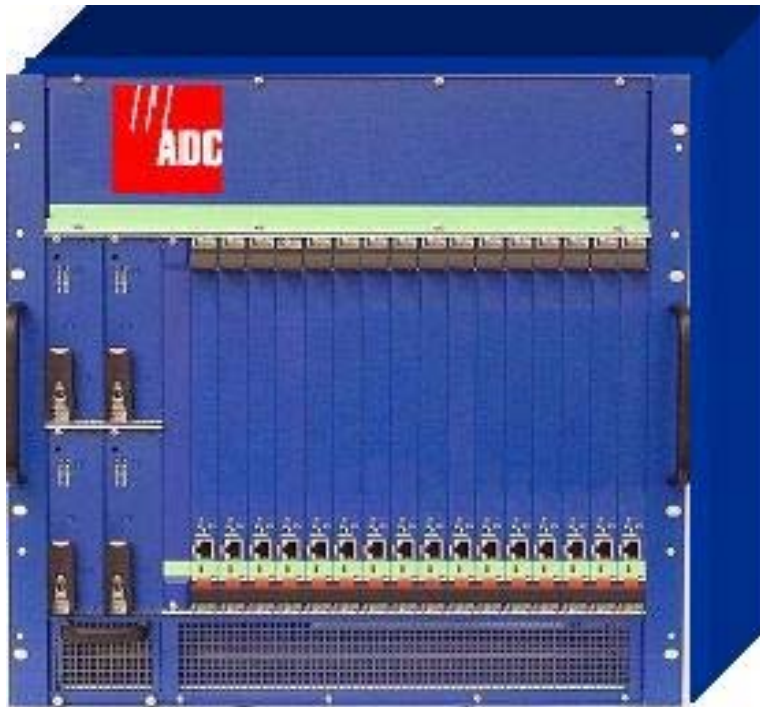

SG-1 Service Gateway System User Manual



Document Number: SG1-UM-8500-03



REVISION HISTORY

The Revision History provides a summary of any changes in this manual. Please make sure you are using the latest revision of this manual.

| Revision | Release Date | Revisions Made |
|----------|------------------|--------------------------------------------------------------------------------------------------------------------------|
| 01 | January 26, 2005 | Initial Release |
| 02 | August 5, 2005 | Revisions to various first-level and second-level commands. Addition of new Appendix: "SG-1 Vendor Specific Attributes." |
| 03 | June 30, 2006 | Software Upgrade. |

This manual is available online at ADC's website (www.adc.com/documentationlibrary/) or you can order copies of the manual by contacting your sales representative. Please ask for document SG1-UM-8500-03.

Copyright

©2006 ADC Telecommunications, Inc. All rights reserved.

Trademark Information

ADC is a registered trademark of ADC Telecommunications, Inc. No right, license, or interest to such trademarks is granted hereunder, and you agree that no such right, license, or interest shall be asserted by you with respect to such trademark.

Other product names mentioned in this practice are used for identification purposes only and may be trademarks or registered trademarks of their respective companies.

Disclaimer of Liability

Information contained in this document is company private to ADC Telecommunications, Inc. and shall not be modified, used, copied, reproduced or disclosed in whole or in part without the written consent of ADC.

Contents herein are current as of the date of publication. ADC reserves the right to change the contents without prior notice. In no event shall ADC be liable for any damages resulting from loss of data, loss of use, or loss of profits, and ADC further disclaims any and all liability for indirect, incidental, special, consequential or other similar damages. This disclaimer of liability applies to all products, publications and services during and after the warranty period.

Table of Contents

| | |
|--------------------------------------------------------------|-------------------|
| About This Manual | <i>x</i> |
| Introduction | <i>x</i> |
| Organization | <i>x</i> |
| Intended Audience | <i>x</i> |
| Conventions | <i>xii</i> |
| EU Compliance | <i>xii</i> |
| Inspecting Your Shipment | <i>xii</i> |
| Chapter 1: Overview | <i>1-1</i> |
| Features | 1-1 |
| Before You Begin | 1-2 |
| Site Preparations | 1-2 |
| Unpacking and Checking the Contents of Your Shipment | 1-3 |
| Required Tools and Equipment | 1-5 |
| Specific SG-1 Chassis Installation Requirements | 1-5 |
| Power Requirements | 1-6 |
| Blank Faceplate Requirement | 1-6 |
| Environmental Requirements | 1-6 |
| System Cabling Requirements | 1-6 |
| Chapter 2: Installation | <i>2-1</i> |
| Mounting the SG-1 Chassis | 2-1 |
| Connecting the SG-1 Chassis Ground | 2-1 |
| Connecting the Power Source | 2-1 |
| Installing Interface Cables | 2-2 |
| Connecting the Craft Port Interface | 2-4 |
| Powering Up the SG-1 | 2-4 |
| Installing Cards and Blank Faceplates | 2-4 |
| Serial Cable | 2-5 |
| Chapter 3: Command-Line Interface (CLI) | <i>3-1</i> |
| Overview | 3-1 |
| Understanding the Interface Structure | 3-1 |
| Commands and Navigation | 3-2 |
| Command-Line Editing | 3-3 |
| Chapter 4: Accessing the Command Line Interface | <i>4-1</i> |
| Connecting to the Craft Port | 4-1 |
| Logging on to the Craft Port | 4-1 |
| Setting the IP Address | 4-2 |
| Displaying the IP Address | 4-4 |

| | |
|----------------------------------------------------------|------------|
| Chapter 5: Using the Command Line Interface | 5-1 |
| Configuring the SG-1 | 5-1 |
| Logging On | 5-1 |
| Logging Off | 5-1 |
| What to Do Next | 5-2 |
| Chapter 6: First-Level Commands | 6-1 |
| Showing a List of Available Parameters | 6-2 |
| Using Debug Mode | 6-34 |
| Chapter 7: Second Level Commands | 7-1 |
| Banner Command | 7-3 |
| Ethernet Commands | 7-3 |
| Loopback Commands | 7-8 |
| VLAN Commands | 7-9 |
| Authentication Commands | 7-11 |
| ATM Commands | 7-14 |
| RADIUS Commands | 7-22 |
| Access List Commands | 7-25 |
| SNMP Commands | 7-28 |
| Tunnel commands | 7-30 |
| Timeouts Commands | 7-33 |
| Native IP Commands | 7-34 |
| Maximum Segment Size (MSS) Changing | 7-39 |
| L2TP and PPP Commands | 7-40 |
| DHCP Commands | 7-48 |
| IGMP Commands | 7-51 |
| Routing Command | 7-52 |
| Debug Commands | 7-71 |
| Appendix A: SG-1 Vendor-Specific Attributes | A-1 |
| Overview | A-1 |
| User Group | A-6 |
| dhcp group | A-16 |
| protocol group | A-18 |
| service group | A-18 |
| route group | A-26 |
| vpdn group | A-29 |
| qos group | A-33 |
| dns group | A-36 |

| | |
|---------------------------------------------|-------------|
| Appendix B: Redirection Server | B-1 |
| ORUP Commands | B-3 |
| Service Name Commands | B-4 |
| TFTP Commands | B-5 |
| Show Commands | B-8 |
| Ethernet Commands | B-10 |
| Default Gateway Commands | B-12 |
| Using Reload Command | B-13 |
| Write Commands | B-14 |
| Using Poweroff Command | B-15 |
| SNMP Commands | B-15 |
| Using Reset Configuration Command | B-17 |
| HTTP Commands | B-17 |
| Name Server | B-19 |
| Using Hostname Command | B-21 |
| Using EDS-URL-IDENTITY Command | B-21 |
| Using No EDS-URL-IDENTITY Command | B-22 |
| IP-IN-IP Commands | B-22 |
| Show User Commands | B-24 |
| Using Debug Protocol Command | B-25 |
| Using Rest Web Command | B-25 |
| Using Date Command | B-26 |
| Appendix C: Product Support | C-1 |
| Glossary | GL-1 |

List of Figures

| | |
|------------------------------------------------------------------|-----|
| Figure 1-1. SG-1 10U | 1-2 |
| Figure 2-1. Straight-Through and Cross-Over Cable Pin-Outs | 2-3 |

List of Tables

| | |
|---------------------------------------------------------|------|
| Table 1-1. Packing List | 1-4 |
| Table 1-2. System Installation Notes | 1-4 |
| Table 1-3. Possible SG-1 Options | 1-6 |
| Table 2-1. RJ-45 Pin-Outs | 2-3 |
| Table 3-1. SG-1 Sub-Menus and Associated Commands | 3-1 |
| Table 3-2. General Commands | 3-2 |
| Table 3-3. Navigation Commands | 3-3 |
| Table 4-1. Default Username/Password | 4-1 |
| Table 4-2. Interface Identification | 4-3 |
| Table 4-3. Ethernet Mode | 4-3 |
| Table 7-1. Configure Ethernet Ports | 7-4 |
| Table 7-2. Ethernet Operating Mode | 7-5 |
| Table 7-3. Configure Ethernet ports | 7-6 |
| Table 7-4. def-service-auth command parameters | 7-12 |
| Table 7-5. pppoa enable interface parameters | 7-17 |
| Table 7-6. interface atm command parameters | 7-19 |
| Table 7-7. radius-proxy client parameters | 7-23 |
| Table 7-8. ip radius source-interface parameters | 7-24 |
| Table 7-9. service internal parameters | 7-45 |
| Table 7-10.pppoe enable parameters | 7-47 |
| Table 7-11.ip dhcp relay server parameters | 7-48 |
| Table 7-12.ip dhcp relay information parameters | 7-49 |
| Table 7-13.ip igmp proxy command parameters | 7-51 |
| Table 7-14.ip route command parameters | 7-52 |
| Table 7-15.no ip route command parameters | 7-55 |
| Table 7-16.ip ospf advertise network command | 7-62 |
| Table 7-17.vrrp command parameters | 7-68 |
| Table 7-18.no vrrp command parameters | 7-70 |
| Table 7-19.vrrp preempt command parameters | 7-70 |
| Table A-1. Vendor-Specific Attribute List | A-1 |

ABOUT THIS MANUAL

INTRODUCTION

This manual applies to ADC's Service Gateway (SG) system, hereafter referred to as "SG-1." This document includes an overview of the platform, installation procedures, and an SG-1 commands reference.

ORGANIZATION

This manual includes the following chapters:

| Chapter | Description |
|-------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Chapter 1: Overview | Details features and pre-installation requirements for the SG-1 platform, including site requirements for power and cabling. |
| Chapter 2: Installation | Provides detailed procedures for installing SG-1. |
| Chapter 3: Command-Line Interface (CLI) | Describes the SG-1 Command-Line Interface (CLI), the steps to access the CLI, and the steps to perform initial configuration using the CLI. |
| Chapter 4: Accessing the Command Line Interface | Details how to log on locally to an SCC or rear I/O port (if a rear I/O card option is used) and set an IP address to allow for remote management via a Telnet session. |
| Chapter 5: Using the Command Line Interface | Describes how to access the command-line interface; it also directs you to other manuals for administering, configuring, and managing the SG-1. |
| Chapter 6: First-Level Commands | Defines the commands available at the first command level of each SCC. |
| Chapter 7: Second Level Commands | Defines commands available at the second command level. |
| Appendix A: SG-1 Vendor-Specific Attributes | Describes the vendor-specific attributes related to SG-1 EDS architecture. |
| Appendix B: Redirection Server | Explains how ADC's product redirects all peers' HTTP requests to their personal sites as predefined in the Radius Server. |
| Appendix C: Product Support | Provides information on how to contact the ADC Technical Support group. |
| Glossary | Defines abbreviations and acronyms for the SG-1 Service Gateway system. |







INTENDED AUDIENCE

This manual is intended for anyone needing to operate, administer, or maintain ADC's line of Service Creation Gateway products.

CONVENTIONS

The following style conventions and terminology are used throughout this guide.

| Element | Meaning |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bold font | Text that you must input exactly as shown (e.g., type 1 for card 1), menu buttons (e.g., ACCEPT SHELF OPTIONS) or menu screen options (e.g., ALARMS screen) that you must select |
| Italic font | Variables that you must determine before inputting the correct value (e.g., <i>Password</i>) |
| Monospace font | References to screen prompts (e.g., Invalid Password...Try Again:.) |

| Reader Alert | Meaning |
|-------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
|  | Alerts you to supplementary information |
| <u>IMPORTANT</u>  | Alerts you to supplementary information that is essential to the completion of a task |
|  | Alerts you to possible equipment damage from electrostatic discharge |
|  | Alerts you to possible data loss, service-affecting procedures, or other similar type problems |
|  | Alerts you that failure to take or avoid a specific action might result in hardware damage or loss of service |
|  | Alerts you that failure to take or avoid a specific action might result in personal harm |

EU COMPLIANCE

This product has been CE marked in accordance with the requirements of European Directive 73/23/EEC; the following mentioned product is in conformity with Low Voltage Directive 73/23/EEC in order to comply with the requirements in the Council Directive 73/23/EEC relating to electrical equipment designed for use within certain voltage limits and the Amendment Directive 93/68/EEC.

For safety evaluation of the compliance with this Directive 73/23/EEC, these standards were applied: IEC 60950:1999, EN 60950:2000.

INSPECTING YOUR SHIPMENT

Upon receipt of the equipment:

- Unpack each container and visually inspect the contents for signs of damage. If the equipment has been damaged in transit, immediately report the extent of damage to the transportation company and to ADC. Order replacement equipment, if necessary.
- Check the packing list to ensure complete and accurate shipment of each listed item. If the shipment is short or irregular, contact ADC as described in [Appendix C: Product Support on page C-1](#). If you must store the equipment for a prolonged period, store the equipment in its original container.

OVERVIEW

The SG-1 is a service creation platform optimized for delivering differentiated services to residential, mobile, and private subscribers. The SG-1 enables service providers to offer attractive new services that can be selected dynamically and automatically by their wireless, dial-up and broadband users.

The SG-1 can provide services over existing infrastructure, integrating smoothly with leading network access servers, RADIUS servers, databases, and billing systems.

The SG-1 can be deployed to meet the requirements of Internet Service Providers (ISPs), Digital Subscriber Loop (DSL) providers, cable providers, or "hot spot" (802.11) wireless Local Area Network (LAN) providers.

FEATURES

Designed to easily integrate into operations of existing and emerging service providers, the SG-1 has the following features:

- **Network Centric and Scalable Architecture:** is designed for a service provider network, scales to carrier class requirements, and allows for a more easily maintained solution because it is deployed in a centralized location. This minimizes the number of systems and makes service changes faster and more economical. As part of this network centric architecture, Customer Premises Equipment (CPE) software, "cookies", and local service points can be eliminated.



Note: Not all can be eliminated. For example, CPE equipment (such as, DSL modem, etc.) will still be needed. Cookies may still be needed depending on the application.

- **Dynamic Provisioning:** allows users to change a service package or user profile "on the fly" without forcing the user to disconnect and then reconnect. This vital feature enables streamlined service provisioning without involvement of a service provider's personnel and immediate service creation following the user's service-selection decision.
- **Flexible Network Integration:** can be easily operated with existing access infrastructure, requiring minor changes in network and billing servers.
- **Universal Platform:** the processing engines support different access types: xDSL, cable, wireless, and dial-up. The SG-1 also handles ATM and Gigabit Ethernet.
- **Flexible Scalability:** is available in a range of capacities, with varying configurations up to 64,000 virtual ports per chassis. As a result, service providers can easily scale SG-1's capacity to meet their changing needs.
- **Hot Swappable Cards:** has redundancies built in including the ability of cards to be mixed and matched.
- **Subscriber Redirection:** subscribers can be redirected to selected web sites or portals, regardless of their individual Uniform Resource Locator (URL) selection. This feature enables increased traffic to specific sites and personalized communications with individual users. It also enables a service provider to control and authorize access to incidental users that are not regular subscribers to the network. Subscriber redirection is a key building block for customized services that can be easily created using SG-1 including walled gardens, selective and targeted promotional activities, anti-virus protection, and other services and applications.
- **Anti-Spoofing Mechanism:** supplies an anti-spoofing prevention mechanism, which blocks an unauthorized user's computer from pretending to have a different IP address than it really has.
- **Billing and Accounting:** provides accounting information for each selected service using the standard RADIUS protocol. This enables network providers to bill their users based on the actual services used. Unique accounting capabilities are provided to simplify billing including those required for Session accounting and Service accounting using simple RADIUS commands.
- **Carrier Class:** supports telecom standards including Point-to-Point Protocol (PPP), Layer 2 Tunneling protocol (L2TP), SNMP, and others.

The SG-1 comprises two types of system chassis: 1U and 10U. The 1U chassis (or Mini System Chassis) has 2 service creation slots and 1 power supply slot built into the chassis. The 10U chassis (or Full-Size System Chassis) has 16 service creation slots and 4 power supply slots for load sharing redundancy (see [Figure 1-1](#)).

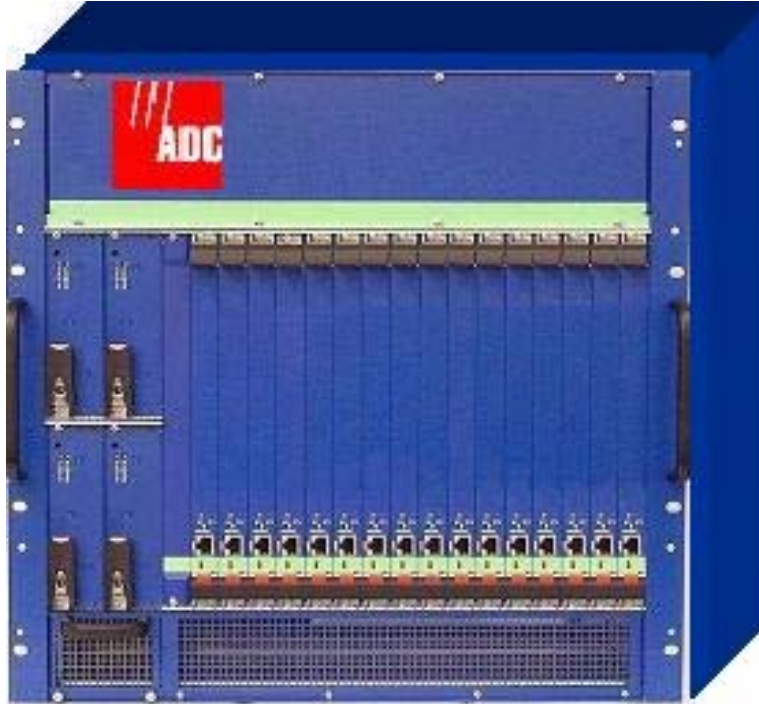


Figure 1-1. SG-1 10U

BEFORE YOU BEGIN

Before installing the SG-1 chassis and its associated modules, it is important to prepare for installation by:

- Preparing the site (site preparations) and reviewing the installation plans.
- Establish a Method of Procedure (MOP).
- Unpacking and inspecting the system components.
- Gathering the tools to properly install the SG-1 chassis and its associated modules.

SITE PREPARATIONS

Typically, you should have prepared the installation site beforehand. As part of your preparation, obtain a plan of the site or Telco environment where the SG-1 chassis will be installed. All personnel involved in the installation of the SG-1 chassis, including installers, engineers, and supervisors, should participate in the preparation of a MOP for approval by the customer.

Method of Procedure

An example of a pre-installation checklist of tasks and considerations (Method of Procedure) that needs to be addressed and agreed upon before proceeding with the installation is given below:

- Assign personnel.
- Determine protection requirements for personnel, equipment, and tools.
- Evaluate potential hazards that may affect service.
- Schedule time for installation.
- Determine any power and space requirements.

- Identify any required procedures and tests.
- On an equipment plan, make a preliminary decision that locates each of the SG-1 chassis that you plan to install.
- Read this manual, whether you are replacing or adding a SG-1 chassis that is being installed.
- Verify the list of replaceable parts for the installation (screws, bolts, washers, and so on) so that the parts are identified (see [Table 1-1 on page 1-4](#)).
- Check the required tools list to make sure the necessary tools are available (see [“Required Tools and Equipment” on page 1-5](#)).
- Purchase necessary parts.
- Identify work steps and any necessary notifications to CO personnel or engineers before work begins.
- Perform the installation (see [Chapter 2: Installation](#)).

UNPACKING AND CHECKING THE CONTENTS OF YOUR SHIPMENT

The shipping package for the SG-1 chassis is designed to reduce the possibility of product damage associated with routine material handling experienced during shipment. To reduce the potential damage to the product, transport the chassis in its ADC-specified packaging. Failure to do so may result in damage to the chassis. Do not remove the chassis from its shipping container until you are ready to install it.



Note: Do not discard the packaging materials used in shipping your SG-1 chassis. You will need the packaging materials in the future if you move or ship your SG-1 chassis.

[Table 1-1 on page 1-4](#) provides a list of required and optional components that may not be included in the SG-1 chassis kit, but are either required or recommended for the SG-1. A notes section ([Table 1-2 on page 1-4](#)) has been provided to document any components not listed below.

Table 1-1. Packing List


| Item | | Catalog/Part Number |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|---------------------|
| SG-1 Service Gateway System Chassis | | |
| SG-1 Service Gateway System User Manual | | |
| SG-1 Service Gateway System Installation Kit | | |
| <ul style="list-style-type: none"> • Ten (10) 6-32 x 3/16 screws: • Two (2) 8-32 x 5/16 screws: • Eight (8) 12-24 x 3/8 screws: • Two (2) mounting brackets • One (1) #6 ground cable • DB9/RJ45 “F” connector |  | |

Table 1-2. System Installation Notes

| |
|-----------------------|
| <p>Notes:</p> |
| Empty space for notes |

REQUIRED TOOLS AND EQUIPMENT

The following tools are required to install the SG-1 chassis:

- Grounding or ESD-preventive wrist strap
- No. 2 Phillips-head screwdriver
- Multimeter (for continuity testing)
- Wire stripper
- Wire-wrap tool
- Box cutter
- #26 AWG wire

SPECIFIC SG-1 CHASSIS INSTALLATION REQUIREMENTS

The SG-1 chassis dimensions are:

- Height of 17.50 inches (44.4 cm) (10U)
- Width of 19.0 inches (44.8 cm) without rack adapters attached to the left and right side of the unit
- Depth of 11.4 inches (28.9 cm)

IMPORTANT



Observe the clearances specified below in “Location Requirements” to provide proper air flow for chassis cooling.

