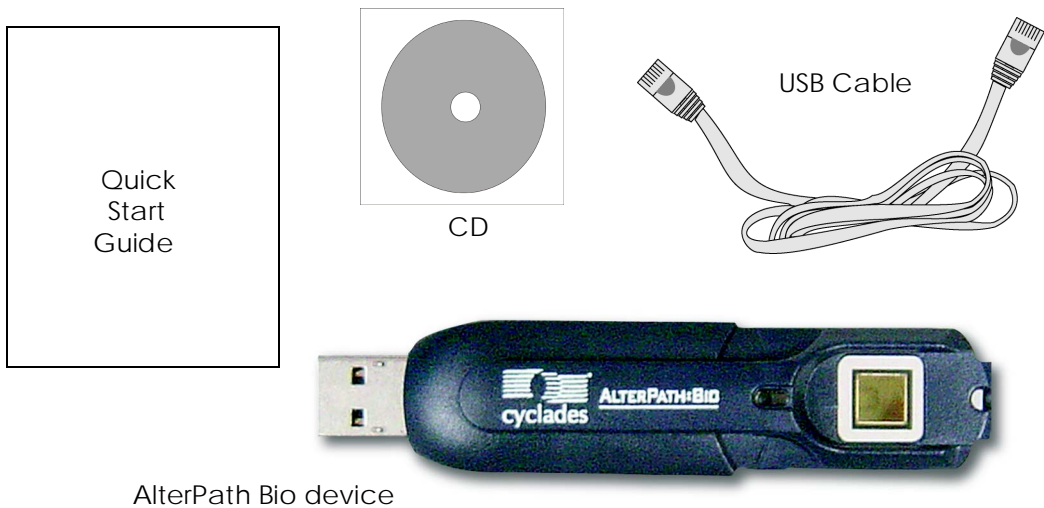


*Figure 65: The AlterPath Bio, its cables, connectors and other box contents*

# Appendix N - AlterPath Bio

## Software Installation and Setup

### How to Install the AlterPath Bio Software

Follow these steps to install the AlterPath Bio software on your computer:

**Step 1:** Insert the ClipDrive Bio CD.

**Step 2:** Double-click the setup icon.

**Step 3:** Follow the installer instructions.

**Step 4:** Enroll in ClipGuard feature (optional).

When the installer is nearing completion, it will start the *ClipGuard Bio Enrollment Wizard*. This is for the ClipGuard feature only and is not necessary for the use of AlterPath Bio. You may want to complete this enrollment anyway.

**Step A:** Type the user name.
The user name does not necessarily have to match any user name on the ACS or the remote server.

**Step B:** Select finger whose fingerprint you would like to enroll. Click *Next*.

**Step C:** Press/remove finger to/from the biometric sensor as prompted.

**Step 5:** Upon computer restart, run the *Unlock ClipDrive Bio* utility.

**Step 6:** Prepare to enroll new users.

On a new device, *Unlock ClipDrive Bio* will display a message "Fingerprint Database is empty. Please, select 'User Accounts' to enroll new users." Click *OK*, then click *User Accounts*.

**Step 7:** Decide whether you would like a fingerprint and password combination.

On the *ClipDrive Bio Administration screen*, decide whether you would like both a fingerprint and a password to unlock the ClipDrive Bio, or just a fingerprint. Make the appropriate selection.

# Appuidix N - AlterPath Bio

**Step 8:  Click** *New User*:

**Step 9:  Enroll new user.**

> Enter a username into the *User ID* field. Select finger to enroll. Click *OK*. Note that the user does not have to exist on the AlterPath Console Server or remote server.

**Step 10:  Press/remove finger to/from the biometric sensor as prompted.**

**Step 11:  Set up administrator password.**

> Once enrolled, it is a good idea to set up an administrator password for managing enrolled users. To do this, click *Set Administrator Password*, enter the desired password twice, and click *OK.*

---
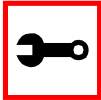
**Note:** Use ClipDrive Bio software version 3.5 or greater. Additionally, the driver needs to be placed in the \Program Files\BXI\ClipDrive Bio directory. Do not install the software in any other directory.

---

## How to Generate and Install SSH Keys

AlterPath Bio is able to provide increased security through fingerprint authentication enhanced by SSH public key authentication. Public key authentication works by verifying a key pair that includes a public key (stored on the ACS) and a private key (stored in the AlterPath Bio). If the public key verifies successfully against its corresponding private key (usually stored in a safe location on a file system), then the user is allowed to log in. In the case of the AlterPath Bio, the private key is stored on the AlterPath Bio device itself.

### How to Generate the SSH Keys

Technically speaking, the SSH key generation can be done on any SSH-compatible software package that offers this capability. This Quick Start guide will use the AlterPath Console Server to generate the keys so that no additional software package is required.