Location Requirements

To install the chassis in an equipment room or central office:

- Install in a 19-inch, 23-inch, 24-inch, or 600 mm rack using the customer provided rack adapter brackets.
- Allow a 1-inch minimum clearance between the chassis to provide for proper air flow for cooling.

Use the following card types in the SG-1 chassis:

- Fast Ethernet card in slots 1 through 16
- Gigabit Ethernet card in slots 1 through 16
- ATM card in slots 1 through 16
- Rear I/O card on the rear of the chassis in slots 1 through 16

When selecting system components, consider future expansion of your SG-1 with items in [Table 1-3](#) as possible options.

Table 1-3. Possible SG-1 Options

| If you want to add: | Then: |
|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SG-1 chassis | Consider installing the first chassis in the top position of a rack to allow for further expansion below it. |
| Fast Ethernet Port | Add a Rear I/O card on the corresponding rear slot of an SCC that Fast Ethernet access is desired. |
| Service Creation Cards: | |
| VRRP | Use an SCC of the same type in any slot (1 through 16). Consider placing cards participating in functional groups in adjacent slots for ease of identification. |
| Redundancy | Add an additional SCC, of the same type, in slots 1 through 16 to have an additional card that provides the same service in the event of a card or connectivity failure. |

POWER REQUIREMENTS

The following specifies the power versions available for the SG1 chassis, then specifies the power requirements for your facility relative to the SG1 power version you selected.

SG-1 Chassis

The SG-1 chassis provides four 100 to 240 Vac (50 to 60 Hz) power supplies with AC power connectors. You must install one power supply for every five Service Creation Cards installed in a chassis (three power supplies are required for a chassis having slots 1 through 16 fully populated). If you want redundant power, install an additional power supply in an open power module slot. It is recommended to have one power module above the minimum to provide uninterrupted service should a power module fail.

Facility Requirements for AC Power

Verify that the facility AC power source for the primary connection falls within the recommended voltage range of 110 to 220 Vac with a maximum current of 10 amps for 110 Vac and 5 amps for 220 Vac.

BLANK FACEPLATE REQUIREMENT

When slots in an SG-1 chassis do not contain a card, the slot must be covered with a blank faceplate to prevent personnel contact with back panel connectors and to maintain proper air flow within the chassis.

ENVIRONMENTAL REQUIREMENTS

The SG-1 chassis has an ambient operating temperature range of +32 to +104° F (0 to +40° C) with a maximum humidity of 95% when installed according to the instructions in this installation manual.

The storage temperature range is from -4 to +158° F (-20 to +70° C).

SYSTEM CABLING REQUIREMENTS

You will complete only the cabling appropriate for the cards installed in the chassis.

Chassis Ground and Power Cabling

The recommended cabling to ground the SG-1 chassis is 6 AWG (minimum) stranded copper wire.

For the SG-1 chassis, the recommended cabling is 14 AWG (1.88 mm diameter) stranded copper or 14 AWG (1.628 mm diameter) solid wire to connect the DC terminal block to the facility provided power.

For the SG-1 chassis power cable, use one of the following:

- PC US for North America
- PC EURO for Europe
- PC UK for the United Kingdom

Network Cabling

Network connectors interface the SG-1 to an ATM backbone network, a LAN, or a WAN.

Configuration Port Cabling

In addition to the RJ-45 craft port provided on the front panel of each SCC card, each card may have a rear access RS-232 interface through a RJ-45 connector for craft access and configuration if a Rear I/O card is installed. There are two Ethernet ports located on the Rear I/O card.



Note: The Rear I/O card is optional when used with an ATM or Gigabit SCC.

INSTALLATION

This chapter provides detailed information about installing the SG-1.

MOUNTING THE SG-1 CHASSIS

To mount the SG-1, complete the following procedure.

| Step | Action |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | If required, securely attach the customer provided rack adapters to the left and right sides of the rack in which the SG-1 chassis will be installed. |
| 2 | Position the chassis in the rack. |
| 3 | Align the chassis adapter holes with the vertical rack mounting holes. |
| 4 | Secure the rack adapters to the rack using a Phillips screwdriver and four 12-24 x ½ inch pan head screws for each rack adapter. |

CONNECTING THE SG-1 CHASSIS GROUND

To connect the SG-1 chassis ground, complete the following procedure.

IMPORTANT



The recommended copper wire is a minimum 6AWG stranded copper wire with a maximum length of 5 feet (1.52 m).

| Step | Action |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Use the vendor provided cable. |
| 2 | Attach one end of the ground wire to the chassis ground lug and tighten the screw. Make sure the ground wire has a secure connection. |
| 3 | Connect the other end of the ground wire to the CO ground termination point or building earth ground. Make sure the ground wire has a secure connection. |

IMPORTANT



You must wear an antistatic wrist strap connected to the ESD jack on the SG-1 chassis to perform the installation procedures. You must also observe normal ESD precautions when handling electronic equipment. Do not hold electronic plugs by their edge. Do not touch components or circuitry.

Attach your antistatic wrist strap to the ESD ground jack on the SG-1 chassis.



Note: Procedures marked with an ESD symbol require you to use the antistatic wrist strap to complete the step.

CONNECTING THE POWER SOURCE

The SG-1 AC chassis (SG1-400-005) supports 110 to 240 Vac (50 to 60 Hz) power. Connect facility power to the chassis as described in the following section.

Connecting AC Power to an SG-1 AC Chassis

Connect an AC power cord(s) to AC power connectors, as required.

| Step | Action |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Turn off the AC power switch on the back of the SG-1 AC chassis. |
| 2 | Plug the power cord into the chassis power connector. |
| 3 | Connect the AC power cord from the power supply to the facility power source. Do not turn on the power switches at this time. You will turn on the power switches when you power up the chassis (as described in “Powering Up the SG-1” on page 2-4) and install the cards (as described in “Installing Cards and Blank Faceplates” on page 2-4). |

INSTALLING INTERFACE CABLES

This section provides procedures for installing the cabling for the network, subscriber, and management interfaces.

Connecting Network Cards

Connect the SG-1 system, through a network card interface, to an ATM backbone network, WAN, or LAN for a network uplink.

Refer to the following sections to complete cabling for the network interface connectors for these network cards:

- WAN ATM-OC3/STM1
- LAN GEthernet

Connecting the Ethernet Interface

The Rear I/O card provides two back panel access 10/100Base-T Ethernet interfaces.



Note: The Rear I/O is required for Fast Ethernet.

The Rear I/O card connector is MDI. Use one of the following cables as described below:

- Straight through cable to connect to a device with an MDI X port such as a hub, repeater, bridge, or router
- Cross over cable to connect to a device that also has an MDI port such as a PC with an Ethernet Network Interface Card (NIC)

Table 2-1 shows the pin-outs for the RJ-45 connectors.

Table 2-1. RJ-45 Pin-Outs

| MDI Pin Number | MDI-X Pin Number | Signal ^a | Symbol | Direction |
|----------------|------------------|---------------------|------------|---------------|
| 1 | 3 | Transmit Data (+) | TX+ (TX0+) | Out(Bidirect) |
| 2 | 6 | Transmit Data (-) | TX-(TX0-) | Out(Bidirect) |
| 3 | 1 | Receive Data (+) | RX+(TX1+) | In(Bidirect) |
| 4 | 4 | NC | (TX2+) | (Bidirect) |
| 5 | 5 | NC | (TX2-) | (Bidirect) |
| 6 | 2 | Receive Data (-) | RX-(TX1-) | In(Bidirect) |
| 7 | 7 | NC | (TX3-) | (Bidirect) |
| 8 | 8 | NC | (TX3+) | (Bidirect) |
| case | case | Chassis Ground | | |

a.NC = no connection.

Figure 2-1 shows the pin-outs for straight-through and cross-connect cabling.

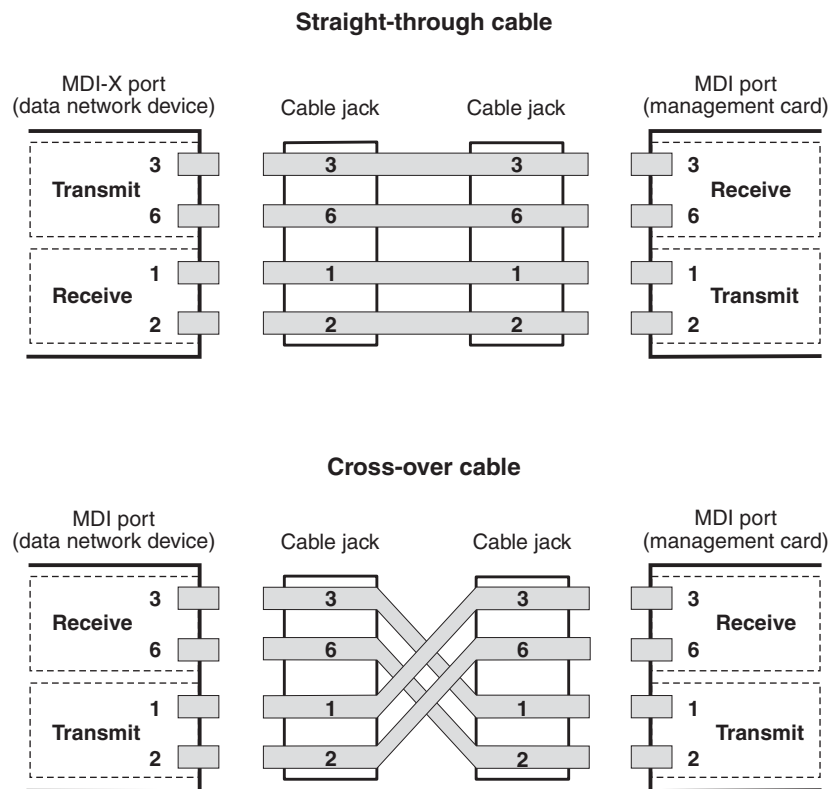


Figure 2-1. Straight-Through and Cross-Over Cable Pin-Outs

Connecting to an Ethernet Port

| Step | Action |
|------|---------------------------------------------------------------------------------------------------------|
| 1 | Plug the RJ-45 connector of the Ethernet cable into the FAST E-NET port on the SG-1 chassis back panel. |
| 2 | Connect the other end of the cable into the Ethernet port on the PC, hub, or other Ethernet device. |

CONNECTING THE CRAFT PORT INTERFACE

In situations where a Rear I/O card is installed, the default craft port is on the Rear I/O card. Moving a jumper on the Rear I/O card is required if you wish to use the front craft port.

POWERING UP THE SG-1

IMPORTANT *Electrical and mechanical shock hazards are present throughout the system; be aware of this possibility when power is applied to the chassis. Only qualified personnel should service the system.*



Connect to facility power using an AC power cord and confirm proper function of the AC power supplies.

| Step | Action |
|------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Connect the power cord from the AC power connector on the SG-1 AC chassis back panel to the facility power source. Turn on the power switch. |
| 2 | Verify that all cabling is securely terminated. |
| 3 | On each AC power supply, verify that the power LED lights green, indicating that the power supply is receiving power. |

INSTALLING CARDS AND BLANK FACEPLATES

Install SG-1 cards in the appropriate slots in the SG-1 chassis as indicated below. When slots do not have cards installed, use blank faceplates as indicated below.

Installing Cards



Note: SG-1 cards are inserted under power (hot inserted).

CAUTION *When inserting the cards, make sure the cards are properly aligned on the tracks to prevent equipment damage or personal injury.*

Once you've powered up the chassis, as described in "Powering Up the SG-1," you can begin to install cards in the SG-1 chassis.

Refer to guidelines for "Site Preparations" on page 1-2 to select the appropriate slot for a card, or refer to the applicable card installation manual to select the appropriate slot and for detailed information on the card.

Installing Blank Faceplates

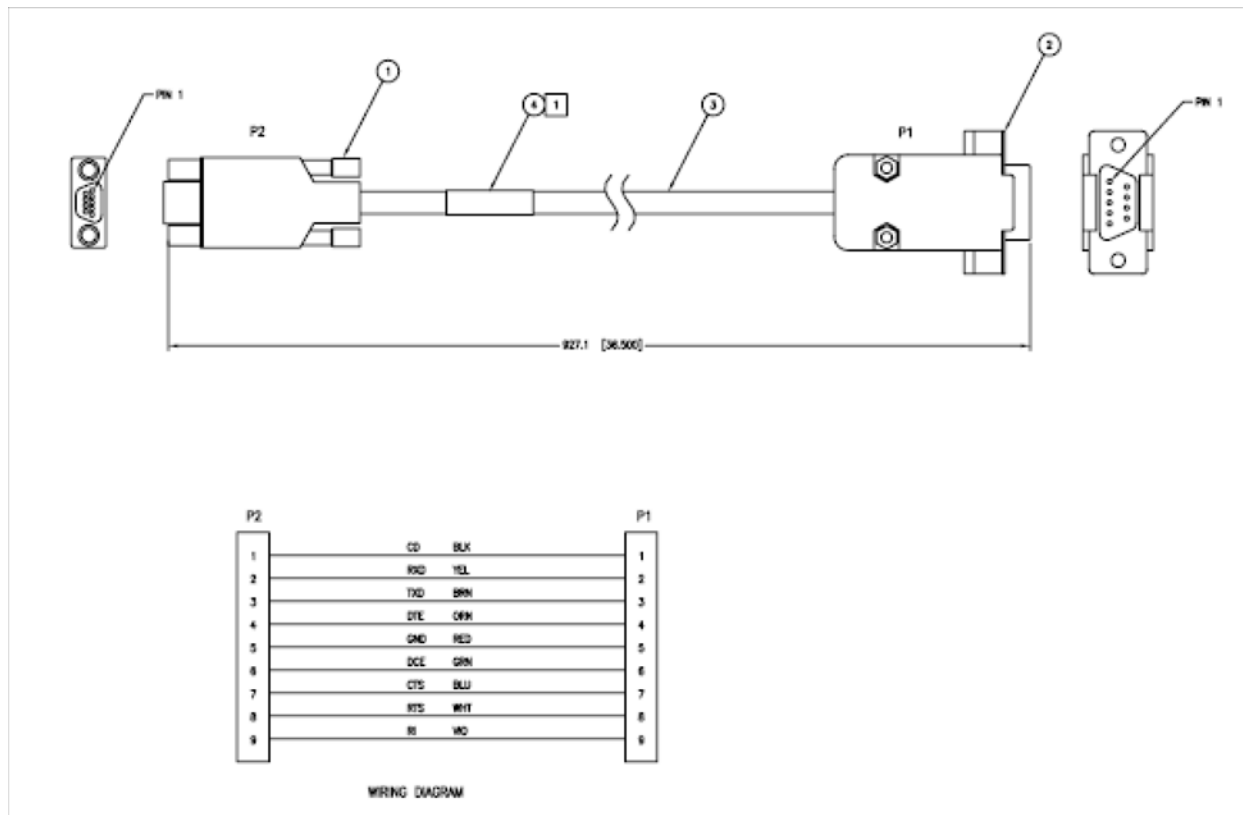
Use the blank faceplate identified in the “Blank Faceplate Requirement” on page 1-6.

IMPORTANT *Install blank faceplates in the SG-1 chassis to cover unused slots. Unused slots must be covered to prevent personnel contact with back panel connectors under power and to maintain proper airflow within the chassis.*



| Step | Action |
|------|-----------------------------------------------------------------------------------------|
| 1 | Slide the blank faceplate into the empty slot. Ensure the retaining latches are lifted. |
| 2 | Push the blank faceplate in until the retaining latches touch the SG-1 chassis. |
| 3 | Gently close the retaining latches until they snap into place. |
| 4 | Tighten the captive screw on the top and bottom retaining latches. |

SERIAL CABLE



COMMAND-LINE INTERFACE (CLI)

This chapter describes the SG-1 Command-Line Interface (CLI), the steps to access the CLI, and the steps to perform initial configuration using the CLI.

OVERVIEW

The SG-1 Service Gateway System management interface is accessed using a CLI, which provides comprehensive SG-1 system management including configuration, performance monitoring, and system maintenance and administration. An SG-1 Service Gateway System comprises an SG-1 10U Chassis with associated Service Creation Cards (10/100 SCC, GiG-E SCC, ATM/GiG-E SCC) and rear I/O cards.

The command-line interface is accessed through the SCC (or rear I/O card if installed) COM port using either a terminal connected directly to the COM port or over a network using a Telnet session. You can connect your Telnet session from:

- A PC connected on the SG-1 Ethernet Local Area Network (LAN).

OR

- A remote PC connected over a router to the SG-1 Ethernet port.

This chapter provides an introduction to the command-line interface structure and then provides information on how to use it.

UNDERSTANDING THE INTERFACE STRUCTURE

The command-line interface has four system management menus for administration, configuration, display, and diagnostics.

These four system management menus comprise four sub-menus (Configuration, Main, Debug, and Configuration-Debug). [Table 3-1](#) provides a description of these sub-menus and their associated commands.

Table 3-1. SG-1 Sub-Menus and Associated Commands

| Menu | Types of Commands |
|---------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Configuration | Use the commands in this second level menu to configure: <ul style="list-style-type: none">• port configuration• system passwords• default authentication service type• RADIUS parameters• ACL permissions• Tunnel server parameters• ATM parameters• PPP and LCP configuration• service parameters• system parameters• native IP parameters• VRRP parameters• IP parameters• debug message levels |
| Main | Use the commands in this menu to access other command levels and perform system updates, system reloading, and network troubleshooting. |

| Menu | Types of Commands |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Debug | Use the commands in this first level menu to display system parameters such as: <ul style="list-style-type: none"> • Memory allocation • Network/tunnel connections • Sonet clock source |
| Configuration-Debug | Use the commands in this third level menu to configure system debug messages such as: <ul style="list-style-type: none"> • Error and event level • Time server IP address • Log server IP address |

COMMANDS AND NAVIGATION

Navigate the command-line interface by entering a command name or a command string to move to the appropriate command level. The command level is indicated by the prompt. You can abbreviate command-line interface commands if the abbreviations are distinct; however, you must use at least two letters of the command. Also, the commands are not case sensitive. The following general and navigation commands are available from each prompt (see [Table 3-2](#) below and [Table 3-3](#) on page 3-3, respectively).

Table 3-2. General Commands

| Enter the following command: | To: |
|------------------------------------|---------------------------------------------------------------|
| ? | Display a list of commands available from the current prompt. |
| command name? (for example, show?) | Display an explanation of a particular command. |
| Exit | Leave the current level and return to the upper level. |

Command Path Navigation

For each command that provides configuration or management of a SG-1 system, a path is provided in the applicable section of the user document to help locate that command in the command-line interface structure. The path will be displayed in a box before the description of the command.

COMMAND-LINE EDITING

The command-line interface provides a DOS-like environment for editing. It provides special key functions and other special functions developed for a VT100-type terminal.



Note: Commands may not be recognized under some vendor's versions of Telnet. With Microsoft® Windows® HyperTerminal, or other terminal emulation programs, you may need to set the terminal preferences to VT100 arrows to use these functions (see [Table 3-3 on page 3-3](#)).

Table 3-3. Navigation Commands

| Use This Feature: | To: |
|--------------------------|----------------------------------------------------------------------------------------------------------------------|
| Up arrow key | Provide the capability to scroll backward through a list to retrieve information. |
| Down arrow key | Provide the capability to scroll forward through a list to retrieve information. |
| Tab key | Allows the completion of a command given the input of at least two characters and no other ambiguous commands occur. |

ACCESSING THE COMMAND LINE INTERFACE

The initial step for managing the SG-1 Service Gateway System is to log on locally to an SCC or rear I/O port (if a rear I/O card option is used) and set an IP address to allow for remote management via a Telnet session. This IP address should place the SG-1 system on the same subnet as a router or other device to which it connects upstream through its Ethernet port.

CONNECTING TO THE CRAFT PORT

Complete the following procedure to connect to the Craft port.

| Step | Action |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Connect one end of a straight-thru Ethernet cable to the front of the SCC or the COM 1 port of a rear I/O, if a rear I/O card is installed. Connect the other end of the straight-thru Ethernet cable to the DB9/RJ45 connector (provided in the SG-1 shipment) and then to the PC's COM1 or COM2 port. |
| 2 | Power up the ASCII terminal or PC. |

Please refer to the appropriate SCC installation manual for more detailed installation instructions, including the location of the respective SCC Craft port and a description of the SCC Craft port connector pin-outs.

LOGGING ON TO THE CRAFT PORT

Complete the following procedure to log on to the Craft port.



Note: If the keyboard remains inactive for five minutes, the command-line interface Inactivity Timer automatically logs the current user off. If this happens, log on to the command-line interface again. If debugging is activated, you may have to press **CTRL+C** to access the login prompt.

Table 4-1. Default Username/Password

| Username | Password | Authority |
|------------|----------|----------------------------------------------------|
| technician | ggcon | Full-access read/write |
| supervisor | sg1 | Read/write to second level |
| operator | Popgate | Read-only to first level with user drop capability |
| viewer | Popgate | First-level show capabilities |

If you are using a PC as a terminal, use a terminal emulation program such as HyperTerminal or Procomm™. Refer to your terminal emulation program's documentation for instructions.

| Step | Action |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Configure the terminal or the PC terminal emulation program as follows: <ul style="list-style-type: none">• Baud rate: 19200 bps• Data bits: 8• Parity: none• Stop bits: 1• Flow control: none |
| 2 | When using a PC, select the COM setting of the port to which the RS-232 cable is connected (for example, COM1 or COM2) using the terminal emulation program. |

| Step | Action |
|------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3 | Press ENTER to initiate the terminal session. |
| 4 | Enter your user name at the Username: prompt. |
| 5 | <p>Enter your password at the Password: prompt.</p> <p>The system will display the Host> prompt:</p> <pre>Welcome to SG-1 System Username: technician Password: ***** Login Successful Host></pre> |

SETTING THE IP ADDRESS

```
Host> configure terminal
Host(config)# interface ethernet
```

Set the management card IP address, subnet mask, and default gateway (if a gateway exists) to enable communication with external networks and to enable access to the SG-1 CLI for Telnet sessions.