AlterPath Console Server

# Appendix N - AlterPath Bio

**Step 1:** **Log into the ACS.**

Log into the ACS using the same user that you will use to log in for Biometric Authentication. For the purpose of this example, we will use the *root* user. Log in using any method that is convenient: ssh, telnet, console port, etc.

**Step 2:** **At the UNIX command prompt**, execute the following command:

```
ssh-keygen -t dsa
```

Follow the instructions on the screen. Accept the default location for the key by pressing *Enter*. Make sure that the passphrase you enter has more than four characters in it or *ssh-keygen* will not generate the public/private key pair. Your output should be similar to the following:

```
[root@CAS root]$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
af:83:ff:1a:21:57:d3:5b:f8:14:55:e1:92:5b:a2:87 root@CAS
[root@CAS root]$
```

## How to Install the Private Key

To complete the SSH component of the AlterPath Bio configuration, the private key must be installed on the AlterPath Bio device. The private key is stored in the file *id_dsa* stored in the */root/.ssh* directory (or the */home/<username>/.ssh* directory if you are logged in as *<username>*). This file needs to be transferred to the Windows computer that is going to use the AlterPath Bio and stored on the *BIO SECURE* partition of the unlocked AlterPath Bio device.

In order to transfer the *id_dsa* private key file to the Windows computer, it is recommended that an SSH client with scp on the Windows computer be used. It is also possible to use FTP. If these methods are not available, a floppy disk might also be used.

# Appendix N - AlterPath Bio

PSCP Private Key transfer to AlterPath Bio

**Step 1: Open a Run window under the START menu.**

**Step 2: Run: pscp with user, network address, originating file, and destination file. For example:**

```
pscp user@11.12.13.2:/root/.ssh/id_dsa  F:
```

**Step 3: Close the Run window.**
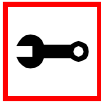
> **Note: PSCP.exe can be downloaded from:**
> **http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html.**

## How to Install the Public Key

The public key is called *id_dsa.pub* and is also located in the */root/.ssh* directory (or the */home/<username>/.ssh* directory if you are logged in as <username>) where the private key was generated. This file needs to be appended to the master public key file located in */etc/ssh/authorized_keys*. This can be done with the following command:

```
cat id_dsa.pub >> /etc/ssh/authorized_keys

saveconf
```

It may be desirable, for increased security, to remove the generated public and private keys from the user's home directory (in this case */root/.ssh*) once they have been copied to the appropriate locations.

         **AlterPath Console Server**

# Appendix N - AlterPath Bio

## How to Configure the AlterPath Console Server for Biometric Authentication

### Overview

Biometric authentication can be used either at the *console-port level* or at the *user level*. If biometric authentication is enabled for a user, then that user's account will always require it. If biometric authentication is enabled for a port, then only biometric users will be able to access that port.

### Configuration

**Step 1:  Go to Web User Management -> Users.**

Biometric authentication for a user is set in the *Web User Management -> Users* feature of the AlterPath Console Server Web interface.
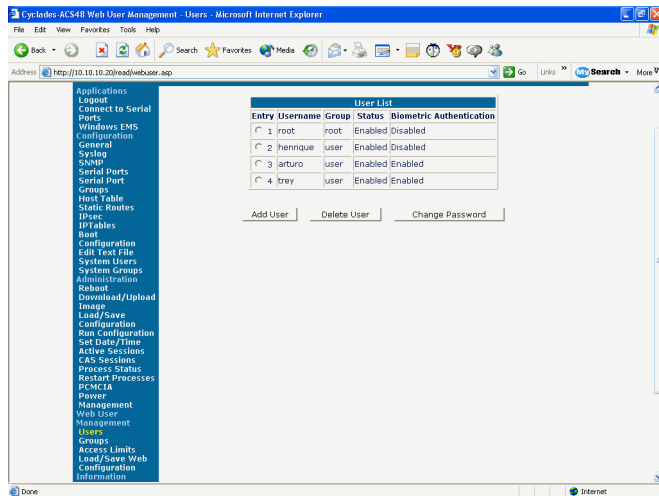


*Figure 66:  User List page*

**Step 2:  Decide whether a new user should be biometric or not.**

When creating a new user, you are given a choice to make that user a biometric user or not. (Please refer to Biometric Port Configuration in the ACS manual.)