To set a new IP address and subnet mask for the SCC, do the following.

| Step | Action |
|------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | From the Host (config)# prompt, enter the interface command in the following format: interface ethernet <slot number>\<port number> <ipaddr> <netmask> mode <ethernet mode> [mtu <1500 1544>] |
| 2 | From the Host (config)# prompt, enter the ip default-gateway command in the following format: ip default-gateway <ipaddr> |
| 3 | For this change to become permanent, the user must perform a write memory command from the Host> first level prompt to save this change to NVRAM (when the save command is issued, it saves all systems parameters): write memory |

Parameter(s)

<slot number>\<port number>

The SCC and rear I/O interfaces have specific designations as shown in [Table 4-2](#).

Table 4-2. Interface Identification

| Card/Type | Slot/Port 1 | Slot/Port 2 |
|------------------|-------------|-------------|
| Rear I/O | 1/1 | 1/2 |
| Gigabit Ethernet | 1/1 | 1/2 |
| ATM/GiG-E Card | | |
| ATM | 2/1 | 2/2 |
| GiG-E | 1/1 | 1/2 |

<ipaddr>

The SCC IP address for interface in question. This address is set up to be on the same subnet as the Ethernet network to which the SG-1 system is attached (format xxx.xxx.xxx.xxx).

<netmask>

The subnet mask associated with the SCC IP address. This address is based on the Ethernet network to which the SG-1 system is attached (format xxx.xxx.xxx.xxx).

<mode>

Table 4-3. Ethernet Mode

| Value | Explanation |
|-------|-------------------|
| 10H | 10M half-duplex |
| 10F | 10M full-duplex |
| 100H | 100M half-duplex |
| 100F | 100M full-duplex |
| 1000H | 1000M half-duplex |
| 1000F | 1000M full-duplex |
| Auto | Auto select mode |

Example(s)

```
Host> configure terminal
Host(config)# interface ethernet 0 \ 2 12.3.66.211 255.255.255.0 auto mtu 1500

Added Item
```

DISPLAYING THE IP ADDRESS

```
Host> show configuration
```

From the `Host>` prompt, enter the **show configuration** command to verify your configuration.

Example(s)

```
Host> show configuration
...
  interface ethernet 0\1 192.168.0.1 255.255.255.0 mode auto
  ip default-gateway 192.168.0.253
...
```

When the **show configuration** command is entered, A screen similar to the one above displays the SG-1 SCC card's Ethernet port IP address, subnet mask, and default gateway (if applicable).

USING THE COMMAND LINE INTERFACE

There are multiple ways to access an SG-1 for management. Also, there are rules that determine the number of accesses that can be made at one time to an SG-1 system. Once you have access, you can complete the configuration and management of the SG-1.

CONFIGURING THE SG-1

The operational software for an SCC resides on each individual card. The software is accessed through a command-line interface to configure and manage an SG-1.

The command-line interface can be accessed on an SG-1 locally through a serial interface to the Craft port on the SCC (or rear I/O card) or through a Telnet session as shown below. The command-line interface modifies and views the SG-1 Management Information Base (MIB) objects to implement system configuration and management.

Additionally, SG-1 can also be configured and managed through a customer-provided Element Management System (EMS). The EMS uses SNMP to modify and view the SG-1 MIB objects.

LOGGING ON

In addition to logging on to an SG-1 locally through the Craft port, up to four additional remote users can log on through a Telnet session.

The following example shows how to access the SG-1 CLI using a Telnet session from a remote system. Specify the IP address previously designated for this purpose:

```
C:\users\default>telnet 192.168.0.1
```

You receive a Welcome prompt when you successfully log on to the SG-1 CLI. Log on with the user account information assigned to you.

```
Welcome to SG-1 System

Username: technician
Password: *****

Host>
```

LOGGING OFF

To log off, enter the **exit** command at the first level `Host>` prompt.

```
Host> exit
```



Note: The CLI Inactivity Timer automatically logs the current user off the system if the keyboard remains inactive for five minutes.

WHAT TO DO NEXT

From the command-line interface, use the procedures in this manual to (among other things):

- Configure the network card ports, followed by services for the network connections, including Automatic Protection Switching (APS) for the OC3 card.
- Configure ATM traffic, including traffic profiles, policing, packet discard, over-subscription, and traffic shaping for Unspecified Bit Rate (UBR) traffic.
- Set up ATM routing parameters for either IISP (static ATM routing) or Private Network to Node Interface (PNNI) dynamic ATM routing.
- Set up system ATM configuration, such as signaling characteristics, ATM prefixes and addresses, ATM interfaces, and aliases for ATM addresses.
- Configure ATM connections for Permanent Virtual Circuits (PVC's).
- Set up bridging and routing global parameters and sessions over PVC's.
- Set up an IP/Ethernet uplink (bridging and routing over ATM PVC's) using an SCC card.

Use [Table 3-1 on page 3-1](#) to navigate command menus and perform the necessary actions.

FIRST-LEVEL COMMANDS

This chapter describes the commands available at the first command level of each SCC.

You can enter the entire command or the first two letters of most commands and command-line arguments. If there are two commands with the same first two letters, enter enough letters to differentiate between the two commands. The remaining letters can be displayed, if you wish to see the complete command, by pressing **TAB** after the first two letters. If a command-line argument is missing, the system responds by displaying the words `Incomplete command` or `illegal command` directly below the position of the missing argument.

Previously used command lines can be retrieved by pressing the **Up**↑ and **Down**↓ arrow keys to browse the list of these command lines. Any of the command lines can be executed by pressing **ENTER** when it appears on the screen.

You can view the command options and command-line argument options by typing `?` at any point before a command line is complete. For example, typing `?` right after the prompt will provide a list of available commands, and typing `?` after a command will display a list of options for the next command-line argument. Similarly, typing `?` after an option will display a list of options for the following command-line argument.

The help facility can be used to obtain a list of the first-level commands and their functions by typing `?` immediately after the first-level prompt. The question mark does not appear on the screen, but the system responds by displaying the information requested.

To verify the functionality of the initial configuration of the SCC, the commands at the first level may be used to:

1. Show hardware, software, and license versions
2. Show running configuration and port status
3. Show system information
4. Write to and Load from the system configuration on a TFTP server
5. Access the configuration menu
6. Disconnect users
7. Confirm network connectivity
8. View debug parameters

SHOWING A LIST OF AVAILABLE PARAMETERS

Using the GREP command

The GREP command may be operated on any of the system commands.

Usage

```
system-command | grep "string"
```

Example(s)

```
Host> show users | grep "moshe" <cr>

1      ANet  PPP      moshe          192.168.2.12    00:04:23  9568432

Host>
```

```
Host> write terminal |grep "interface" <cr>

interface loopback 1 10.1.208.1 255.255.255.0
interface ethernet 0\2 1.1.1.1 255.255.255.0 auto

Host>
```

Using the ? command

```
Host> ?
```

The list of available parameters can be viewed by typing ? at the first-level prompt.

Example(s)

```
Host> ?
  show           - Display running configuration and status
  write          - Write running configuration
  copy-TFTP      - copy-TFTP file from server
  ping           - Ping command
  reload         - Reload the system
  clear          - Disconnect line
  traceroute     - Traceroute command
  exit           - Exit SG-1 management session
  configure      - Modify running configuration
  debug          - Show debug information
Host>
```

Using the show ? command

```
Host> show ?
```

From the first-level Host> prompt, enter **show ?** to view a list of available commands.

Example(s)

```
Host> show ?
  show           - Display running configuration and status
  write          - Write running configuration
  copy-TFTP      - copy-TFTP file from server
  ping           - Ping command
  reload         - Reload the system
  clear          - Disconnect line
  traceroute     - Traceroute command
  exit           - Exit PopMaestro management session
  configure      - Modify running configuration
  debug          - Show debug information
Host>
```

These commands are discussed below.

Using the show version command

```
Host> show version
```

Enter **show version** to see version levels of hardware and software.

Usage

```
show version <software|hardware|pack>
```

```
show version software <SCC>
```

```
show version hardware <2>
```

Parameter(s)

```
<software|hardware|pack>
```

The first-level parameter has three options:

- **software**—Displays the software version of the SCC in question.
- **hardware**—Displays hardware information of the SCC in question.
- **pack**—Displays detailed information of the installed software version.

```
<SCC>
```

The software version for the Service Creation Card in question.

```
<2>
```

Module number.

Example(s)

```
Host> show version software

Module      Num  Application
-----
SCC-ATM155  1    10.0T2.05 Jun 08 2006 17:18:19

Host>
```

```
Host> show version hardware

Module      Num  Part No.          Serial No.      Slot
-----
Backplane   1    710-200-0 Rev 0   0
SCC-ATM155  2    650-038 Rev 1     2079600287     1

Host>
```

```
Host> show version hardware 2
Service Creation Card with 256MByte memory module

Module      Num  Part No.          Serial No.      Slot
-----
SCC         1    650003            8200935         0

Host>
```

Displaying the configuration in NVRAM

```
Host> show configuration
```

Usage

```
show configuration
```

Example(s)

```
Host> show configuration
# version: 10.0T2.05 Jun 08 2006 17:25:51
interface ethernet 0\1 172.16.1.13 255.255.255.128 auto
interface ethernet 0\2 172.16.13.193 255.255.255.128 auto
password viewer Tw)wtx-
password operator Tw)wtx-
password superuser +5z!#r-MGA
password technician Koz!#
password pre-authentication +k(~#i+^#43\,6
def-service-auth ppp-auto
multilink-mode multi-cage
radius-server host 172.16.1.15 auth-port 1812 acct-port 1813 m priority 3
radius-server key netix
access-list SNMP-permit 0.0.0.0 0.0.0.0
access-list SNMP-permit 10.0.0.0 255.0.0.0
access-list SNMP-permit 172.16.0.0 255.255.0.0
access-list SNMP-permit 192.168.1.1 255.255.0.0
access-list EDS-permit 10.0.1.203 255.255.255.255
access-list EDS-permit 162.10.1.0 255.255.255.0
access-list EDS-permit 172.16.1.2 255.255.255.255
access-list EDS-permit 172.16.1.15 255.255.255.255
access-list native-ip 172.16.13.0 255.255.255.0
SNMP-server community get T}y||g
SNMP-server community set T}y||g
tunnel-server host 0.0.0.0 mask 0.0.0.0
pppoe enable interface 0\1 0
router id 172.16.1.13
hostname Mankali
banner BannerString
session-timeout 500000
idle-timeout 1800
lcp renegotiate
service cache off
service internal framed-PPP
native-ip dhcp pre-auth-mode mac
native-ip def-service-auth Guest
native-ip enable interface Ethernet 0\2
```

```
ip local-pool pool1 162.10.1.1 162.10.1.254 internal
ip domain-name POPmaestro
ip primary-name-server 62.90.133.233
ip secondary-name-server 0.0.0.0
ip default-gateway 172.16.1.1
ip tcp adjust-mss on
ip dhcp relay server Ethernet 0\2 1 172.16.1.15
debug
watchdog-TimeValue 60
time-server-ip 0.0.0.0
error-level default 3 output-device console
event-level default 5 output-device console
trace default off
sysLog-server-ip 192.168.1.1
```

Host>



Note: Using **show configuration** for the first time on a blank system may return a File does not exist message until the configuration is written to NVRAM using the **write memory** command.

Displaying Ethernet port configurations

```
Host> show terminal
```

Displaying Ethernet Port Statistics

```
Host> show ethernet 0 \ 1
```

Use the **show ethernet** command to display the Ethernet port parameters for the Rear I/O Ethernet port.



Note: When keying in the command, the backward slash '\ ' is optional. The command will work with just a space between the slot number and port number.

Usage

```
show ethernet <slot number> <port number>
```

Parameter(s)

<slot number>

The slot number refers to *0* (for the Rear I/O) card and *1* for a Gigabit Ethernet port on the SCC card itself.

<port number>

The port number refers to a value of *1* to *3* for the card in question.

Example(s)

```
Host> show ethernet 0 \ 1
Interface Slot 0 Port 1 is up, line protocol is up
Hardware address is 008042195FB7
Internet address is 10.0.1.220 Mask is 255.255.255.0
Gateway IP address is 10.0.1.253
Duplex mode sensed by auto-negotiation is full-duplex
Ethernet speed is 1 Gbps
MTU 1500 bytes, BW 1000 Mbps
23778 packets input, 1997552 bytes
Received 5473 broadcast
, 0 runts, 0 giants, 0 CRC
Input frame discard = 0
Assign Rx buffers failure = 0
Free Rx buffers = 1005
13879 packets output, 1386325 bytes
0 output errors, 0 output late collisions, 0 retry
0 re-transmission limit
Output discards = 0
Redundancy status: redundancy is not configured

Host>
```

Displaying SONET port status

```
Host> show port sonet
```

Usage

```
show port sonet
```

Parameter(s)

None.

Example(s)

```
Host> show port sonet

ATM_SCC> show port sonet

  Slot      Port      Status      Capacity      Redundant      Redundant
  _____  _____  _____  _____  _____  _____
  2          1          OK          155          working       active
  2          2          OK          155          protected    not-active

ATM_SCC>
```

Displaying ATM Port Status

```
Host> show atm pvc
```

Usage

```
show atm pvc
```

Parameter(s)

None.

Example(s)

```
Host> show atm pvc
```

| Name | VPI | VCI | Slot | Port | Sub-port | PCR | Status |
|------|-----|-----|------|------|----------|-----|--------|
| none | 0 | 32 | 2 | 1 | 1 | 155 | UP |
| test | 2 | 32 | 2 | 1 | 1 | 155 | UP |
| test | 2 | 33 | 2 | 1 | 1 | 155 | UP |
| test | 2 | 34 | 2 | 1 | 1 | 155 | UP |
| test | 2 | 35 | 2 | 1 | 1 | 155 | UP |
| next | 2 | 36 | 2 | 1 | 1 | 155 | UP |
| next | 2 | 37 | 2 | 1 | 1 | 155 | UP |
| next | 2 | 38 | 2 | 1 | 1 | 155 | UP |

```
Host>
```


Displaying User Status

```
Host> show user
```

Usage

```
show [<cr>|<number>]
```

Parameter(s)

```
[<number>]
```

The line number of the user to be viewed.

Example(s)

```
Host> show users

Line Line  User User Name          IP Address      Duration  Calling
   Type Type
-----
52  Eth   PPP  status             155.226.20.50  00:01:00  0010A4C15AFB

Total number of Network connected lines: 1
  ANet (Analog source) lines: 0, INet (ISDN source) lines: 0
  Eth (Ethernet source) lines: 1
  ATM (ATM source) lines: 0
  EATM (ATM source) lines: 0
  PPP (PPP source) lines: 1
Total number of Framed users: 1
  PPP users: 1, MLP users: 0
Total number of tunnel switch users: 0
Total number of native IP users: 0
  NIPP (radius-proxy triggered) users: 0
  NIPD (dhcp-proxy triggered) users: 0
  NIPI (ip triggered) users: 0

Host>
```

```
Host> show users 704
Line number: 702 Line type: ANet User type: PPP
User name: 0_220
IP address: 10.220.3.191, IP pool name: 1
Next Hop: 10.0.1.253
Tunnel ID(in): 4798 Tunnel Session ID(in): 21182 LAC source IP 10.0.1.64
Session duration/timeout: 00:00:39 / 17:59:21
Idle duration/timeout: 00:00:35 / 00:30:00
Slot: 0 Port: 1
Calling number: <N/A> Called number: 0
Input packets/Octets: 3 / 30,
Output packets/Octets: 3 / 42,
Used Data Quota: 0
PPP Native IP Pipe: Off
Redirect Gateway: <N/A> Accounting Type: Standard
User Group: 0 DHCP Server IP: <N/A>
Service name: Original-Service
Service duration: 00:00:40

Host>
```

Displaying Routing Tables

```
Host> show ip-route
```

Use this command to display the configured routes.

Usage

show ip-route

Parameter(s)

None.

Example(s)

```
Host> show ip-route

Network          NetMask          Gateway          Interface
Address          Address          Address
-----          -
155.226.21.0     255.255.255.0   0.0.0.0         Ethernet [ 0\2 ]
155.226.22.128  255.255.255.128 0.0.0.0         Ethernet [ 0\2 ]

Host>
```

Displaying System Administrators

```
Host> show telnet-users
```

Usage

Use this command to display the system administrators that are logged onto the system.

Parameter(s)

None.

Example(s)

```
Host> show telnet-users

Number  User Level  Duration  Source
-----  -
1       Technician 00:01:07  Console
2       Technician 00:00:08  Network

Host>
```

Displaying System Parameters

```
Host> show system
```

Usage

```
show system  
show system <load>
```

Parameter(s)

```
<load>
```

Calculates the throughput through each interface in Mbps.

Example(s)

```
Host> show system  
Up-time: 1 Hours, 56 Minutes, 58 Seconds  
Total number of network incoming calls: 702  
Total number of network connected calls: 702  
Current number of Network connected lines: 1  
ANet (Analog source) lines: 1, INet (ISDN source) lines: 0  
Eth (Ethernet source) lines: 0  
ATM (ATM source) lines: 0  
Ethernet Over ATM (EATM source) lines: 0  
Current number of PPP (PPP source) lines: 0  
Current number of connected Framed users: 1  
PPP users: 1, MLP users: 0  
Current number of connected native IP users: 0  
  
Host>
```

```
Host> show system load
Calculating load ...
Total current connected users: 0
Total sessions' capacity: 2000
System load: 0%
CPU usage: 1%

Interface 0/2
Throughput [5 sec. Avg.]: 0.0 Mbit/s In, 0.0 Mbit/sec Out
Total available throughput: 100 Mbit/s In, 100 Mbit/sec Out
Traffic Usage: In 0.0%, Out 0.0%
Interface 1/1
Throughput [6 sec. Avg.]: 0.0 Mbit/s In, 0.0 Mbit/sec Out
Total available throughput: 1000 Mbit/s In, 1000 Mbit/sec Out
Traffic Usage: In 0.0%, Out 0.0%
Interface 2/1
Throughput [6 sec. Avg.]: 0.0 Mbit/s In, 0.0 Mbit/sec Out
Total available throughput: 155 Mbit/s In, 155 Mbit/sec Out
Traffic Usage: In 0.0%, Out 0.0%

Host>
```

Displaying License Attributes

```
Host> show license
```

This command displays the system license information.

Usage

show license

Parameter(s)

None.

Example(s)

```
Host> show license
Working license : permanent
Temp License Magic: T001001086

[CREATION]
DATE=December 22 2005 16:03:57
[VERSION]
MAJOR=10
[SN]
SNSOURCE=1
SNNUM=1
SN1=6046838
[MAGIC]
Magic String=C000000067
MaxAllowedDays=30
[OPTIONS]
Allow Maximum 500 Users=off
Allow Maximum 1000 Users=off
Allow Maximum 2000 Users=on
Allow Maximum 4000 Users=off
Gigabit Ethernet=on
ATM=on
Pre Paid=on
Bandwidth Control=on
Hierarchical Bandwidth Control=on
Customized Guided Entry=on
Dynamic COS=on
Differentiated Routing=on
```

```
Filter Redirection=on
Location Based Service=on
Service Selection=on
Native IP=on
Dynamic IP Changing=on
Application Awareness=on
MPLS=on
Native IP Roaming=on

Host>
```


Displaying VRRP attributes

```
Host> show vrrp interface
```

Use this command to display the configured Virtual Router Redundancy Protocol (VRRP) status on the specified interfaces.

Usage

```
show vrrp interface (<ethernet>|<vlan>) <slot number> <port number> <number>
```

Parameter(s)

(<ethernet>|<vlan>)

The interface will be either an Ethernet or VLAN interface.

<slot number>

Slot number; valid values are 0 to 3.

<port number>

Defines the port; valid values are 1 or 2.

<number>

Defines the Virtual Router ID number (VRID); valid values are 1 to 15.

Example(s)

For the master:

```
Host> show vrrp interface Ethernet 0\1
 Ethernet 0\1 - Group 1
 State is Master
 Virtual IP address is 192.168.1.1
 Master router is 192.168.1.2 (local)
 Virtual MAC address is 00-00-5E-00-01-01
 Advertisement interval is 1 seconds
 Priority 254
 Preemption mode: off
Host>
```

For the backup:

```
Host> show vrrp interface Ethernet 0\1
Ethernet 0\1 - Group 1
State is Backup
Virtual IP address is 192.168.1.1
Master router is 192.168.1.2
Virtual MAC address is 00-00-5E-00-01-01
Advertisement interval is 1 seconds
Priority 100
Preemption mode: on
Host>
```

Displaying active GRE and IP-in-IP tunnels

```
Host> show ip-tunnel
```

This command displays the active GRE and IP-in-IP tunnels in the system.

Usage

```
show ip-tunnel [gre | ip-in-ip [<remote endpoint Ip address> <tunnel direction>]]
```

Parameter(s)

None.