# Appendix N - AlterPath Bio
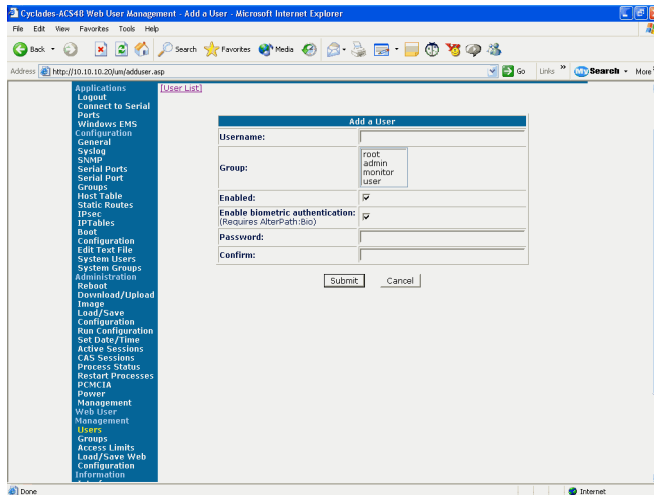


*Figure 67: Add a User page*

**Step 3: Create a system user that corresponds to the Web user.**

There also needs to be a system user that corresponds to the Web user. It is recommended that this user be created with the same username and password as the Web user.

**Step 4: Confirm that protocol is Socket SSH when configuring serial ports.**

When configuring the serial ports, there is an option for biometric authentication in the *Profile* section of the interface. This feature works only when the selected protocol is *Socket SSH*.

# Appendix N - AlterPath Bio
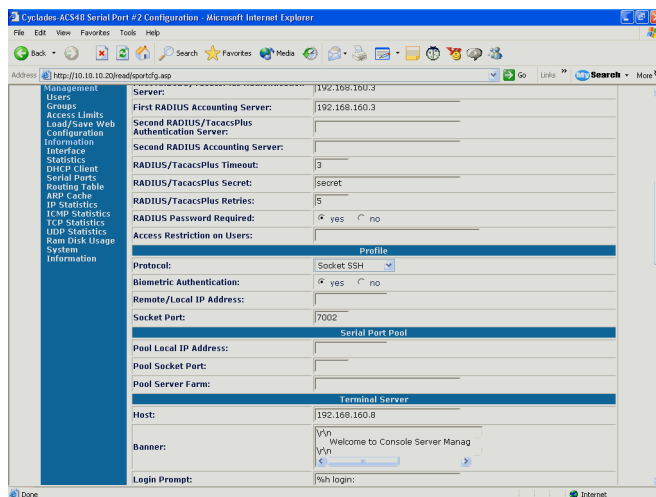


*Figure 68:  Serial Port Profile page*

**Step 5:** Confirm /etc/ssh/sshd_config file contains line.

If you have upgraded to the current ACS software (which has Biometric capability) from a previous version, you will need to make sure the following line exists in your */etc/ssh/sshd_config* configuration file:

```
AuthorizedKeysFile      /etc/ssh/authorized_keys
```

**Step 6:** Restart sshd process.

The sshd process will need to be restarted after this change is made:

```
daemon.sh<enter>
```

Daemon.sh will examine /etc/ssh/sshd_config and find the change to the file. It will then restart sshd.

**Step 7:** Select *Load/Save Configuration* from the right hand menu.

This will save AlterPath Bio settings.

# Appendix N - AlterPath Bio

## How to Use Biometric Authentication

This section will walk you through using biometric authentication to access console ports through the Web interface.

Step 1:  **Connect to the desired serial port.**

Using the *Connect to Serial Ports* feature of the Web interface, connect to the desired serial port with a biometric user.

Step 2:  **Run the indicated Java applet.**

A message will appear asking permission to run the Java applet. The applet is self-signed by Cyclades and it is safe to run. Click *Yes* to give the applet permission to run.

Step 3:  **Enter a username, password, and passphrase.**

The login prompt asks for a username, password, and passphrase. Depending on authentication method, the password will either be the local password or the remote authentication server password. The passphrase is the passphrase that was assigned to the SSH key when it was created.

Step 4:  **If login is successful, the *Unlock ClipDrive Bio* utility will start.**

This utility will prompt for a fingerprint and password.

Step 5:  **Enter the password (if required) and fingerprint.**

At this point, the *Unlock ClipDrive Bio* will disappear after informing the user of a successful authentication.

Step 6:  **Access the serial port via the Java SSH applet console.**

The Java SSH applet console will now let you into the serial port in question.

# Appendix O - Start and Stop Daemons

## Introduction

This feature allows daemons (services) to be enabled or disabled without need of reboot the TS or ACS. A simple engine detects configuration changes (file comparison). This feature is implemented with shell scripts. There is one main shell script called *daemon.sh* and one sourced shell script (included by daemon.sh) for every daemon (service) that runs in the TS or ACS. The shell script daemon.sh must be run once by inittab and every time a configuration change is made. The daemon.sh reads a file */etc/daemon_list* which contains the names of all sourced shell scripts and performs the start/stop/restart operation needed if any file related to service was changed. The daemon.sh will keep a hidden copy, prefixed with "." and suffixed with *.tmp*, of all related files in the directory */var/run.*

Each sourced shell script has a set of mandatory shell variables handled directly by the shell script daemon.sh. The sourced shell scripts may have other shell variables not handled directly by daemon.sh. Such variables have the sole purpose of facilitating the configuration of command line parameters.