Example(s)

```
Host> show ip-tunnel <cr>
```

| Tunnel IP | Tunnel Endpoint | Total Sessions | Tunnel Status | Tunnel Type |
|-------------|-----------------|----------------|---------------|-------------|
| 192.168.1.1 | remote | 20 | unknown | gre |
| 10.10.2.234 | remote | 11 | up | ip-in-ip |
| 10.10.1.20 | remote | 7 | down | ip-in-ip |

```
Host>
```

```
Host> show ip-tunnel ip-in-ip <cr>
```

| Tunnel IP | Tunnel Endpoint | Total Sessions | Tunnel Status | Tunnel Type |
|-------------|-----------------|----------------|---------------|-------------|
| 10.10.2.234 | remote | 11 | unknown | ip-in-ip |
| 10.10.1.20 | remote | 7 | up | ip-in-ip |

```
Host>
```

```
Host> show ip-tunnel ip-in-ip 10.10.1.20 remote <cr>
```

| Line | Line Type | User Type | User-name | IP address |
|------|-----------|-----------|-----------|-------------|
| 2 | Eth | NIPI | test1 | 212.168.1.4 |
| 5 | Eth | NIPI | test2 | 212.168.1.5 |
| 56 | Eth | NIPI | test3 | 212.168.1.6 |

```
Host>
```

Displaying show mpls-labels commands

```
Host> show mpls-labels
```

It displays the incoming labels binding received from the neighbors and the out-going label binding distributed by the system. When no specific FEC is defined, the system shows all FECs.

Usage

```
Show mpls-labels <standard | vc> <out | in> [<FEC identifier>]
```

Parameter(s)

```
<standard | vc>
```

Standard labels are the first labels on the stack, while vc labels (tunnels) are the second label on the stack.

```
<out | in>
```

Labels that are out-going or incoming.

```
<FEC identifier>
```

It is the FEC (Forward Equivalence Class) identifier.



Note: The system default value is 0 (which means all FECs).

Example(s)

Standard in:

```
Host> show labels standard in
FEC ID      IP destination FEC      Label      Next-Hop      Interface
  1          192.168.1.0/24      123876      10.0.1.8      Ethernet 1
  2          212.8.1.0/24        1034        10.0.1.3      Ethernet 1
```

Standard out:

```
Host> show labels standard out
FEC ID      IP destination FEC      Label
  1          192.168.1.0/24      123876
  2          212.8.1.0/24        1034
```

VC in for Martini draft:

```
Host> show labels vc in
```

| VC ID | VC Type | Group ID | Label | Tunnel Endpoint | Upper stack FEC ID |
|-------|---------|------------|--------|-----------------|--------------------|
| 1 | LAN | 0xc2010000 | 123876 | 192.0.1.8 | 1 |
| 2 | LAN | 0xc2010000 | 123876 | 192.0.1.8 | 1 |
| 1 | VLAN | 0xc2010001 | 1034 | 212.1.3.4 | 2 |
| 1 | PPP | 0xc2010002 | 1035 | 195.3.4.5 | 3 |

VC OUT for Martini draft:

```
Host> show labels vc OUT
```

| VC ID | VC Type | Group ID | Label | Tunnel Endpoint |
|-------|---------|------------|--------|-----------------|
| 1 | LAN | 0xc2010000 | 123876 | 192.0.1.8 |
| 2 | LAN | 0xc2010000 | 123876 | 192.0.1.8 |

Standard in specific FEC table:

```
Host> show labels in 192.168.1.0/24 <cr>
```

| FEC | Label | Next-Hop | Interface |
|----------------|--------|----------|------------|
| 192.168.1.0/24 | 123876 | 10.0.1.8 | Ethernet 1 |

Displaying show mpls l2transport vc commands

```
Host> show mpls l2transport vc
```

It shows confine redirected interface traffic through MPLS I2vpn tunnel based on Martini draft.

Example(s)

```
Host> show mpls l2transport vc
Dest address          VC ID      Status Type
194.90.1.4            200        UP  redirect
194.90.1.4            201        UP  redirect
```

Displaying a list of available write commands

```
Host> write ?
```

This command shows the available write commands enabled on the SCC.

Usage

```
write [<memory>] | [<network><ip address><alpha numeric string>] | [<terminal>]
```

Parameter(s)

<memory>

Writes running configuration (volatile memory) to start-up configuration (non-volatile).

<network>

Writes running configuration to a TFTP server.

<ip address>

IP address of TFTP server.

<alpha numeric string>

Name to be used for configuration saved to a TFTP server.

<terminal>

Writes running configuration to terminal screen.

Example(s)

```
Host> write terminal

# version: 10 May 21 2006 15:14:31
password viewer Tw)wtx-
password operator Tw)wtx-
password superuser +5z!#r-MGA
password technician Koz!#
password pre-authentication +k(~#i+^#43)\,6
def-service-auth ppp-auto
multilink-mode multi-cage
radius-server key netix
SNMP-server community get T}y||g
SNMP-server community set T}y||g
banner BannerString
session-timeout 64800
idle-timeout 1800
service cache on aging-time 10
service internal framed-PPP
ip domain-name SG1
ip tcp adjust-mss off
debug
watchdog-TimeValue 600
time-server-ip 0.0.0.0
trace off
sysLog-server-ip 0.0.0.0

Host>
```

```
Host> write network 155.226.20.250 test
Preparing configuration file.....
Done.
Starting the TFTP upload.....
Done.
Host> write terminal
```

Using the copy-TFTP command

```
Host> copy-TFTP flash
```

Use this command to copy a new application software (image) or license file from a TFTP server to the flash memory of the SCC-ETH card.

Usage

```
copy-TFTP flash [<IP address> | <license>] <ip address> <file name>
```



Note: The SG-1 as a service creation machine uses a license mechanism, which enables the activation and deactivation of specific services. Use the [license] parameter to upgrade the current version of the license file running on the SCC.

Parameter(s)

<ip address>

This is the IP address of the TFTP server to be used for the software or license upgrade.

<license>

This parameter will allow the upgrading of a license file on the SCC.

<file name>

This is the file name used for the software or license upgrade (the file name is 1 to 64 alphanumeric characters).

Example(s)

```
Host> copy-TFTP flash 192.168.0.1 V10_PM
Download in progress
 99%
Closing the download process.
Pack loaded successfully.

In order to use new software, please reload the system

Host>
```

```
Host> copy-TFTP flash license 192.168.1.2 lic101.txt
```

Using the ping command to test network connectivity

Use the Ping command to elicit an ICMP ECHO_RESPONSE from a host or gateway. This command can be used to send a PING via an ATM or IP interface.

Usage

```
ping <ip address> [-c <number> | -i <number> | -s <number>] [atm <atm slot number>
<atm sub-interface number> <atm vpi number> <atm vci number> (end-loopback|repeat
<number>)]
```

Parameter(s)

<ip address>

The destination ip address to ping.

`[-c <number>]`

The number of echoes.

`[-i <number>]`

The number of wait seconds.

`[-s <number>]`

The number of bytes.

`[atm]`

The ATM interface.

`[atm <slot number>]`

The ATM slot number (ATM uses slot 2).

`[atm <atm port number>]`

The ATM port number (either port 1 or port 2).

`[atm <sub-interface number>]`

The ATM sub-interface number.

`[atm <vpi number>]`

The ATM vpi number.

`[atm <vci number>]`

The ATM vci number.

`[atm (end-loopback)]`

Verifies end-to-end PVC integrity.

`[atm (repeat <number of echoes>)]`

The number of echoes to be sent.

Example(s)

```
Host> ping -c 5 -i 5 -s 500 155.226.20.250
Press <Ctrl C> to abort.
508 bytes from 155.226.20.250:icmp_seq=1 ttl=63 time<20 ms
508 bytes from 155.226.20.250:icmp_seq=2 ttl=63 time<10 ms
508 bytes from 155.226.20.250:icmp_seq=3 ttl=63 time<10 ms
508 bytes from 155.226.20.250:icmp_seq=4 ttl=63 time<10 ms
508 bytes from 155.226.20.250:icmp_seq=5 ttl=63 time<20 ms
Host>
```

Using the reload command to restart the system

```
Host> reload non-graceful
```

Use the reload non-graceful command to reset the system and reload the software. Using this command will terminate all sessions.

Usage

```
reload non-graceful
```

Parameter(s)

None.

Example(s)

```
Host> reload non-graceful
```

```
Resetting the system in 10 seconds...
```

Clearing Users

```
Host> clear user
```

A specific user can be disconnected from the SG-1 by writing the **clear user** command followed by its line number.

Usage

```
clear user <line number>
```



Note: You can use the **show user** command to determine the list of connected users.



Note: If the user is a multi-link user, the following message will appear: Note that multi-link users may have other lines that are still connected.

Parameter(s)

```
<line number>
```

The line number of the user to be cleared.

Example(s)

```
Host> clear user 22
```

```
Note that multi-link users may have other lines that are still connected
```

Using the Traceroute Command

```
Host> traceroute
```

Use the traceroute command to track the route a packet takes to a network host.

Usage

```
traceroute <ip address> [-h <number>| -i <seconds>]
```

Parameter(s)

<ip address>

The IP address to which the trace is to be performed.

[-h <number>]

The maximum number of hops to be attempted (max = 30).

[-i <seconds>]

The max. number of seconds to wait for a response.

Example(s)

```
Host> traceroute 155.226.10.200

Press <Ctrl C> to abort.
 1   < 20 ms   < 10 ms   < 10 ms   155.226.20.240
 2   < 10 ms   < 10 ms   < 10 ms   155.226.248.14
 3   < 10 ms   < 10 ms   < 10 ms   155.226.1.2
 4   < 10 ms   < 10 ms   < 10 ms   155.226.10.200

Traceroute completed successfully

Host>
```

Using the exit command

```
Host> exit
```

This command exits the user from the current configuration level. When used at the first level, the user is logged out of the session.

Usage

exit

Parameter(s)

None.

Example(s)

```
Host> (config-debug)# exit  
  
Host> (config)#  
  
Host> (config)# exit  
  
Host> exit
```

USING DEBUG MODE

This section provides information on the commands and options available in debug mode.

Switching to Debug Mode

```
Host> debug
```

Use the **debug** command at the first-level prompt (`Host>`) to switch the system to the second-level debug prompt: `Host(debug)#`. This prompt indicates that the user is now in the second-level debug mode and has access to the five second-level commands in the debug menu.

Usage

| Step | Action |
|------|---------------------------------------------------------------------------------------------------------------|
| 1 | Use the show command to display memory, system and fragmentation information. |
| 2 | The port sonet source-clock command determines timing configuration for the ATM port specified. |
| 3 | At this level, the exit and end commands will return the user to the first configuration level. |
| 4 | From the <code>Host></code> prompt, enter the debug command in the following format: debug |

Parameter(s)

None.

Example(s)

```
Host(debug)# ?
show          - Display debug parameters
port         - Select a port to configure
port-ethernet - Port ethernet interface
radius-server - radius server command
clear        - Clear arp table
exit         - Exit current level
end          - Return to first level

Host(debug)#
```

Using the show command in debug mode

```
Host (debug)# show
```

Use the **show** command to display memory, system, and fragmentation information.

Usage

```
show [memory | system | log-modules | statistics | arp]
```

Parameter(s)

[memory]

Display memory allocation.

[system]

Display connection information.

[log-modules]

Display system log modules.

[statistics]

Display fragmentation information.

[arp]

Display ARP information table.

Example(s)

```
Host(debug)# show statistics fragmentation

Total number of packets that were fragmented: 0

Total number of packets that were reassembled: 0

Total number of upstream packets in which the MSS field was adjusted: 0

Total number of downstream packets in which the MSS field was adjusted: 0

Host(debug)#
```

```
Host(debug)# show system

Up-time: 4 Days, 20 Hours, 56 Minutes, 12 Seconds

Total number of:
Network incoming calls: 0
Network calls in which PPP was established: 0
Network calls in which PPP established via LCP options: 0
Network calls in which link was authenticated: 0
Network connected calls: 0

Current number of Network connected lines: 0
  ANet (Analog source) lines: 0, INet (ISDN source) lines: 0
  Eth (Ethernet source) lines: 0
  ATM (ATM source) lines: 0
  Ethernet Over ATM (EATM source) lines: 0
Current number of PPP (PPP source) lines: 0:
Current number of connected Framed users: 0
  PPP users: 0, MLP users: 0
Current number of connected native IP users: 0

Total connected LNS tunnels: 0
Total connected PPPoE tunnels: 0
Total connected PPPoA tunnels: 0
Total connected LAC tunnels: 0
Maximum used of tunnel sessions: 0

Host(debug)#
```



```
Host(debug)# show log-modules
```

| Group | Error | Error | Event | Event | Trace |
|--------------|--------------|--------------|--------------|--------------|--------------|
| Name | Min | Max | Min | Max | |
| AAA | 0 | DEF | 0 | DEF | DEF |
| User | 0 | DEF | 0 | DEF | DEF |
| IP | 0 | DEF | 0 | DEF | DEF |
| Route | 0 | DEF | 0 | DEF | DEF |
| System | 0 | DEF | 0 | DEF | DEF |
| PPP | 0 | DEF | 0 | DEF | DEF |
| Service | 0 | DEF | 0 | DEF | DEF |
| NativeIP | 0 | DEF | 0 | DEF | DEF |
| Interface | 0 | DEF | 0 | DEF | DEF |
| L2TP | 0 | DEF | 0 | DEF | DEF |

| Module | Group | Error | Error | Event | Event | Trace |
|----------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Name | Name | Min | Max | Min | Max | |
| ABM | User | 0 | DEF | 0 | DEF | DEF |
| AbmFSM | User | 0 | DEF | 0 | DEF | DEF |
| AbmIpPool | User | 0 | DEF | 0 | DEF | DEF |
| AbmMIPPP | User | 0 | DEF | 0 | DEF | DEF |
| AbmRadius | AAA | 0 | DEF | 0 | DEF | DEF |
| AbmService | User | 0 | DEF | 0 | DEF | DEF |
| AbmRadiusProxy | AAA | 0 | DEF | 0 | DEF | DEF |
| EDS | Service | 0 | DEF | 0 | DEF | DEF |
| PPP | PPP | 0 | DEF | 0 | DEF | DEF |
| PPPWrapper | PPP | 0 | DEF | 0 | DEF | DEF |
| PPPService | PPP | 0 | DEF | 0 | DEF | DEF |
| NativeIP | NativeIP | 0 | DEF | 0 | DEF | DEF |
| L2TP | L2TP | 0 | DEF | 0 | DEF | DEF |
| Telnet | IP | 0 | DEF | 0 | DEF | DEF |
| CPM | User | 0 | DEF | 0 | DEF | DEF |
| System | System | 0 | DEF | 0 | DEF | DEF |
| DataPoller | System | 0 | DEF | 0 | DEF | DEF |
| BSP | Interface | 0 | DEF | 0 | DEF | DEF |
| ARP | Interface | 0 | DEF | 0 | DEF | DEF |
| DHCP | NativeIP | 0 | DEF | 0 | DEF | DEF |
| Router | Route | 0 | DEF | 0 | DEF | DEF |
| OSPFv2 | Route | 0 | DEF | 0 | DEF | DEF |
| IPMgr | IP | 0 | DEF | 0 | DEF | DEF |
| IPinIP | IP | 0 | DEF | 0 | DEF | DEF |
| ICMP | IP | 0 | DEF | 0 | DEF | DEF |
| POPUDP | IP | 0 | DEF | 0 | DEF | DEF |
| RsmFrgm | IP | 0 | DEF | 0 | DEF | DEF |
| VRRP | Route | 0 | DEF | 0 | DEF | DEF |
| MPLS | Route | 0 | DEF | 0 | DEF | DEF |
| Netlf | Interface | 0 | DEF | 0 | DEF | DEF |
| SysLogger | System | 0 | DEF | 0 | DEF | DEF |
| Timer | System | 0 | DEF | 0 | DEF | DEF |
| Default Max Error Level: 3 | | | | | | |
| Default Max Event Level: 5 | | | | | | |
| Default Trace Setting: Off | | | | | | |

show arp command

This command displays the arp table information.

```
Host(debug)# show arp
```

Usage

Show arp [<index><NextAddress>]

Parameter(s)

<index>

The interface on which this entry's equivalence is effective. Numbers are 1 to 65,000.

<NextAddress>

This is the IP address corresponding to the media-dependent "physical address. It should be a legal IP address.

Example

```
Host(debug)# show arp
```

| IfIndex | PhyAddress | NetAddress | MediaType |
|---------|--------------|---------------|--------------|
| 1 | 000000000000 | 127.0.0.1 | static |
| 4004 | 00D0B715A7E3 | 172.16.1.1 | non-volatile |
| 4004 | 0001AF0A1CB3 | 172.16.1.13 | static |
| 4004 | 00D0B7174CF7 | 172.16.1.15 | dynamic |
| 4005 | 0001AF0A1CB2 | 172.16.13.193 | static |

Clear arp command

It clears the arp table entry (only dynamic entries), the clear arp specific, clears any entry except the static ones.

Note: when deleting a non-volatile arp entry the system might re-creates it as a dynamic entry.

Usage

Clear arp [arp-specific <ifindex><NetAddress>]

Parameter(s)

<ifindex>

This is the interface on which this entry's equivalence is effective. Numbers are 1 to 65,000.

<NetAddress>

This is the IP address corresponding to the media-dependent "physical" address. It should be a legal IP address.

Example

```
Host(debug)# clear arp-specific 4004 172.16.1.1 <cr>
```

```

Host(debug)# show memory

Free memory:
  region 0: 9583616
  region 1: 56918016
Largest memory   :
  region 0 buffer: 9583616
  region 1 buffer: 56901632

pNA statistics:
Number of classes:   8 - blocks: 15000 free: 12492 wait: 0 drops:
0
Buffer size  0 - blocks:  9000 free:  6500 wait:  0 drops:  0
Buffer size 32 - blocks:  2048 free:  2043 wait:  0 drops:  0
Buffer size 64 - blocks:   512 free:   512 wait:  0 drops:  0
Buffer size 128 - blocks:  768 free:  766 wait:  0 drops:  0
Buffer size 256 - blocks:  512 free:  512 wait:  0 drops:  0
Buffer size 512 - blocks:  128 free:  128 wait:  0 drops:  0
Buffer size 1024 - blocks:  256 free:  256 wait:  0 drops:  0
Buffer size 2048 - blocks:   48 free:   47 wait:  0 drops:  0

pSOS statistics:
Buffer size  8 - blocks: 450000 free: 429307 wait:  0 drops:  0
Buffer size 20 - blocks:  60000 free:  39843 wait:  0 drops:  0
Buffer size 24 - blocks:  90000 free:  86428 wait:  0 drops:  0
Buffer size 32 - blocks: 550000 free: 535673 wait:  0 drops:  0
Buffer size 44 - blocks: 330000 free: 325103 wait:  0 drops:  0
Buffer size 48 - blocks: 800000 free: 798995 wait:  0 drops:  0
Buffer size 52 - blocks: 260000 free: 186211 wait:  0 drops:  0
Buffer size 56 - blocks:  96000 free:  94486 wait:  0 drops:  0
Buffer size 68 - blocks:  45000 free:  35440 wait:  0 drops:  0
Buffer size 72 - blocks: 10000 free:  9879 wait:  0 drops:  0
Buffer size 76 - blocks: 12000 free: 11771 wait:  0 drops:  0
Buffer size 80 - blocks: 70000 free: 69802 wait:  0 drops:  0
Buffer size 92 - blocks: 10000 free:  9116 wait:  0 drops:  0
Buffer size 96 - blocks: 10000 free:  8781 wait:  0 drops:  0
Buffer size 104 - blocks: 11000 free: 10084 wait:  0 drops:  0
Buffer size 128 - blocks: 10000 free:  9159 wait:  0 drops:  0
Buffer size 160 - blocks:  6000 free:  5756 wait:  0 drops:  0
Buffer size 192 - blocks:  6000 free:  5607 wait:  0 drops:  0
Buffer size 244 - blocks: 15000 free: 14783 wait:  0 drops:  0
Buffer size 272 - blocks: 10000 free:  9597 wait:  0 drops:  0
Buffer size 304 - blocks:  4000 free:  3895 wait:  0 drops:  0
Buffer size 348 - blocks:  8000 free:  7992 wait:  0 drops:  0
Buffer size 528 - blocks: 24000 free: 23442 wait:  0 drops:  0
Buffer size 636 - blocks:  7000 free:  6536 wait:  0 drops:  0
Buffer size 684 - blocks:  5000 free:  4896 wait:  0 drops:  0
Buffer size 800 - blocks:  5000 free:  4893 wait:  0 drops:  0
Buffer size 908 - blocks:  4000 free:  3949 wait:  0 drops:  0
Buffer size 1064 - blocks:  8100 free:  7783 wait:  0 drops:  0
Buffer size 2048 - blocks:  6200 free:  6011 wait:  0 drops:  0
Buffer size 2304 - blocks:  2018 free:  1912 wait:  0 drops:  0

Host(debug)#

```

Defining port-ethernet redundancy-mode command

```
Host(debug)# port-ethernet redundancy-mode
```

It immediately activates the Ethernet redundancy operation.

Usage

```
port-ethernet redundancy-mode <working slot>\<working port> [<auto | force-protecting | force-working>]
```

Parameter(s)

[working slot]

It is the working, Ethernet interface slot number; legal values include 0 and 1.

[working port]

It is the working, Ethernet interface physical port number; legal values include 1 and 2.

[auto | force-protecting | force-working]

- auto – set redundancy mode to auto causing a normal redundancy behavior
- force-protecting – force traffic through the protecting interface
- force-working – force traffic through the working interface

Example(s)

```
Host(debug)# port-ethernet redundancy-mode 1\1 force-protecting
```

Defining the SONET port source clock

```
Host(debug)# port sonet source-clock
```

The **port sonet source-clock** command defines the sonet port source clock.

Usage

```
port sonet source-clock <slot> <port> [loop | free-running]
```

Parameter(s)

<slot>

Specify atm/sonet slot, 2 is the only value used since this command is only applicable to the ATM/SONET port.

<port>

ATM/SONET port 1 or 2.

[loop]

The transmit clock is derived from the clock source received on the same interface.

[free-running]

The transmit clock on the interface is delivered from the port adapter's local oscillator.

Checking the system RADIUS interface

```
Host(debug)# radius-server check
```

This command checks the system RADIUS interface, by authenticating user-name and password the same way the system authenticates a connected call (including retries and RADIUS redundancy).

Usage

```
radius-server check <user-name> <password> auth-type [<PAP>]
```

Parameter(s)

<user-name>

Username to be validated at the RADIUS server.

<password>

Password to be validated at the RADIUS server.

<PAP>

Protocol used for authentication.

Example(s)

```
Host(debug)# radius-server check utest ptest <cr>
```

| Radius IP | Status | Retries |
|-------------|---------|---------|
| 192.168.1.2 | Failed | 3 |
| 192.168.1.2 | Success | 1 |

```
Host>
```


SECOND LEVEL COMMANDS

This chapter describes the primary commands available at the second command level. For additional second level commands, refer to [Appendix B: Redirection Server](#).



Note: Some non-applicable information was removed from sample screens for ease of viewing.

Using the `configure` command

Use the **configure terminal** command at the first level prompt to switch the system to the second-level configuration prompt, `Host(config)#`. This indicates that the user is now in the second level configuration mode and has access to the second level commands in the configuration menu. These commands change the parameters only in RAM. You can save a new configuration in the flash memory by returning to the first level prompt and using the **write memory** command.



Note: Use the help facility to view a list of the second level configuration commands and their functions by typing `?` immediately after the second level configuration prompt.

Usage

```
configure [<network> <ip address> <alpha numeric string>]
```

```
configure [<terminal>]
```



Using the **write memory** command following the use of the `configure` parameter will overwrite the configuration loaded from the TFTP server.

Parameter(s)

```
[<network>]
```

Copies a previously saved configuration from a TFTP server to NVRAM.

```
[<ip address>]
```

The IP address of the TFTP server to which the running configuration will be written.

```
[<alpha numeric string>]
```

The filename of the running configuration to be saved.

```
[<terminal>]
```

Places the user in the second level command level.

Example(s)

```
Host> configure network 155.226.20.250 filename

Loading file ...

Preparing TFTP download...Done.

Starting the TFTP download....completed(downloaded size is 892 ).

Converting file ...

File loaded successfully

Host>
```

```
Host> configure terminal

Host(config)#
```

BANNER COMMAND

Creating a Login Banner

You may create a greeting message or banner, to be displayed on the user's terminal when they log in. The banner may be a string of up to 32 alphanumeric characters.

To set "Welcome to SG-1" as a banner:

At the second level command prompt, type: `banner welcome to SG-1` (then press **ENTER**).

Configuring Banner command

```
Host(config)# banner command
```

Usage

`Host(config)# banner <Alpha numeric string>`

Parameter

<Alpha numeric string>

It is the Terminal greeting (maximum 32 characters).

Example

```
Host(config)# banner ADC
```

ETHERNET COMMANDS

This section contains instructions for configuring service for Gigabit Ethernet cards installed in a SCG1664A/D.

Configuring Gigabit Ethernet ports

```
Host(config)# interface ethernet
```

Use this command to configure service for Ethernet cards.

Usage

```
interface ethernet <port> \ <slot> <ip address> <mask> <ethernet mode> (mtu <1500|1544>)
```

Parameter(s)

<port>

The port number that you want to configure.

<slot>

The slot number that you want to configure (see table 7-1)<ip address>

The ip address to be assigned to the interface.

<mask>

The valid subnet mask for the configured ip address.

<ethernet mode>

The operating mode of the interface to be configured (see [Table 7-2](#)).

(mtu)

The maximum transmission unit. Select either:

- 1500 (default)
- 1544

Example 1

```
Host(config)# interface ethernet 0 \ 2 12.3.66.211 255.255.255.0 auto mtu 1500
interface ethernet 0 \ 2 12.3.66.211 255.255.255.0 auto mtu 1500
```



Note: The system should refuse to assign an address to an Ethernet interface which is in the subnet of the other Ethernet interface and indicates the reason.

Example 2

```
Host(config)# interface ethernet 0\2 192.168.3.5. 255.255.255.0
Operation Error:
ethernet address is not valid
```



Note: The system should ignore the **interface ethernet** command if the system is in Ethernet redundancy mode. It will indicate a command failure to the system log and to the user interface (telnet).

Table 7-1. Configure Ethernet Ports

| Card Type | Slot\Port | Slot\Port 2 |
|-------------------------|-----------|-------------|
| Gigabit Front I/O | 0\1 | |
| Gigabit Rear I/O | 0\2 | 0\3 |
| Gigabit Front (PMC) I/O | 1\1 | 1\2 |

Table 7-2. Ethernet Operating Mode

| Value | Explanation |
|--------------|--------------------|
| 10 H | 10M half-duplex |
| 10 F | 10M full-duplex |
| 100 H | 100M half-duplex |
| 100 F | 100M full-duplex |
| 1000 H | 1000M half-duplex |
| 1000 F | 1000M full-duplex |
| auto | auto sensing |

```
Host(config)# no interface Ethernet
```

Usage

```
no interface Ethernet <slot number>\<port number>
```

Parameter(s)

<slot number>

This is the interface slot number you want to configure.

<port number>

This is the physical port number you want to configure.

Example(s)

```
Host(config)# no interface Ethernet 0\1
```

Configuring Ethernet Redundancy

The system supports redundancy between 0\1 and 0\2 ethernet interface or between 1\1 or 1\2 Ethernet interface. The system, automatically while detects a malfunction in the working Ethernet (for example link down), switches to the protecting (redundant) Ethernet interface.

Using the port-ethernet redundancy-enable command

```
Host(config)# port-ethernet redundancy-enable
```

Use this command to configure Ethernet port redundancy.

Usage

```
port-ethernet redundancy-enable <working slot>\<working port> <protection slot>\<protection port> <non-revertive | revertive>
```

Parameter(s)

<working slot>

This is the working Ethernet interface slot number; valid values include 1 or 2.

<working port>

This is the working Ethernet interface physical port number; the value range is 1 to 3.

<protecting slot>

This is the protecting Ethernet interface slot number; valid values include 1 or 2.

<protecting port>

This is the protecting Ethernet interface physical port number; the value range is 1 to 3.

<non-revertive>

This configures non-revertive mode.

<revertive>

This configures revertive mode.

The slot number that you want to configure (see [Table 7-3](#)).

Table 7-3. Configure Ethernet ports

| Card Type | Slot\Port | Slot\Port 2 |
|-------------------------|-----------|-------------|
| Gigabit Front I/O | 0\1 | |
| Gigabit Rear I/O | 0\2 | |
| Gigabit Front (PMC) I/O | 1\1 | 1\2 |

Example(s)

```
Host (config) # port-ethernet redundancy-enable 1\1 1\2 non-revertive
```



Note: The system should refuse to enable Ethernet redundancy mode in case the protecting Ethernet interface is configured. It should indicate such kind of command failure to the system log and to the user interface (telnet/console).

```
Host(config) # port-ethernet redundancy-enable 1\1 1\2 non-revertive
Operation Error:
Protecting interface is configured
```



Note: The system should refuse to enable the Ethernet redundancy mode in case the protecting Ethernet interface is not the associated redundant interface (0/2 to 0/1 and 1/2 to 1/1). It should indicate such kind of command failure to the system log and to the user interface (telnet/console).

```
Host(config) # port-ethernet redundancy-enable 0\1 0\2
Operation Error:
Protecting interface is not the associated redundant interface
```

Using the no port-ethernet redundancy-enable command

```
Host(config)# no port-ethernet redundancy-enable
```

This command disables redundancy between the two Ethernet interfaces.

Usage

```
no port-ethernet redundancy-enable <working slot>\<working port> <protecting slot>
\<protecting port>
```

Parameter(s)

<working slot>

This is the working Ethernet interface slot; valid values include 0 or 1.

<working port>

This is the working Ethernet interface physical port; valid values include 1 or 2.

<protecting slot>

This is the protecting Ethernet interface slot; valid values include 0 or 1.

<protecting port>

This is the protecting Ethernet interface physical port; valid values include 1 or 2.

Example(s)

```
Host(config)# no port-ethernet redundancy-enable 1\1 1\2
```

LOOPBACK COMMANDS

Configuring interface loopback

```
Host(config)# interface loopback
```

This command enables the administrator to either add or change the loopback interface.



Note: The loopback interface can be part of the IP route lines/commands.

Usage

```
interface loopback <interface number> <IP address> <mask>
```

Parameter(s)

<interface number>

This is the interface number of the loopback; the valid range is 1 to 200.

<IP address>

This is the loopback IP address; it must be a valid IP address.

<mask>

This is the loopback mask address.

Example(s)

```
Host(config)# interface loopback 1 192.168.3.4 255.255.255.255
```



Note: When assigning the loopback interface an address that is in the subnet of one of the system interfaces the system should ignore the command and indicates the reason.

```
Host(config)# interface loopback 1 192.168.3.4 255.255.255.0
The loopback address is not valid.
```

Using the no interface loopback command

```
Host(config)# no interface loopback
```

This command deletes the loopback interface.

Usage

no interface loopback <interface number>

Parameter(s)

<interface number>

This is the loopback interface number to be deleted; valid range is 1 to 200.

Example(s)

```
Host(config)# no interface loopback 1
```



Note: The system should refuse to delete a loopback interface if an application source-interface is configured to the loopback.

VLAN COMMANDS

Configuring the VLAN

```
Host(config)# interface vlan
```

A virtual (or logical) LAN (called a VLAN) is a local area network with a definition that maps workstations on some basis other than geographic location (for example, by PVC, type of user, or primary application). VLANs are likely to be used with Gigabit Ethernet networks. The SG-1 supports virtual LAN as specified in the IEEE 802.1Q standard. It allows the LAN to be divided into several, disparate LANs and to serve each virtual LAN differently. The SG-1 handles both incoming and outgoing VLAN traffic and supports the full VLAN range (2 - 4095).

The main benefit of using a VLAN is it enables the support of multiple LANs in the same physical interface.

The following command also enables to configure VLAN on Gigabit Ethernet interface, and also supports QinQ encapsulation in accordance with IEEE 802.1Q



Note: The default name value is *NULL*.

Usage

```
interface vlan <slot number> \name <VLAN name> | QinQ <start ID><range>]
```

Parameter(s):

<slot number>

Slot number of the physical card.

<port number>

Port number of the card (1 or 2).

<vlan id>

VLAN interface identifier; valid number range is 1 to 4095.

<IP address>

IP address for the VLAN.

<mask>

Subnet mask for the VLAN.

[VLAN name]

It is a unique alphanumeric name assigned to the VLAN (it is 1 ÷ 64 alphabetic characters).

<start ID>

This is the start QinQ VLAN ID; valid number range is 2 to 4095.

<range>

This is the QinQ VLAN range; valid number range is 1 to 4095.

Example(s)

```
Host(config)# interface vlan 1\2\32 192.168.3.4 255.255.255.0 name ADC-R
Host(config)# interface vlan 1\2\233 24.36.1.12 255.255.255.0 name Kenvelo
Host(config)# interface vlan 3 1\1 192.168.1.1 255.255.255.0 QinQ 20 100
```



Note: The system should refuse the VLAN command when wrong slot\port parameters are configured.

```
Host(config)# vlan 3 1\3
Operation error
Wrong slot\port
```

Using no interface VLAN command

```
Host(config)# no interface vlan
```

This command deletes a VLAN definition for the Ethernet interface.

Usage

```
no interface vlan <slot number> \ <port number> < id>
```

Parameter(s)

<slot number>

This is the slot number of the physical card.

<port number>

This is the port number of the card (1 or 2).

<id>

This is the VLAN interface identifier; valid number range is 1 to 4095.

Example(s)

```
Host(config)# no vlan 3 1\1
```

AUTHENTICATION COMMANDS

Using the password pre-authentication command

```
Host(config)# password pre-authentication
```

This command defines the system pre-authentication password.

Usage

```
password pre-authentication <password>
```

For <password> options, see [Table 4-1, "Default Username/Password,"](#) on page 4-1.

Setting the default-service authentication mode

```
Host(config)# def-service-auth
```

When using authentication by username and password two protocols are available:

- **PAP** (Password Authentication Protocol)—the most basic form of authentication. In PAP, a user's name and password are transmitted over the network and compared to a table of name-password pairs. The main disadvantage of PAP is that both the username and password are transmitted without encryption.
- **CHAP** (Challenge Handshake Authentication Protocol)—a type of authentication in which the network server sends the client program a key to encrypt the username and password. This enables the username and password to be transmitted in encrypted form to protect them against eavesdroppers.

The **def-service-auth** command is used to set the type of authentication for PPP connections in the default step. Five options may be typed on the command line after this command, as shown in [Table 7-4](#).

Usage

```
def-service-auth no-service | ppp-auto | ppp-none | ppp-pap | ppp-chap | auto-select
```

Table 7-4. def-service-auth command parameters

| Parameter | Description |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| no-service | Disables authentication in the default step, even if authentication by caller ID is enabled. |
| ppp-none | Allows PPP access without authentication in the final step if authentication by caller ID was enabled. |
| ppp-pap | Authentication in the default step by the PAP authentication method, even if authentication by caller ID was enabled. |
| ppp-chap | Authentication in the default step by the CHAP authentication method, even if authentication by caller ID was enabled. |
| auto-select auth-retry on / off | Enables PPP authentication by PAP, CHAP or terminal + enables 3 more retries through an after dialing window if the user inserted wrong username or password (auth-retry on). |

Example(s):

```
Host(config)# def-service-auth ppp-auto
```

Changing domain authentication settings

```
Host(config)# domain-authentication
```

In a SG-1 system, virtual private tunnels (VPNs) are created upon RADIUS request. The tunneling service is always enabled within the SG-1 (there is no configuration command for turning it on or off). The domain-authentication configuration command is used to enable and disable authentication of the user's domain.

The domain-authentication separator command defines a list of characters that will be regarded as separators within usernames. The SG-1 can then extract the domain name by discarding the portion of the username before the separator. This command also enables the domain authentication process. The valid separators are !, @, #, \$, %, and -.

To define @ and # as separators and enable domain authentication, type:

```
domain-authentication separator @# (then press ENTER).
```

To turn off the domain authentication process, type:

```
no domain-authentication (then press ENTER).
```

```
Host(config)# authentication web-auth-method
```

- The authentication web-auth-method command defines the system authentication method to be used in WEB authentication (when authenticating a user via WEB authentication process).
- The system default value is PAP. When configured to its default values the system does not present it in write terminal command.
- The authentication methods are PAP or CHAP. The system should authenticate a WEB authenticated user based on this configuration. When CHAP is configured the system should process all necessary attributes for CHAP authentication (produce challenge, calculate the response based on the challenge and the password and communicate with the RADIUS as defined in the RFC).

Usage:

```
authentication web-auth-method [<PAP | CHAP>]
```

Parameter(s)

```
<PAP | CHAP>
```

Set the system WEB authentication method.

PAP – Authenticate the user using PAP

CHAP – Authenticate the user using CHAP

Example 1:

```
Host(config)# authentication web-auth-method CHAP <cr>
Host(config)# end <cr>
Host # write terminal.
authentication web-auth-method CHAP
...
```

The “no” command set the system web authentication mode to PAP.

```
Host(config)# no authentication web-auth-method
```

Example 2:

```
Host(config)# no authentication web-auth-method <cr>
Host(config)# end <cr>
Host # write terminal
...
```

ATM COMMANDS

This section describes the procedure for configuring an ATM PVC (Permanent Virtual Circuit). The following requirements must be met before a circuit can be created.

- The SONET/SDH port must be configured.
- The interface must be configured as a working port.
- The sub-interface must be type PPP.

Using the port sonet command

```
Host(config)# port sonet
```

Use this command to configure the ATM card port (it has two ports) to reflect whether the physical interface is SONET or SDH.



Note: Setting the interface type for one port will force the other port to the same type. This occurs because the second port is used for implementing Automatic Protection Switching (APS), where the redundant port must have the same interface type as the primary port.

Usage

```
port sonet <slot> \ <port> type (OC3c|STM1)
```

Parameter(s)

<slot>

The line card slot to be configured. Use the value 2 to indicate the configuration of the SONET/SDH port.

<port>

The line card port to be configured (either 1 or 2).

(OC3c|STM)

The type of physical interface:

- **OC3c**—the North American standard for 155.52 Mbps data over optical fiber.
- **STM**—the standard for 155.52 Mbps data over optical fiber outside of North America.

Example(s)

```
Host(config)# port sonet 2 \ 1 type OC3C
Updated item
Host(config)#
```

Configuring SONET/SDH port redundancy

```
Host(config)# port sonet redundancy-enable
```

SG1 systems use APS to switch ATM traffic from the main SONET/SDH channel (the working channel) to a secondary SONET/SDH channel (the protection channel on the same card) when a failure occurs. This redundancy enables service to continue despite failures on the working SONET/SDH channel. Refer to the *System Technology and Applications Overview* for an explanation of how an APS works.

Usage

```
port sonet redundancy-enable <working slot> \ <working port> \ <protect slot> \
<protect port> [SFBER <number>] [SDBER <number>]]
```

Parameter(s)

<working slot>

The slot number to be designated as the working slot (valid value = 2).

<working port>

The port number to be designated as the working port (valid value = 1 or 2).

<protect slot>

The slot number to be designated as the protect slot (valid value = 2).

<protect port>

The port number to be designated as the protect port (valid value = 1 or 2).

[SFBER (value)]

Signal Fail Bit Error Rate Threshold. Valid values = 3 to 12, Default = 3.

[SDBER (value)]

Signal Degrade Bit Error Rate Threshold. Valid values = 5 to 12, Default = 5.

Example(s)

```
Host(config)# port sonet redundancy-enable 2 \ 1 \ 2 \ 2 SFBER 3 SDBER 5
Updated item
Host(config)#
```

Configuring OC3 interfaces

```
Host(config)# interface
```

Usage

```
interface atm <slot> \ <port> <sub-interface> [type <ppp point-to -
multipoint|<routed point-to-point> ip <ip address> <mask>|<bridge-route point-to-
point>(ip <ip address><mask>|loopback <loopback number>)|mtu <(1500|1544)>]
```

Parameter(s)

<slot>

The value 2 is required to select the SONET/SDH interface.

<port>

Use value 1 or 2 to select port 1 or 2.

<sub-interface>

A logical interface number from 1 to 2000.

[type <ppp point-to-multipoint>]

The ppp represents PPP termination.

[type <routed point-to-point> ip <ip address><mask>]

The routed parameter implies routing capability only.

[type <bridge-route point-to-point>]

The bridge-route parameter implies bridge and routing capabilities.

<ip address>

IP address for the interface to be configured.

<mask>

Valid mask for the associated IP address.

[loopback]

Places a logical loopback on the bridge-route interface.

<loopback number>

Numeric value assigned to the bridge-route loopback.

[mtu]

Used to set the interface's Maximum Transmission Unit (MTU) (valid values are 1500 or 1544).

Example(s)

```
Host(config)# interface atm 2 \ 1 \ 1 type bridge-route point-to-point ip
192.168.10.1 255.255.255.0 mtu 1544

Added item

Host(config)#
```

Using the pppoa enable command

```
Host(config)# pppoa enable interface
```

This command enables PPPoA negotiation for a specific interface within the system.

Usage

pppoa enable interface <slot number>\<port number\[sub-interface number]>

Before enabling PPPoA, you must configure the correct port and the ATM interface.

Parameter(s)

Table 7-5. pppoa enable interface parameters

| Parameter | Description | Legal values/range |
|------------------------|-----------------------|--------------------|
| <slot> | Slot number. | 1 - 2 |
| <port number> | Physical port number. | 1 - 2 |
| <sub-interface number> | Sub-interface number. | 1- 2000 |

Example(s)

```
Host(config)# pppoa enable interface 0 \ 1
```

Using the no pppoa enable command

```
Host(Config)# no pppoa enable interface
```

The no pppoa enable interface command disables PPPoA negotiation for a specific interface in the system.

Usage

No pppoa enable interface <slot number>\<port number\[sub-interface number]>

For parameters see [Table 7-6](#).

Using the interface atm command

```
Host(config)# interface atm
```

The **interface atm** command defines the atm interfaces in the system.

- The command is accessible at the "configure terminal" menu.
- The command is only applicable with the SCC-ATM155 card.
- When MTU is configured to 1500, the **write terminal** command should not be used.
- IP address and loopback are not enabled for the PPP interface type.
- The routed type interface does not support unnumbered links.

Usage

```
interface atm <slot>\<port number>\<sub-interface number> [type <ppp point-to-mul-  
tipoint |  
routed point-to-point <ip <ip address> <mask> > |  
routed-bridge <point-to-point> <ip <ip address> <mask> > | loopback <loopback inter-  
face number>  
routed-bridge-ppp <point-to-point> <ip [ip address] <mask> > | loopback <loopback  
interface number>  
]  
[mtu <1500 | 1544>]
```

Table 7-6. interface atm command parameters

| Parameter | Description | Legal values/range |
|---------------------------------------------------------|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <slot > | PMC slot number | 1-2 |
| <port number> | SONET physical port number | 1 |
| <sub-interface number> | Sub-interface number | 1- 2000 |
| <ppp routed-bridge routed routed-bridge- ppp> | Sub-interface type | ppp - supporting PPPoE and PPPoA routed - routing capability only (IP over ATM) routed-bridge - bridge and routing capability (IP over Ethernet) routed-bridge-ppp - bridge and routing capability (PPP and IP over Ethernet simultaneously) (Default value: <i>ppp</i>) |
| point-to-point point-to-multipoint | Network topology type | point-to-point point-to-multipoint |
| <ip address> | Interface IP address | IP address |
| <mask> | Interface mask | Mask |
| <loopback interface number> | Loopback interface number | Number |
| <1500 1544> | MTU value | 1500, 1544 |

Example(s)

```
Host(config)# interface atm 2\1\1 <cr>
```

```
Host(config)# interface atm 2\1\1 type routed point-to-point ip 192.168.1.2  
255.255.255.0 <cr>
```

```
Host(config)# interface atm 2\1\1 type routed-bridge point-to-point loopback 1 <cr>
```

```
Host(config)# no interface atm
```

After this command is run, the system deletes the ATM interface, as well as all of the enabled applications for this interface (such as, PPPoE, PVC, PPPoA, etc.).