The mandatory shell variables define:

1. If the service is enabled or disabled. (ENABLE=YES/NO)

2. The pathname to the daemon. (DNAME=<daemon name, DPATH=<daemon path>)

3. How to restart the daemon: by signal (kill, hup, term, etc) or by command (start, stop. etc). (DTYPE=sig/cmd)

4. Signal to be sent to the daemon. Default is *term*. (DSIG=<signal>)

5. A list of configuration files. The files in this list will be checked for changes. (ConfigFiles=<config file list>)

6. A initialization shell script that will be run before start the service. (ShellInit=<shell_script_name [command line parameters]>)

7. Command line parameters to start the daemon. (DPARM=<command line parameters>)

8. Command Line parameters to stop the daemon. (DSTOP=<command line parameters>)

The daemon.sh may be executed in two ways:

# Appendix O - Start and Stop Daemons

1.  Without parameters in the command line, it will check the configuration files of the service and restart or stop it if needed.

2.  It will perform the requested action (stop/restart) in the list of services given in the command line regardless any configuration changes.

The command *daemon.sh help* will display a list of services available. Currently the following services are handled by daemon.sh. The first column is the service ID, the second is the name of the shell script file.

```
DB        /etc/cy_buffering.sh      // Data buffering    (ACS only)

NTP       /etc/ntpclient.conf       // Network Timer client

SNMP      /etc/snmpd.conf           // SNMP

IPSEC     /etc/ipsec.sh             // VPN - Ipsec       (ACS only)

NIS       /etc/ypbind.conf          // NIS               (ACS only)

SYSLOG    /etc/syslog.sh            // Syslog daemon

SSH       /etc/sshd.sh              // Secure shell

INETD     /etc/inetd.sh             // Network services daemon

PMD       /etc/pmd.sh               // Power Management daemon
```

The following example will restart power management and Data Buffering services and it will stop ssh and network timer client services.

```
daemon.sh PMD stop SSH NTP restart DB
```

# Appendix O - Start and Stop Daemons

## How to Configure Them

**Example of sourced shell script that activates the ntpclient service (type sig).**

```
# This file defines the NTP client configuration
ENABLE=NO             # Must be "NO" or "YES" (uppercase)
DNAME=ntpclient   # daemon name
DPATH=/bin           # daemon path
ShellInit=           # Performs any required initialization
ConfigFiles=         # configuration files
DTYPE=sig            # must be "sig" or "cmd" (lowercase)
DSIG=kill            # signal to stop/restart the daemon (lowercase)
                     # if it's hup term will be used to stop the daemon
# daemon command line parameters
NTPSERVER="-h 129.6.15.28"  # NTP server ip address
NTPINTERVAL="-l 300"        # Time in seconds to ask server
NTPCOUNT="-c 0"             # counter : 0 means forever
DPARM="$NTPCOUNT $NTPSERVER $NTPINTERVAL"
DSTOP=
```

**Example of sourced shell script that activates the ipsec service (type cmd).**

```
# This file defines the ipsec configuration
ENABLE=NO                   # Must be "NO" or "YES" (uppercase)
DNAME=ipsec                 # daemon name
DPATH=/usr/local/sbin       # daemon path
```

# Appendix O - Start and Stop Daemons

```
ShellInit=/etc/ipsec.init    # Performs any required initialization

ConfigFiles=                 # configuration files

DTYPE=cmd                    # must be "sig" or "cmd"

DSIG=kill          # signal to stop/restart the daemon (lowercase)

               # if it's hup term will be used to stop the daemon

# daemon command line parameters

DPARM="setup --start"

DSTOP="setup --stop"
```

# 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 AlterPath Console Server: 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

## A

Access Method 83
Alarm 194
Authentication 111

## B

Basic Wizard 76
Block Connector 351

## C

Cable Length 341
CLI 40
Command Line Interface 40, 75
Configuration using a Web browser 47
Connectors 342
CronD 145
Custom Wizard 43

## D

Data Buffers 148
Default Configuration Parameters 40
DHCP 161
DNS Server 42
Domain 42
Dual Power Management 165

## E

Ethernet 41

## F

Filters 167
Flash Memory Loss 407

## G

Gateway 41
    default 42
Generating Alarms 185

## H

Hardware Specifications 337
Hardware Test 410
HyperTerminal 41

## I

IP Address 42
IPsec 421, 497

## K

Kerberos 69, 112, 116, 360
Kermit 41

## L

Linux File Structure 322
Linux-PAM 381

## M

Minicom 41

# Index