Usage

```
no interface atm <slot number>\<port number>\<sub-interface number>
```

For parameters and examples, refer to [Table 7-6](#).

Configuring a single PVC

```
Host(config)# atm pvc
```

This command creates a permanent virtual circuit (PVC) on an ATM interface.

Usage

```
atm pvc <vpi> <vci> <slot> <port> <sub-interface> [name <pvc name>] | [ OAM  
<on|off|on-cc> |ubr <peak cell rate>]
```

Parameter(s)

<vpi>

The virtual path identifier (valid values 0 - 255).

<vci>

The virtual channel identifier (valid values 32 - 2032).

<slot>

The value 2 is required to select the SONET/SDH interface.

<port>

The port on which the circuit is to be created (valid values 1 and 2).

<sub-interface>

A logical interface number that can be used to group circuits (valid values 1 - 2000).

[name <name>]

A 1 to 64 character alpha-numeric identifier that can be assigned to a circuit for tracking purposes.

[OAM]

Operation And Management mode

- **on** - Enable OAM mode
- **off** - Disable OAM mode
- **on-cc** - Enable OAM mode with continuity check

Example(s)

```
Host(config)# atm pvc 0 \ 60 \ 2 \ 1 \ 1
Added item
Host(config)#
```

Configuring a range of PVC's

```
Host(config)# atm pvc range
```

Use this command to configure a range of PVC's on an ATM interface.

Usage

atm pvc range <vpi> <start vci> <end vci> <slot> <port> <sub-interface> [**name** [pvc name>] | [OAM <on|off|on-cc> | ubr <peak cell rate>]

Parameter(s)

<vpi>

The virtual path identifier (valid values 0 - 255).

<start vci>

The virtual channel identifier (valid values 32 - 2032).

<end vci>

The virtual channel identifier (valid values 32 - 2032).

<slot>

The value 2 is required to select the SONET/SDH interface.

<port>

The port on which the circuit is to be created (valid values 1-2).

<sub-interface>

A logical interface number that can be used to group circuits (valid values 1 - 2000).

[pvc name]

A 1 to 64 character alpha-numeric identifier that can be assigned to a circuit for tracking purposes.

[OAM]

Operation And Management mode

- **on** - Enable OAM mode
- **off** - Disable OAM mode
- **on-cc** - Enable OAM mode with continuity check

Example(s)

```
Host(config)# atm pvc range 2 34 38 1 \ 2 \ 65 name arttt oam on ubr 120
```

RADIUS COMMANDS

Configuring the RADIUS server in the SG-1 configuration

```
Host(config)# radius-server
```

Each RADIUS server should be configured in the system. The radius-server command is used to configure the RADIUS server settings.

Usage

```
radius-server <host | key>
```

Parameter(s)

There are two options for the first command-line argument:

- **host**—to configure the RADIUS server host parameters (see “Configuring the RADIUS server host” below)
- **key**—to define an alphanumeric string, which serves as a password for communicating with the RADIUS server

Configuring the RADIUS server host

```
Host(config)# radius-server host
```

The host parameters include the IP address of the RADIUS server, the authentication port, the accounting port, and the type of RADIUS server. The default authentication port is 1645, and the default accounting port is 1646. The RADIUS server can be defined as one of three types.

- Merit
- Infovia
- Standard (default)

Usage

To set the host parameters of the RADIUS server:

```
radius-server host <IP address> auth-port <port address> acct-port <port number>  
<m,i, or s> priority <radius priority><cr>
```

To set the RADIUS server key:

```
radius-server key <alphanumeric string> <cr>
```

Parameter(s)

<port address>

The authentication port number can be any number between 0 and 65535. The default value is 1812.

<port number>

The accounting port number can be any number between 0 and 65535. The acct-port default value is 1813.

<m, i, or s>

m (Merit), i (Infovia) or s (Standard) are the three types of RADIUS server.

<radius priority>

The RADIUS priority can be any number from 1 to 5.



Note: From Version 4.2 and above, it is possible to define different authentication servers and different accounting servers.

Example(s)

```
Host(config)# radius-server host 12.3.22.63 auth-port 1865 acct-port 45 m priority 2
Host(config)# radius-server key aserwep
```

Configuring the RADIUS proxy for native IP

```
Host(config)# radius-proxy client
```

The proxy RADIUS capability is part of the Native IP support in the SG-1. It enables the SG-1 to serve users being terminated by different access devices (such as, AP or GGSN), and to enable for them the SG-1 services. The SG-1 acts as a RADIUS server to the access devices and proxy the RADIUS requests to its configured RADIUS server. This feature supports as well users using 802.1x dialers and SIM cards.

Usage

radius-proxy client <ip address> <mask> **key** <secret> [[**auth-port** <auth port-number>] | [**acct-port** <acct port-number>]]

Parameter(s)

Table 7-7. radius-proxy client parameters

| Parameter | Description | Values |
|--------------------|----------------------------|---------------------------------|
| <ip address> | Client radius IP address | Valid IP address |
| <mask> | Client radius mask | Valid mask |
| <secret> | Destined Radius secret key | 1 to 64 alphanumeric characters |
| <auth port-number> | Authentication port number | 0-65535 (Default is 1812) |
| <acct port-number> | Accounting port number | 0-65535 (Default is 1813) |

Example(s)

```
Host(config)# radius-proxy client 25.24.1.99 255.255.255.0 key poiplk auth-port 78
acct-port 589
```

Using the service cache command

```
Host(config)# service cache
```

The SG-1 is able to cache each received service's information based on a configured aging time. Operating this capability via the **service cache** command causes the system to authenticate a service once during the specified aging period. The caching key is the service name, while the maximum number of cached services in the system is 200. The system does not cache new services when its reached its caching limit. This is beneficial because it substantially reduces the number of RADIUS access request messages.

Usage

```
service cache off|[on [aging-time] < minutes>]
```

Parameter(s)

<minutes>

Aging-time in minutes (valid values are 1 - 9999; the default value is 10).

Example(s)

```
Host(config)# service cache on aging-time 32
```

Using the ip radius source-interface command

```
Host(config)# ip radius source-interface
```

This command is used to define the radius source interface. It enables configuration of the radius source-interface. Only one source interface may be defined.

The system default radius source interface type is "default," which indicates the primary interface. When configured to its default value, the **write terminal** command does not present the ip radius source interface command line.

Usage

```
ip radius source-interface [Ethernet <slot>\<port number> | ATM <slot>\<port number>\
< sub-interface number> | VLAN <slot>\<port number> \ <id> | Loopback <loopback
interface number> | session-interface | Default]
```

Parameter(s)**Table 7-8. ip radius source-interface parameters**

| Parameter | Description | Values |
|------------------------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <Interface type> | Interface type. | Optional Values: Ethernet, Loopback, ATM, and Default. Default value: Default Session-interface: it is the session incoming interface IP address. |
| <slot> | PMC slot number. | 0 - 2 |
| <port number> | SONET physical port number. | 1 - 2 |
| <sub-interface number> | Sub interface number. | 1- 2000 |

| Parameter | Description | Values |
|-----------------------------|----------------------------|---------|
| <Loopback Interface number> | Loopback interface number. | 1 - 200 |

Example(s)

```
Host(config)# ip radius source-interface Ethernet 0 \ 2
Host(config)# ip radius source-interface ATM 2 \ 1 \ 44
```



Note: The following command is used to configure the SG-1 to issue requests to the RADIUS via the loopback interface (the source address of the packet shall be SG-1's address).

```
Host(config)# ip radius source-interface loopback 1
```



Note: While assigning an undefined Interface to an application the system should ignore the command and indicates the reason.

```
Host(config)# ip radius source-interface loopback 4
Operation error
Interface is not defined
```

Using the no ip radius source-interface command

```
Host(config)# no ip radius source-interface
```

This command deletes the radius source-interface configuration by setting it to its default value.

Usage

no ip radius source-interface

Example(s)

```
Host(config)# no ip radius source-interface
```

ACCESS LIST COMMANDS

```
Host(config)# access list
```

An access list is a mechanism that filters the requests to the SG-1, by reading the source IP address and its net mask. This mechanism allows you to grant certain workstations access to a SG-1 for administrative purposes. The access list command is used to create and modify access lists.

Usage

To allow SNMP access, at the second-level command prompt type one of the following (see "Parameter(s)" below for details):

```
access-list Telnet-permit <source IP address> <cr>
access-list SNMP-permit <source IP address> <mask> <cr>
```

```
access-list EDS-permit <source IP address> <mask> <cr>
access-list native-ip <source IP address> <mask> <cr>
native-ip-pass-through <source IP address> <mask> <cr>
```

After keying in one of the above commands, the system responds by displaying `Added item.`

Parameter(s)

There are two options for the first command line argument following the access list command:

- **Telnet-permit** - to allow Telnet access to a SG-1 from a certain IP address. If no Telnet access list exists, the IP address is added to it. This command must be followed by the IP address.
- **SNMP-permit**—to allow SNMP (Simple Network Management Protocol) access to a SG-1 from a certain IP address. If no SNMP access list exists, the IP address is added to it. This command must be followed by the source IP address and mask.
- **EDS-permit**—to allow access to a SG-1 from a certain IP address. If no EDS-permit access list exists, the IP address is added to it. This command must be followed by the source IP address and mask.
- **native-ip**—to allow native-ip access to a SG-1 from a certain IP address. If no nativ-ip access list exists, the IP address is added to it. This command must be followed by the source IP address and mask.
- **native-ip-pass-through**—to allow native-ip pass trough the SG-1 from a certain IP address. If no native-ip-pass-through access list exists, the IP address is added to it. This command must be followed by the source IP address and mask.

Example(s)

```
Host(config)# access-list SNMP-permit 25.3.2.251 255.255.255.255
```

Using the access-list native-ip command

```
Host(config)# access-list native-ip
```

The access-list native-ip command defines the allowed static native IP networks in the system. It defines the allowed static native IP networks as native-ip potential users. The maximum number of access list line is 200.

Usage

```
access-list native-ip <source ip address> <source mask>
```

Parameter(s)

<source ip address>

This is the allowed network source IP; it must be a legal network IP address.

<source mask>

This is the allowed network source mask; it must be a legal network IP address.

Example(s)

```
Host(config)# access-list native-ip 192.168.1.0 255.255.255.0
```

Using the no access-list native-ip command

```
Host(config)# no access-list native-ip
```

This command deletes the native IP access list configuration. It deletes the static native-ip access list configuration.

Usage

no access-list native-ip <source ip address>

Parameter(s)

<source ip address>

This is the allowed network source IP; it must be a legal network IP address.

Example(s)

```
Host(config)# no access-list native-ip 192.168.1.0
```

Using the access-list native-ip-pass-through command

```
Host(config)# access-list native-ip-pass-through
```

The **access-list native-ip-pass-through** command defines the source IP addresses arrived from a native-ip pipe that should not pass through the standard native ip path. Those IP addresses should not be considered as arriving from a native-ip pipe. This access-list is precedent to the native-ip access-list. The maximum number of access list line is 200.

Usage

access-list native-ip-pass-through <source ip address> <source mask>

Parameter(s)

<source ip address>

This is the allowed network source IP; it must be a legal network IP address.

<source mask>

This is the allowed network source mask; it must be a legal network IP address.

Example(s)



Note: In the following example the whole network 192.168.1.0 is configured as native-ip potential users except for 192.168.1.1, which is probably a server in this network and not a potential user.

```
Host(config)# access-list native-ip 192.168.1.0 255.255.255.0
Host(config)# access-list native-ip-pass-through 192.168.1.1 255.255.255.255
```



Note: In order to activate the **native-ip-pass-through**, IP forward command should be defined.

Using the no access-list native-ip-pass-through command

```
Host(config)# no access-list native-ip-pass-through
```

The **no access-list native-ip-path-through** command deletes a path through native IP access list configuration. It deletes the static native-ip access list configuration.

Usage

no access-list native-ip-pass-through <source ip address>

Parameter(s)

<source ip address>

This is the allowed network source IP; it must be a legal network IP address.

Example(s)

```
Host(config)# no access-list native-ip-pass-through 192.168.1.1
```

SNMP COMMANDS

Using the SNMP-server community

```
Host(config)# SNMP-server community
```

The SNMP-server command is used to modify the settings for the SNMP server.

Usage

SNMP-server community <get | set> <Alpha numeric string>

SNMP-server host <ip> <Alpha numeric string>

SNMP-server group <aaa | network | security> <ip>

Parameter(s)

When using the community parameter, there are two options for the first command line argument:

- **get**—to define the get SNMP community key
- **set**—to define the set SNMP community key

When using the group parameter, there are three options for the first command line argument:

- **aaa**—Major alarm. Any defined RADIUS server marked in the system DB as dead (inactive) will be reported via SNMP. Any defined RADIUS server marked in the system as ALIVE after being DEAD sends trap indication and log indication.
- **network**—Major alarm. Occurs when Ethernet link is down. When the Ethernet connection becomes available, system sends a trap indication only in a clear event and sends a log indication.
- **security**—Warning Alarm. First time that IP spoofing is detected, the system can manually send a trap indication/log indication.



Note: The SNMP key for get and set is encrypted.

Using the SNMP-server trap-source-int

```
Host(config)# SNMP-server trap-source-int
```

This command defines the source of the snmp traps. The system default snmp trap source interface type is "Default," which indicates the primary interface is in use.

Usage

snmp-server trap-source-int [[Ethernet <slot>\<port number> | **ATM** <slot>\<port number>\<sub-interface number> | **VLAN** <slot>\<port number>\<sub-interface number> | [**loopback** <loopback interface number> | **Default**]

Parameters:

| Parameter | Description | Values |
|-----------------------------|-----------------------------|---------------------------------------------------------------------------------|
| Interface type | | Optional Values: Ethernet, Loopback, ATM, VLAN, Default. Default value: Default |
| <slot> | PMC slot number. | 0 – 2 |
| <port number> | SONET physical port number. | 1 – 2 |
| <sub-interface number> | Sub interface number. | 1 – 4095 |
| <Loopback Interface number> | Loopback interface number. | 1 – 200 |

Example(s)

```
Host(config)# SNMP-server trap-source-int Ethernet 0 \ 1
Host(config)# SNMP-server trap-source-int ATM 2 \ 1 \ 34
Host(config)# SNMP-server trap-source-int Loopback 12
Host(config)# SNMP-server trap-source-int VLAN 1 \ 1 \ 12
Host(config)# SNMP-server trap-source-int Default
```

TUNNEL COMMANDS

```
Host(config)# interface tunnel
```

This command defines a remote (the tunnel initiator) tunnel endpoint IP address, which allowed opening ip-in-ip or GRE tunnels to the system.

The local tunnel endpoint ip address (tunnel destination) should be one of the system IP addresses (Ethernet, VLAN, Loopback, or ATM).

The system should refuse the **interface tunnel** command when a local tunnel endpoint IP address is not one of the configured system IP addresses (i.e., interface or loopback).

Usage

interface tunnel <interface number> <remote tunnel endpoint ip address> <local tunnel endpoint ip address> <tunnel type>[<tunnel mode>]

Parameter(s)

<interface number>

This is the tunnel interface number; valid number range is 1 to 500.

<remote tunnel endpoint ip address>

This is the remote tunnel endpoint IP address; it must be a system legal IP address.

<local tunnel endpoint ip address>

This is the local tunnel endpoint IP address; it must be a system legal IP address which is one of the system interfaces (Ethernet, ATM, or Loopback).

<tunnel type>

This is the system allowed tunnel types: ip-in-ip or GRE.

<tunnel mode>

This is the tunnel mode, the default is Standard (roamed).

Example 1

```
Host(config)# interface tunnel 1 192.168.0.1 10.0.1.7 ip-in-ip
Operation error:
Local tunnel endpoint IP address should be on of the system interfaces
```

Example 2

```
Host(config)# interface tunnel 1 192.168.1.8 10.0.1.7 gre roamed <cr>
Host(config)# native-ip enable interface tunnel 1 <cr>
Host(config)# end <cr>
Host# write terminal
...
interface tunnel 1 192.168.1.8 10.0.1.7 gre roamed
native-ip enable interface tunnel 1
...
```



Note: The system should refuse changing a tunnel interface parameter that is already carrying user data.

```
Host(config)# interface tunnel 1 192.168.0.1 10.0.1.7 ip-in-ip
Operation error:
The tunnel interface parameters cannot be changed since users already using it. You
should delete it first
```

The tunnel interface should appear in the ifTable with type tunnel (131) and should include the standard interface information.

The maximum number of tunnel interfaces in the system is 500.

The system should add the tunnel information to the GRE or ip-in-ip tables as a local tunnel endpoint. The total sessions field should present *NA* (Not applicable) in such interfaces that the native-ip enabled was not configured.

The **interface** command creates lines in the IP tunnel interfaces table.

Setting the no interface tunnel command

```
Host(config)# no interface tunnel
```

The **no interface tunnel** command deletes the interface tunnel and all related tunnel interface configuration from the GRE and ip-in-ip tables.

Usage

no interface tunnel <interface number>

Parameter(s)

<interface number>

This is the system interface number; valid number range is 1 to 500.

Example(s)

```
Host(config)# no interface tunnel 1
```



Note: The system should refuse the **interface tunnel** command when an existing interface already includes the same remote tunnel endpoint IP address, local tunnel IP address, and tunnel type.

```
Host(config)# interface tunnel 1 192.168.0.1 10.0.1.7 ip-in-ip
Operation error:
A tunnel interface with the same parameters already configured
```

Using the ip tunnel echo command

```
Host(config)# ip tunnel echo
```

This command defines the source IP address to be used for the ICMP echo message in the redundancy operation.

Usage

```
ip tunnel echo <echo source IP address>
```

Parameter(s)

```
<echo source ip address>
```

It is the source IP address to be used in the ICMP (ping) echo message for tunnel redundancy.

TIMEOUTS COMMANDS

Timeouts are used to stop activities if the SG-1 recognizes a problem with data transfer. session-timeout command

Setting the Session-Timeout

The session timeout is used to disconnect a user after a specified number of seconds.

Usage

session-timeout <number>

Parameter

<number>

It is the seconds number, session timeout in seconds 0 - 4250000 (0 = unlimited).

Example

To set the session timeout to 64800 seconds:

```
Host(config)# session-timeout 64800 <cr>
```

Setting the Idle-Timeout

The idle timeout is used to disconnect a connection without any traffic from the user to the network (upstream). The user will be disconnected if the SG-1 doesn't detect any data traffic after the specified number of seconds.

idle-timeout command

Usage

idle-timeout <number>

Parameter

<number>

It is the seconds number, idle timeout in seconds 0 - 4250000 (0 = unlimited).

Example

To set the idle timeout of 1800 seconds:

```
Host(config)# idle-timeout 1800 <cr>
```

NATIVE IP COMMANDS

Using the `native-ip dhcp pre-auth-mode` command

```
Host(config)# native-ip dhcp pre-auth-mode
```

The system should enable upon configuration to pre-authenticate a Native IP session, which uses DHCP (DHCP discover) for IP allocation based on its MAC address. A successful pre-authentication will forward the DHCP discover message to the DHCP server.

This command defines the system pre-authentication method in a DHCP request for native IP tunnels. The system default pre-authentication value is *none*. When the system **native-ip dhcp pre-auth-mode** is configured to *none* it should not appear via the write terminal command.

Usage

```
native-ip dhcp pre-auth-mode <mode> <none | MAC | reject-MAC>
```

Parameter(s)

<mode>

Pre authentication mode. Valid values are: none, MAC, or reject-MAC:

- **none**—No pre-authentication mode.
- **MAC**—Auth by MAC (on auth-reject, grant default service).
- **reject-MAC**—Auth by MAC (on auth-accept, grant default service).

Example(s):

```
Host(config)# native-ip dhcp pre-auth-mode none
Host(config)# native-ip dhcp pre-auth-mode mac
Host(config)# native-ip dhcp pre-auth-mode reject-MAC
```

Using `native-ip def-service-auth` command

```
Host(config)# native-ip def-service-auth
```

The **native-ip def-service-auth** command instructs that the default native IP service be granted to the connected native IP sessions. It defines the system default native-ip service name.

The default native IP service name is "GNativeIP". When configured to its default value, the **write terminal** command does not present the command line.

Usage

```
native-ip def-service-auth [<service name>]
```

Parameter(s)

<service name>

Default native IP service name. Alpha-numeric string of up to 32 characters.

Example(s)

```
Host(config)# native-ip def-service-auth werbcvxsaq
```

Using the native-ip enable command

```
Host(config)# native-ip enable interface
```

This command enables native ip service on a specific interface.

It enables the native IP for a specific VLAN or interface.

Phase 1 should include Ethernet interface and VLAN-Id only.

Phase 2 should include interface atm.

Usage

```
native-ip enable interface [<Ethernet<slot\port> | VLAN <slot\port\identifier> |  
ATM <slot\port\sub-interface> | tunne <tunnel interface id> | l2transport <destina-  
tion> <vc-id>
```

Parameter(s)

<slot\port>

This is the Ethernet slot number (valid number range is 0 to 2) and Ethernet port number (1 or 2).

<slot\port\identifier>

This is the Ethernet slot number (valid numbers 0 to 2), Ethernet port number (1 or 2), and VLAN identifier (1 to ?).

<slot\port\sub-interface>

This is the Ethernet slot number (valid number range is 0 to 2), Ethernet port number (1 or 2), and VLAN identifier (sub-interface valid number 1 to 4095).

<tunnel interface id>

This is the tunnel interface identifier. Valid number range is 1 to 300.

<destination>

This specifies the LDP IP address of the remote PE. It must be a valid IP address.

<vc-id>

This assigns a VC ID to the virtual circuit between the system and the remote PE.

Example(s)

```
Host(config)# native-ip enable interface Ethernet 0 \ 1
Host(config)# native-ip enable interface ATM 2 \ 2 \ 43
Host(config)# native-ip enable interface VLAN 0 \ 1 \ 45
```

The system should disable the native-ip enable command in case Native IP capability is not licensed (set to *off*). The system should report an operation error with the following message format:

```
Host(config)# native-ip enable interface 0\2
Operation Error:
License is required
```

Using the no native-ip enable command

```
Host(config)# no native-ip enable interface
```

It deletes a native ip enabled configuration.

Usage

no native-ip enable interface [**Ethernet**<slot\port> | **VLAN** <slot\port\identifier> | **ATM** <slot\port\sub-interface> | **tunnel** <tunnel interface id> | **l2transport** <destination> <vc-id>

Parameter(s)

<slot\port>

This is the Ethernet slot number (valid number range is 0 to 2). This is the Ethernet port number (1 or 2).

<slot\port\identifier>

This is the Ethernet slot number (valid number range is 0 to 2). This is the Ethernet port number (1 or 2), and VLAN identifier (valid values are 1 to ?).

<slot\port\sub-interface>

This is the Ethernet slot number (valid number range is 0 to 2), Ethernet port number (1 or 2), and VLAN identifier, sub-interface (valid number range is 1 to 4095).

<tunnel interface id>

This is the tunnel interface identifier (valid number range is 1 to 300).

<destination>

This specifies the LDP IP address of the remote PE; it must be a valid IP address.

<vc-id>

This is used to assign a VC ID to the virtual circuit between the system and the remote PE.

```
Host(config)# no native-ip enable VLAN 1\1\9
```

Using native-ip realm command

```
Host(config)# native-ip realm
```

The **native-ip realm** command specifies the realm string the system should use in the native-ip authenticating, accounting, and service operations.

The realm should be added to the user-name field in all native-ip user's authentication and accounting radius messages, except for web-authentication, regardless of its native-ip type (DHCP, Proxy radius, Plain IP).

The realm string should include the realm separator (such as, the @ symbol).

Usage

```
native-ip realm <realm string>
```

Parameter(s)

<realm string>

This is the native-ip realm string; it is an alpha-numeric string of up to 32 characters.

Example(s)

```
Host(config)# native-ip realm @nipnew
```

Using no native-ip realm command

```
Host(config)# no native-ip realm
```

The **no native-ip realm** command disables the native IP realm operation. The no operation should set the native IP realm value to *null*.

Usage

```
no native-ip realm
```

Example(s)

```
Host(config)# no native-ip realm
```

Using native-ip roaming command

```
Host(config)# native-ip roaming
```

The native-ip roaming command enables seamless roaming of users between SCCs. It enables the native-ip roaming between SCCs share the same subnet.

Usage

```
native-ip roaming [interface <Ethernet <slot\port> | VLAN <slot\port\sub-interface>  
l2transport <destination> <vc-id>]
```

Parameter(s)

<slot\port>

The Ethernet slot number (valid number range is 0 to 2). The Ethernet port number (1 or 2).

<slot\port\sub-interface>

The Ethernet slot number (valid number range is 0 to 2), The Ethernet port number (1 or 2), and VLAN identifier, sub-interface (valid number range is 1 to 4095).

<destination>

This specifies the LDP IP address of the remote PE; it must be a valid IP address.

<vc-id>

This is used to assign a VC ID to the virtual circuit between the system and the remote PE. Valid number range 1 – (2³² – 1).

Example 1:

```
Host(config)# native-ip roaming
```

Example 2:

```
Host(config)# native-ip roaming interface Ethernet 0/2
```

```
Host(config)# no native-ip roaming
```

The native-ip roaming command enables seamless roaming of users between SCCs. It enables the native-ip roaming between SCCs share the same subnet.

Usage

No native-ip roaming

Example:

```
Host(config)# no native-ip roaming
```

MAXIMUM SEGMENT SIZE (MSS) CHANGING

Using the `ip tcp adjust-mss` command

```
Host(config)# ip tcp adjust-mss
```

The system, when configured so that `ip tcp adjust-mss` is set to *on*, should adjust the TCP MSS option value on SYN packets to 1436 (for MSS option larger than 1436) in both directions for each connected user. (**Note:** When the mtu is configured to 1544, there is no need to adjust the mss.)

- The system should extract 64 from the MTU=1500 because of the TCP header (20 bytes).
- This command configures the system to work in `adjust-mss` mode.
- The system default `adjust-mss` value is *on*.
- When the system `adjust-mss` is configured to *off* it should not appear in the **write terminal** command.

Usage

```
ip tcp adjust-mss <on | off>
```

Example(s)

```
Host(config)# ip tcp adjust-mss on
```

Term Descriptions

- **MTU:** The maximum transmission unit is a link layer restriction on the maximum number of bytes of data in a single transmission.
- **Path MTU:** The smallest MTU of any link on the current path between two hosts. This may change over time since the route between two hosts, especially on the Internet, may change over time. It is not necessarily symmetric and can even vary for different types of traffic from the same host.
- **MSS:** The MSS is the maximum segment size. It can be announced during the establishment of a TCP connection to indicate to the other end the largest amount of data in one packet that should be sent by the remote system. Normally the packet generated will be 40 bytes larger than this: 20 bytes for the IP header and 20 for the TCP header. Most systems announce an MSS determined from the MTU on the interface that traffic to the remote system passes out through.

L2TP AND PPP COMMANDS

The L2TP commands are used for configuring the L2tp source-address. When the l2tp source-address is configured, the system sets the l2tp source address in the response packets regardless of the original l2tp LAC request.

Only one source interface may be defined.

Using the ip l2tp source-address command

```
Host(config)# ip l2tp source-address
```

This command deletes the radius source-interface configuration by setting it to its default value.

Usage

```
ip l2tp source-address <IP address>
```

Parameter(s)

<IP address>

This is the l2tp source IP address; it must be a legal IP address.

Example(s)

```
Host(config)# ip l2tp source-address 192.168.1.2
```

Using the no ip l2tp source-address command

```
Host(config)# no ip l2tp source-address
```

This command deletes the L2TP source-address configuration.

Usage

```
no ip l2tp source-address
```

Example(s)

```
Host(config)# no ip l2tp source-address
```

Defining the domain name

```
Host(config)# ip domain-name
```

Usage

```
ip domain-name <valid domain name>
```

Example(s)

```
Host(config)# ip domain-name My-Company
```


Configuring the primary DNS

```
Host(config)# ip primary-name-server
```

Usage

```
ip primary-name-server <IP address>
```

Example(s)

```
Host(config)# ip primary-name-server 55.12.1.24
```

Configuring the secondary DNS

```
Host(config)# ip secondary-name-server
```

Usage

```
ip secondary-name-server <IP address>
```

Example(s)

```
Host(config)# ip secondary-name-server 55.12.1.24
```

Setting tunnel servers

```
Host(config)# tunnel-server
```

Each IAS (Internet Access Switch) which will send its users to the SG-1 must be configured in the SG-1 (or several IAS which are in the same Mask and use the same tunnel password). The **tunnel-server host** command is used to configure the tunnel server (IAS) settings.

Usage

```
tunnel-server host <IP address> mask <mask> [password <alpha-numeric string>]
```

After keying in the above string, the tunnel password is encrypted.

Parameter(s)

<IP address>

The LAC machine IP address.

<mask>

The mask of the network allowed connecting the LNS.

<alpha-numeric string>

Password used for authenticating between LAC and LNS.

Example(s)

```
Host(config)# tunnel-server host 12.25.3.15 mask 255.255.255.0 password rewed
```

Setting multi-link mode

```
Host(config)# multilink-mode
```

The SG-1 allows the user to use higher bandwidth by using ML-PPP. Typically ML-PPP is used to combine the speed of a few ISDN BRI B-Channels to get 128 Kbps or other N x 64 Kbps of virtual bandwidth, depending on the RADIUS database and the SG-1 configuration.

Usage

multilink-mode <multi-cage | single-cage | none>

Parameter(s)

This command has the following options:

- **multi-cage**—ML-PPP uses PPP projection to call master method. The port limit is received from the RADIUS. If no port limit is defined, the system default is 1 (according to the RFC). This means 64 Kbps is the maximum bandwidth (none mode behavior). This is for a scenario of various SG-1SCC cards in a POP.
- **single-cage**—ML-PPP uses PPP projection to call master method. The port limit is received from the RADIUS. If no port limit is defined, the system default is 1 (according to the RFC). This means 64 Kbps is the maximum bandwidth (none mode behavior). The difference with the Multi-Cage option is the efficiency of the bundling (in this scenario there is only one SCC card).
- **none**—no Multi link operation is allowed in the SG-1. All the PPP links port limit definitions are 1. This means the MLP protocol negotiation is allowed (MLP user), but only one link is available.

Defining an address pool

```
Host(config)# ip local-pool
```

The **ip local pool** command allows you to define IP pools for all the incoming sessions. There are three required command line arguments—the pool name, the starting IP address in the pool, and the last IP address in the pool.

Usage

ip local-pool <pool_name> <starting_IP> <last_IP> <internal | external>

Parameter(s)

<pool_name>

The IP pool name (an alpha-numeric string).

<starting_IP>

Starting IP address.

<last_IP>

Ending IP address.

<internal | external>

This fourth command-line argument is optional and has two options:

- **internal**—For internal use only.
- **external**—Only when specifically requested. For users receiving their IP addresses from the RADIUS using the Token-Pool RADIUS attribute.

Example(s)

```
Host(config)# ip local-pool pool1 12.2.3.56 12.2.3.240
```

Using the lcp renegotiate command

```
Host(config)# lcp renegotiate
```

In some cases, there is a need to renegotiate the LCP with all the connected peers. The **lcp renegotiate** command enables this request.

Usage

lcp renegotiate (or **no lcp renegotiate**)

Term Description

LCP: In the Point-to-Point Protocol (PPP), the Link Control Protocol (LCP) establishes, configures, and tests data-link Internet connections. Before establishing communications over a point-to-point link, each end of the PPP link must send out LCP packets. The LCP packet either accepts or rejects the identity of its linked peer, agrees upon packet size limits, and looks for common misconfiguration errors. Basically, the LCP packet checks the telephone line connection to see whether the connection is good enough to sustain data transmission at the intended rate. Once the LCP packet accepts the link, traffic can be transported on the network; if the LCP packet determines the link is not functioning properly, it terminates the link.

LCP packets are divided into three classes:

- Link configuration packets used to establish and configure a link
- Link termination packets used to terminate a link
- Link maintenance packets used to manage and debug a link

Using the lcp echo command

```
Host(config)# lcp echo
```

This command configures the LCP echo behavior in all PPP sessions.

- The system default LCP configuration is off.
- The system LCP echo retries value is 3.
- The lcp on default is 30 seconds. In this mode the LCP echo is active for all PPP sessions.

Usage

```
lcp echo [on [<lcp echo period>] | off]
```

Parameter(s)

[on]

This sets lcp echo to active.

[off]

This sets lcp echo to inactive.

<lcp echo period>

This is the lcp echo period in seconds; valid number range is 2 to 32767.

Example(s)

```
Host(config)# lcp echo on 89
```

Using the service internal command

```
Host(config)# service internal
```

The system default internal service is Framed-PPP. When configured to the default value, write terminal does not present the configuration line.

Usage

```
service internal <Framed-PPP | VPDN <tunnel id> <l2tp tunnel password> <Primary IP address> [Secondary IP address]>
```

Parameter(s)

Table 7-9. service internal parameters

| Parameter | Description | Legal values / range |
|------------------------|----------------------------------------------------------|--------------------------------------------|
| <Tunnel id> | The tunnel ID. | Alpha numeric string (up to 64 characters) |
| <L2tp tunnel password> | The password used to authenticate to a remote server. | Alpha numeric string (up to 64 characters) |
| <Primary IP address> | The internal service line definition, as defined in VSA. | IP address |
| <Secondary IP address> | The internal service line definition, as defined in VSA. | IP address |

Example(s)

```
Host(config)# service internal vpdn oipoy bbbdfe 9.22.4.12 10.9.8.33
```

PPPoE support

SG-1 PPP over Ethernet (PPPoE) support enables multiple hosts at a remote site to connect through the same customer premise access device. It also provides access control and billing functionality in a manner similar to dial-up services using PPP. In many access technologies, the most cost effective method to attach multiple hosts to the customer premise access device, is via Ethernet. In addition, it is desirable to keep the cost of this device as low as possible while requiring little or no configuration.

This feature of the SG-1, provides the ability to connect a network of hosts over a simple bridging access device to a remote Access Concentrator. With this model, each host utilizes its own PPP stack and the user is presented with a familiar user interface. Access control, billing and type of service can be done on a per-user, rather than a per-site, basis. To provide a point-to-point connection over Ethernet, each PPP session must learn the Ethernet address of the remote peer, as well as establish a unique session identifier. The PPPoE protocol includes a discovery protocol that provides this.

The main aim of this feature is to provide ISPs with xDSL support with an easy consumer end adoption. This model is preserving the point-to point session, which is familiar to both the end-user and to the ISP.

A PPPoE session establishment performs the following scenario:

- Discovery stage:
 - Host broadcasts a PPPoE Active Discovery Initiation (PADI) packet.
 - When SG-1 receives a PADI, it replies by sending a PPPoE Active Discovery Offer (PADO) packet to the host.
 - The host then sends a single PPPoE Active Discovery Request (PADR) packet to the SG-1.
 - When the SG-1 receives a PADR packet, it prepares to begin a PPP session. It generates a unique SESSION_ID for the PPPoE session and replies to the host with a PPPoE Active Discovery Session-confirmation (PADS) packet
 - PPP session stage

Usage Scenarios

1. SG-1 as xDSL aggregator using ATM network:

In this scenario, a PPP session is initiated on an Ethernet-connected client through a standard ADSL modem. The session is transported over the ATM DSL link via RFC 1483 Ethernet-bridged frames and is terminated by the SG-1 which is acting as an xDSL aggregator (using the SCC-ETH ATM card).

Scenario highlights:

- PVCs (Permanent Virtual Connection) are defined between the DSLAM and the SG-1.
- The SG-1 terminates the PPPoE sessions initiated at users PCs and grants a service.
- User packets are passed as follows:
 - The user PC, which runs PPP client, encapsulates the PPP packet with Ethernet.
 - The PPPoE packet is being sent from the user PC to the xDSL modem.
 - The xDSL modem encapsulates it with DSL.
 - The DSLAM de-encapsulates the DSL ATM switches to pre-configured PVC to the SG-1.
 - The SG-1 de-encapsulates the ATM de-encapsulates the Ethernet and terminate the PPP session.

2. SG-1 as xDSL aggregator using Gigabit Ethernet network

In this scenario, a PPP session is initiated on an Ethernet-connected client through a standard ADSL modem. The session is transported over the Ethernet and terminated by the SG-1, which is acting as an xDSL aggregator.

Scenario highlights:

- The SG-1 terminates the PPPoE sessions initiated at users PCs and grants a service.
- User packets are passed as follows:
 - The user PC, which runs PPP client, encapsulates the PPP packet with Ethernet.
 - The PPPoE packet is being sent from the user PC to the xDSL modem.
 - The xDSL modem encapsulates it with DSL.
 - The DSLAM de-encapsulates the DSL header and send the Ethernet packets including broadcasts.
 - The SG-1 de-encapsulates the Ethernet and terminate the PPP session.

Using the pppoe enable command

```
Host(config)# pppoe enable interface
```

The pppoe enable interface command enables PPPoE negotiation for a specific interface in the system.

Usage

```
pppoe enable interface <slot number>\<port number\[sub-interface number]>
```

Parameter(s)

Table 7-10. pppoe enable parameters

| Parameter | Description | Legal values/range |
|------------------------|-----------------------|--------------------|
| <slot number> | Slot number. | 0 - 2 |
| <port number> | Physical port number. | 0 - 2 |
| <sub-interface number> | Sub interface number. | 1- 2000 |

Example(s)

```
Host(config)# pppoa enable interface 0 \ 1
```

Using the no pppoe enable command

```
Host(config)# no pppoe enable interface
```

The no pppoe enable interface command disables PPPoE negotiation for a specific interface in the system.

Usage

```
no pppoe enable interface <slot number>\<port number\[sub-interface number]>
```

For parameters and examples, refer to “Using the pppoe enable command”.

DHCP COMMANDS

Dynamic Host Configuration Protocol (DHCP) is a communications protocol that lets network administrators manage centrally and automate the assignment of Internet Protocol (IP) addresses in an organization's network. Using the Internet Protocol, each machine that can connect to the Internet needs a unique IP address. When an organization sets up its computer users with a connection to the Internet, an IP address must be assigned to each machine. Without DHCP, the IP address must be entered manually at each computer and, if computers move to another location in another part of the network, a new IP address must be entered. DHCP lets a network administrator supervise and distribute IP addresses from a central point and automatically sends a new IP address when a computer is plugged into a different place in the network.

The SG-1 enables in this feature the DHCP relay capabilities. It acts as a DHCP relay agent and forwards the DHCP request to a DHCP server, which handles the IP-addresses allocation for the connected peers.

Using the ip dhcp relay server command

```
Host(config)# ip dhcp relay server
```

Use this command to define the DHCP relay server.

Usage

```
ip dhcp relay server <Ethernet | VLAN | ATM | default> <slot>\<port number>[\<sub-
interface number>] <priority> <ip address>
```

Parameter(s)

Table 7-11. ip dhcp relay server parameters

| Parameter | Description | Legal values/range |
|------------------------|--------------------------------------|--------------------|
| <port number> | Physical port number. | 1 - 2 |
| <sub-interface number> | Sub interface number. | 1- 4095 |
| <priority> | DHCP server priority. | 1- 10 |
| <ip address> | IP address of the DHCP relay server. | IP address |

Example(s)

```
Host(config)# ip dhcp relay server Ethernet 0 \ 1 7 12.1.44.55
Host(config)# ip dhcp relay server Atm 2 \ 1 \ 54 5 12.33.54.1
Host(config)# ip dhcp relay server VLAN 1 \ 2 32 4 1.3.2.4
```

Using the no ip dhcp relay server command

```
Host(config)# no ip dhcp relay server
```

Use this command to define DHCP relay servers.

Usage

```
ip dhcp relay server <Ethernet | VLAN | ATM | default> <slot>\<port number>[\<sub-
interface number>] <priority>
```

For parameters and examples, see "Using the ip dhcp relay server command".

Using the ip dhcp relay information option command

```
Host(config)# ip dhcp relay information option
```

The ip dhcp relay information option command enables the system to insert a DHCP relay agent information option in forwarded BOOT REQUEST messages to the DHCP server.

Usage

```
ip dhcp relay information option <Ethernet | VLAN | ATM | default> <slot>\<port number>\[sub-interface number] <priority> <ATM | WLAN>
```

Parameter(s)

Table 7-12. ip dhcp relay information parameters

| Parameter | Description | Legal values / range |
|------------------------|-----------------------|----------------------|
| <port number> | Physical port number. | 1 - 2 |
| <sub-interface number> | Sub interface number. | 1- 4095 |
| <priority> | DHCP server priority. | 1- 10 |

Example(s)

```
Host(config)# no ip dhcp relay server Ethernet 0 \ 1 2
Host(config)# no ip dhcp relay server Atm 2 \ 1 \ 33 5
Host(config)# no ip dhcp relay server VLAN 1 \ 2 3 5
Host(config)# no ip dhcp relay server Default 0 \ 1 \ 4 8
```

Using the no ip dhcp relay information option command

```
Host(config)# no ip dhcp relay information option
```

Usage

```
no ip dhcp relay information option <Ethernet | VLAN | ATM | default> <slot>\<port number>\[sub-interface number] <priority>
```

For parameters and examples, see “Using the ip dhcp relay information option command”.

DHCP Agent ID Overwrite

When configured, the SG-1 should overwrite the DHCP agent ID and the server identifier to its ID (IP address) in the DHCP reply messages. The system does not overwrite the DHCP agent ID in its default behavior.

Using the ip dhcp relay agent-id-overwrite command

```
Host(config)# ip dhcp relay agent-id-overwrite interface
```

Usage

```
ip dhcp relay agent-id-overwrite interface <Ethernet | ATM | VLAN | default>
<slot>\<port number>\[sub-interface number] <priority> [replace]
```

Parameter(s)

| Parameter | Description | Legal values/range |
|-----------------------------------|-----------------------|--------------------------|
| <slot> | PMC slot number. | 0 - 2; Default value 0 |
| <port number > | Physical port number. | 1 - 2; Default value 0 |
| <sub-interface number> | Sub interface number. | 1- 4095; Default value 0 |
| <priority> | DHCP server priority. | 1- 10 |
| <Ethernet ATM VLAN default> | Interface format. | Ethernet, ATM, VLAN |

Example(s)

```
Host(config)# ip dhcp relay agent-id-overwrite interface Ethernet 0 \ 1 3
Host(config)# ip dhcp relay agent-id-overwrite interface Ethernet 0 \ 1 3 replace
Host(config)# ip dhcp relay agent-id-overwrite interface Atm 2 \ 1 \ 43
Host(config)# ip dhcp relay agent-id-overwrite interface Atm 2 \ 1 \ 43 8 replace
Host(config)# ip dhcp relay agent-id-overwrite interface VLAN 1 \ 2 34
Host(config)# ip dhcp relay agent-id-overwrite interface VLAN 1 \ 2 34 5 replace
Host(config)# ip dhcp relay agent-id-overwrite interface default 0 \ 1 \ 1
Host(config)# ip dhcp relay agent-id-overwrite interface default 0 \ 1 \ 1 2 replace
```

Using the ip dhcp relay advertise-protocol command

```
Host(config)# ip dhcp relay advertise-protocol
```

This command defines the default protocol the system is using to advertise the user's DHCP allocated IP address.

Usage

```
ip dhcp relay advertise-protocol <rip | ospf>
```

IGMP COMMANDS

Using the ip igmp proxy command

```
Host(config)# ip igmp proxy upstream-interface
```

The **ip igmp proxy upstream-interface** command enables the IGMP Proxy capabilities on a specific upstream interface. When the upstream interface is not configured, the system does not support the IGMP proxy capability. Additionally, the command enables you to add or change the IGMP Proxy upstream interface.



Note: Only routed ATM interfaces are allowed to be upstream interface.

Usage

```
ip igmp proxy upstream-interface <Ethernet | ATM | VLAN> <slot>\<port number>
[\sub-interface]
```

Table 7-13. ip igmp proxy command parameters

| Parameter | Description | Legal values/range |
|------------------------|-----------------------|--------------------------|
| <slot> | PMC slot number. | 0 - 1 |
| <port number> | Physical port number. | 1 - 2 |
| <sub-interface number> | Sub interface number. | 1- 4095; Default value 0 |

Example(s)

```
Host(config)# ip igmp proxy upstream-interface Ethernet 0 \ 1
Host(config)# ip igmp proxy upstream-interface Atm 2 \ 1 \ 3
Host(config)# ip igmp proxy upstream-interface VLAN 1 \ 2 54
```

Using the no ip igmp proxy command

```
Host(config)# no ip igmp proxy upstream-interface
```

The no igmp proxy upstream-interface command disables the IGMP Proxy capabilities.

Usage

```
no igmp proxy upstream-interface <Ethernet | ATM | VLAN> <slot>\<port number>
[\sub-interface]
```

ROUTING COMMAND

Using the ip forward command

```
Host(config)# ip forward
```

This command enables IP forwarding between the Ethernet interfaces. The system's default setting does not use IP forwarding.

Usage

```
ip forward
```

```
Host(config)# no ip forward
```

This command disables IP forwarding between the Ethernet interfaces.

Usage

```
no ip forward
```

Using the IP route command

```
Host(config)# ip route
```

The **ip route** command is used for establishing static routes. A static route is appropriate when the SG-1 is not dynamically building the routes to the destinations.

Usage

```
ip route [<nip <destination prefix> <destination prefix mask> <forwarding router's>]
| [<destination prefix> <destination prefix mask> <[<forwarding router's>] |
[<<Ethernet interface | ATM interface> <slot number>\<port number>\[<sub-interface
number]>>| <loopback interface>]>>
```

Parameter(s)

Table 7-14. ip route command parameters

| Parameter | Description | Values |
|--------------------------------------|----------------------------|---------------------------------|
| <destination prefix> | Destination IP or Network | Valid IP or Network |
| <destination prefix mask> | Network mask | Valid Mask (Except to 0.0.0.0) |
| <forwarding router's> | Next-hop IP address | Valid IP address |
| <Ethernet interface ATM interface> | Interface type | Interface type: Ethernet or ATM |
| <slot number> | PMC slot number | 0 - 1 |
| <port number> | SONET physical port number | 1 - 2 |
| <sub-interface number> | Sub interface number | 1- 2000 |

| Parameter | Description | Values |
|-----------------------|-----------------------------|---------|
| <loopback interface> | Loopback interface number | 1 - 200 |
| <tunnel interface ip> | Tunnel interface identifier | 1 - 500 |

Example(s)

Add an IP route:

```
Host(config)#ip route 192.168.3.0 255.255.255.0 194.90.1.12 <cr>
Host(config)# end
Host> write terminal
. . .
ip route 192.168.3.0 255.255.255.0 194.90.1.12
. . .
```

Modify an Existing IP route:

```
Host(config)# ip route 192.168.3.0 255.255.255.0 194.90.1.22 <cr>
Host(config)# end
Host> write terminal
...
ip route 192.168.3.0 255.255.255.0 194.90.1.22 Ethernet 0\1
...
```

Route network 192.168.4.0 to Ethernet interface 2:

```
Host(config)# ip route 192.168.4.0 255.255.255.0 194.90.1.22 ethernet 2 <cr>
Host(config)# end
Host> write terminal
...
ip route 192.168.4.0 255.255.255.0 194.90.1.22 ethernet 0\2
...
```

Route network 192.168.4.0 to ATM interface 2\1\3:

```
Host(config)# ip route 192.168.4.0 255.255.255.0 194.90.1.22 ATM 2\1\3 <cr>
Host(config)# end
Host> write terminal
...
ip route 192.168.4.0 255.255.255.0 194.90.1.22 ATM 2\1\3
...
```

Route network 192.168.4.0 to Loopback interface 2:

```
Host(config)# ip route 192.168.4.0 255.255.255.0 194.90.1.22 loopback 3 <cr>
Host(config)# end
Host> write terminal
...
ip route 192.168.4.0 255.255.255.0 194.90.1.22 loopback 3
...
```

Route network 192.168.4.0 to Ethernet interface 0\2:

```
Host(config)# ip route 192.168.4.0 255.255.255.0 ethernet 0\2 <cr>
Host(config)# end
Host> write terminal
...
ip route 192.168.4.0 255.255.255.0 Ethernet 0\2
...
```

Route native IP user side network 192.168.4.0 to 192.168.1.1:

```
Host(config)# ip route nip 192.168.4.0 255.255.255.0 192.168.1.1 <cr>
Host(config)# end
Host> write terminal
...
ip route nip 192.168.4.0 255.255.255.0 192.168.1.1
...
```

Route network 10.0.4.0/24 to tunnel interface 3:

```
Host(config)# ip route 10.0.4.0 255.255.255.0 tunnel 3 <cr>
Host(config)# end
Host> write terminal
...
ip route nip 10.0.4.0 255.255.255.0 tunnel 3
...
```



Note: When assigning a next-hop address which is not in the interfaces subnet, the system should ignore the command and indicate the reason.

```
Host(config)# ip route 192.168.4.0 255.255.255.0 194.90.1.22 <cr>
Operation Error:
Next-hop is not valid to this interface
```



Note: It is not valid to add an ip route when the destination address is already in the subnet of the system interfaces.

```
Host(config)# ip route 192.168.1.0 255.255.255.0 192.168.1.2
```

Deleting an IP route line

```
Host(config)# no ip route
```

This command deletes the existing route.

Usage

no ip route [nip] <destination prefix> <network mask> <next-hop IP-address>

Parameter(s)

Table 7-15. no ip route command parameters

| Parameter | Description | Values |
|-----------------------|---------------------------|--------------------------------|
| [nip] | Native IP | Valid IP or Network |
| <destination prefix> | Destination IP or network | Valid IP or Network |
| <network mask> | Network mask | Valid Mask (Except to 0.0.0.0) |
| <next-hop IP-address> | Next-hop IP address | Valid IP address |

Example(s)

```
Host(config)# no ip route 192.168.0.0 255.255.0.0 192.168.0.1
```

Using the ip default-gateway command

```
Host(config)# ip default-gateway
```

Use this command to specify the system's default gateway.

Usage

ip default-gateway <gateway IP address>

Parameter(s)

<gateway IP address>

The IP address of the default gateway.

Example(s)

```
Host(config)# ip default-gateway 12.3.11.56
```

Using the no ip default-gateway command

```
Host(config)# no ip default-gateway
```

Use this command to specify the system's default gateway.

Usage**no ip default-gateway****Example(s)**

```
Host(config)# no ip default-gateway
```

When assigning a default-gateway that is not in the subnet of the primary or secondary Ethernet interfaces, the system provides a warning.

```
Host(config)# ip default-gateway 194.90.2.1
  Operation Warning:
  The default gateway is out of subnet
```

Using the router command

```
Host(config)# router
```

This command defines the system default routing process. In cases where advertisement should take place and there is no other higher priority definition, the system uses this definition. The routing modes include Routing Information Protocol Version 2 (RIPv2) and Open Shortest Path First protocol (OSPF).

Usage**router** <ripv2 | ospf>**Example(s)**

```
Host(config)# router ripv2
Host(config)# router ospf
```

Using the router id command

```
Host(config)# router id
```

This command defines the system router id.

Usage**router id** < router ip>**Parameter(s)**

<router ip>

OSPF router ID in IP address format.

Example(s)

```
Host(config)# router id 10.33.21.88
```

```
Host(config)# no router
```

This command disables the system default routing process.

Usage

no router

Using the IP rip authentication key command

```
Host(config)# ip rip authentication key
```

The **ip rip** command is located beneath the "configure terminal" menu. It is used to define the password for the Router Information Protocol (RIP) authentication process.

Usage

ip rip authentication key <key name>

Parameter(s)

<key name>

This is the authentication key.

Using the ip ospf interface command

```
Host(config)# ip ospf interface ... area
```

This command defines an interface on which OSPF runs and defines the area ID for that interface.

Usage

ip ospf interface <Ethernet | VLAN | ATM> <slot number>\<port number>
[<sub-interface>] **area** <area-id>

Parameter(s)

<area-id>

The area ID for the interface on which OSPF is running.

For additional parameters, refer to "Using the ip igmp proxy command" on page 7-51.

Example(s)

```
Host(config)# ip ospf interface Ethernet 1 \ 2 area 12.3.5.6
Host(config)# ip ospf interface ATM 2 \ 1 \ 32 area 12.5.45.8
Host(config)# ip ospf interface VLAN 1 \ 2 \ 33 area 44.55.2.15
```

Using the no ip ospf interface command

```
Host(config)# no ip ospf interface
```

This command disables OSPF on an interface.

Usage

```
no ip ospf interface <Ethernet | VLAN | ATM> <slot number>\<port number>
[ \<sub-interface>]
```

For parameters and examples, refer to “Using the ip ospf interface command” on page 7-57.

Using the ip ospf interface hello-interval command

```
Host(config)# ip ospf interface ... hello-interval
```

This command specifies the interval time in seconds between hello packets.

Usage

```
ip ospf interface <Ethernet | VLAN | ATM> <slot number>\<port number>
[ \<sub-interface>] hello-interval <seconds>
```

Parameter(s)

<seconds>

The time interval between hello packets.

For additional parameters, refer to “Using the ip igmp proxy command” on page 7-51.

Example(s)

```
Host(config)# ip ospf interface Ethernet 1 \ 2 hello-interval 22
Host(config)# ip ospf interface ATM 2 \ 1 \ 44 hello-interval 32
Host(config)# ip ospf interface VLAN 1 \ 2 \ 33 hello-interval 23
```

Using the no ip ospf interface hello-interval command

```
Host(config)# no ip ospf interface ... hello-interval
```

The command sets the interval time between hello packets to its default value.

Usage

```
no ip ospf interface <Ethernet | VLAN | ATM> <slot number>\<port number>
[ \<sub-interface>] hello-interval
```

For parameters and examples, refer to “Using the ip ospf interface hello-interval command”.

Using the ip ospf interface dead-interval command

```
Host(config)# ip ospf interface ... dead-interval
```

The command specifies the number of seconds that a device's hello packets must not have been seen before its neighbor declares the OSPF router down.

Usage

```
ip ospf interface <Ethernet | VLAN | ATM> <slot number>\<port number>  
[<sub-interface>] dead-interval <seconds>
```

Parameter(s)

<seconds>

Interval a device's hello packets must not have been seen before its neighbor declares the OSPF router down.

For additional parameters, refer to [“Using the ip igmp proxy command” on page 7-51](#).

Example(s)

```
Host(config)# ip ospf interface Ethernet 1 \ 2 dead-interval 33  
Host(config)# ip ospf interface ATM 2 \ 1 \ 44 dead-interval 90  
Host(config)# ip ospf interface VLAN 1 \ 2 \ 88 dead-interval 78
```

Using the no ip ospf interface dead-interval command

```
Host(config)# no ip ospf interface ... dead-interval
```

The command set the number of seconds that a device's hello packets must not have been seen before its neighbor declares the OSPF router down to its default value.

Usage

```
no ip ospf interface <Ethernet | VLAN | ATM> <slot number>\<port number>  
[<sub-interface>] dead-interval
```

For parameters and examples, refer to [“Using the ip ospf interface dead-interval command” on page 7-59](#).

Using the ip ospf interface authentication command

```
Host(config)# no ip ospf interface ... authentication
```

This command is used to set the type of authentication (simple password, MD5, or none) used by neighboring OSPF routers.

Usage

```
ip ospf interface <Ethernet | VLAN | ATM> <slot number>\<port number>  
[<sub-interface>] authentication <simple-pass | message-digest | null>
```

Parameter(s)

<simple-pass | message-digest | null>

This is the authentication type specified for neighboring OSPF routers. Options include:

- **simple-pass**—using simple password authentication; to configure see [“Using the ip ospf interface authentication-key command” on page 7-60](#).
- **message-digest**—using Message Digest 5 (MD5) authentication; to configure see [“Using the ip ospf interface message-digest-key command” on page 7-61](#).
- **null**—no authentication. (**Note:** If no value is keyed in, the system assumes the default value, which is null).

For additional parameters, refer to [“Using the ip igmp proxy command” on page 7-51](#).

Example(s)

```
Host(config)# ip ospf interface Ethernet 1 \ 2 authentication simple-pass
Host(config)# ip ospf interface Ethernet 1 \ 2 authentication message-digest
Host(config)# ip ospf interface Ethernet 1 \ 2 authentication null
```

Using the ip ospf interface authentication-key command

```
Host(config)# ip ospf interface ... authentication-key
```

This command configures the password to be used by neighboring OSPF routers on a network segment that is using the OSPF simple password authentication.

Usage

```
ip ospf interface <Ethernet | VLAN | ATM> <slot number>\<port number>
[<sub-interface>] authentication-key <key>
```

Parameter(s)

<key>

The password used by neighboring OSPF routers for simple password authentication (password is 1 to 16 alpha-numeric characters).

For additional parameters, refer to [“Using the ip igmp proxy command” on page 7-51](#).

Example(s)

```
Host(config)# ip ospf interface Ethernet 0 \ 1 authentication-key jkljlll
Host(config)# ip ospf interface ATM 2 \ 1 \ 44 authentication-key wqweqrr
Host(config)# ip ospf interface VLAN 1 \ 1 \ 44 authentication-key jkjkjk
```

Using the no ip ospf interface authentication-key command

```
Host(config)# no ip ospf interface ... authentication-key
```

On a network segment using the OSPF simple password authentication, this command sets the password used by neighboring OSPF routers to its default value.

Usage

```
no ip ospf interface <Ethernet | VLAN | ATM> <slot number>\<port number>
[\<sub-interface>] authentication-key
```

For parameters and examples, refer to [“Using the ip ospf interface authentication-key command”](#).

Using the ip ospf interface message-digest-key command

```
Host(config)# no ip ospf interface ... message-digest-key
```

This command configures the OSPF MD5 (Message Digest 5) authentication parameters.

Usage

```
ip ospf interface <Ethernet | VLAN | ATM> <slot number>\<port number>
[\<sub-interface>] message-digest-key <keyid> md5 <key>
```

Parameter(s)

<keyid>

This is the MD5 identifier to be used by neighboring OSPF routers.

<key>

This is the MD5 password (1 to 16 alphanumeric characters).

For additional parameters, refer to [“Using the ip igmp proxy command” on page 7-51](#).

Example(s)

```
Host(config)# ip ospf interface Ethernet 1 \ 2 message-digest-key 25 md5 jkljlll
```

Using the no ip ospf interface message-digest-key command

```
Host(config)# no ip ospf interface ... message-digest-key
```

This command removes the OSPF MD5 (Message Digest 5) authentication parameters.

Usage

```
no ip ospf interface <Ethernet | VLAN | ATM> <slot number>\<port number>
[\<sub-interface>] message-digest-key
```

For parameters and examples, refer to [“Using the ip ospf interface message-digest-key command” on page 7-61](#).

Using the ip ospf area stub command

```
Host(config)# ip ospf area
```

This command configures an OSPF area as a stub area. The system default stubbing option is *no-stub* and the system default advertisement behavior is *summary*.

Usage

```
ip ospf area <area-id> [stub | no-stub] [no-summary | summary]
```

Parameter(s)

<area-id>

OSPF identifier of the area the identifier specified (as a value or an IP address).

Example(s)

```
Host(config)# ip ospf area 1.2.3.5 stub no-summary
Host(config)# ip ospf area 1.2.3.5 stub summary
```

Using the no ip ospf area stub command

```
Host(config)# no ip ospf area ... stub
```

The command disables OSPF stub area functionality.

Usage

```
no ip ospf area <area-id> stub
```

For parameters and examples, refer to “Using the ip ospf area stub command” on page 7-62.

Using the ip ospf advertise network command

```
Host(config)# ip ospf advertise network
```

This command defines advertisement of an entire network per area.

Usage

```
ip ospf advertise network <area-id> <network> <mask>
```

Parameter(s)

Table 7-16. ip ospf advertise network command

| Parameter | Description | Legal values/range |
|-----------|---------------------------------------------------------|--------------------|
| <area-id> | OSPF Identifier of the area specified as an IP address. | IP address |
| <network> | Advertised network. | IP address |
| <mask> | Network mask. | Mask |

Example(s)

```
Host(config)# ip ospf advertise-network 1.2.3.4 22.32.55.4 255.255.255.0
```

Using the no ip ospf advertise network command

```
Host(config)# no ip ospf advertise network
```

This command deletes advertisement of a network.

Usage

```
no ip ospf advertise network <area-id> <Network>
```

For parameters and examples, refer to “Using the ip ospf advertise network command” on page 7-62.

Using the mpls ip interface command

```
Host(config)# mpls ip interface
```

This command enables MPLS forwarding of Ipv4 packets for a specific system interface. One limitation is that currently the MPLS may be configured for one interface only of type VLAN or Ethernet.

Usage

```
mpls ip interface <Ethernet | VLAN | ATM > <slot number> \ <port number>  
[\<sub-interface>]
```

Parameter(s)

<slot number>

This is the SCC slot number; valid number range is 0 to 2.

<port number>

This is the SCC port number; valid values are 1 or 2.

<sub-interface>

This is the SCC for ATM sub-interface number; valid number range is 1 to 4095.

Example(s)

```
Host(config)# mpls ip Ethernet 0\1
```

Using the no mpls ip interface command

```
Host(config)# no mpls ip interface
```

This command disables mpls forwarding of IPv4 packets from the system interface.

Usage

```
no mpls ip interface <Ethernet | VLAN | ATM > <slot number> \ <port number>  
[\<sub-interface>]
```

Parameter(s)

<slot number>

This is the SCC slot number; valid number range is 0 to 2.

<port number>

This is the SCC port number; valid values are 1 or 2.

<sub-interface>

This is the SCC for ATM sub-interface number; valid number range is 1 to 4095.

Example(s)

```
Host(config)# no mpls ip Ethernet 0\1
```

Using the mpls l2transport interface command

```
Host(config)# mpls l2transport interface
```

This command defines an MPLS I2vpn interface based on Martini. In order to change the mpls interface parameters, the command should be deleted using the **no mpls l2transport interface** command and then reconfigured. The maximum number of mpls l2transport interfaces in the system is 2,000.

Usage

```
mpls l2transport interface <destination> <vc-id> [<type <<ethernet> | <VLAN  
<VLAN-ID>>> <IP address> <mask>]
```

Parameter(s)

<destination>

This specifies the LDP IP address of the remote PE; it should be a valid IP address.

<vc-id>

This assigns a VC ID to the virtual circuit between the system and the remote PE.

[<type>

This is the terminated interface type L2 encapsulation.

VLAN-ID

This the VLAN identifier.

<IP address>

This is the interface IP address.

<mask>

This is the interface IP mask.

Example 1: create mpls L2 VPN based on Martini draft for redirection

```
Host(config)# mpls ip Ethernet 0\1
Host(config)# mpls l2transport interface 194.90.1.4 200
```

Example 2: create mpls L2 VPN based on Martini draft for termination

```
Host(config)# mpls ip Ethernet 0\1
Host(config)# mpls l2transport interface 194.90.1.4 200 type Ethernet
```

Using the no mpls l2transport interface command

```
Host(config)# no mpls l2transport interface
```

This command detects an MPLS l2vpn interface based on Martini.

Usage

```
no mpls l2transport interface <destination> <vc-id>
```

Parameter(s)

<destination>

This specifies the LDP IP address of the remote PE; it should be a valid IP address.

<vc-id>

This assigns a VC ID to the virtual circuit between the system and the remote PE.

Example: delete MPLS L2 VPN based on Martini draft

```
Host(config)# no mpls l2transport interface 194.90.1.4 200
```

Using mpls l2transport route command

```
Host(config)# mpls l2transport route interface
```

It redirects all interface traffic through MPLS l2vpn tunnel based on Martini. The maximum number of mpls l2transport routes in the system is 2,000.

Usage

```
mpls l2transport route interface <Ethernet | VLAN> <slot>/<port number>/
[<sub-interface number>] <destination> <vc-id>
```

Parameter(s)

<slot number>

This is the SCC slot number; valid number range is 0 to 2.

<port number>

This is the SCC port number; valid number range is 1 to 3.

<sub-interface>

This is the SCC sub-interface number; valid number range is 2 to 4095.

<destination>

This specifies the LDP IP address of the remote PE; it should be a valid IP address.

<vc-id>

This assigns a VC ID to the virtual circuit between the system and the remote PE.

Example: Route VLAN interface through an MPLS L2 VPN based on Martini draft

```
Host(config)# mpls ip Ethernet 0\1
Host(config)# mpls l2transport interface 194.90.1.4 200
Host(config)# mpls l2transport route interface VLAN 0\1\100 194.90.1.4 200
```



Note: The system should refuse the mpls l2transport route command if the type is configured on the mpls interface.

```
Host(config)# mpls l2transport interface 194.90.1.4 200
Host(config)# mpls l2transport route interface VLAN 0\1\100 194.90.1.4 200
Host(config)# mpls l2transport route interface VLAN 0\1\101 194.90.1.4 200
Operation error
Tunnel is used by another interface
```



Note: The system should refuse the mpls l2transport route command if the type is configured on the mpls interface.

Using the no mpls l2transport route command

```
Host(config)# no mpls l2transport route interface
```

It detects redirected interface traffic through MPLS l2vpn tunnel based on Martini draft.

Usage

```
no mpls l2transport route interface <Ethernet | VLAN> <slot>/<port number>/
[<sub-interface number>]
```

Parameter(s)

<slot number>

This is the SCC slot number; valid number range is 0 to 2.

<port number>

This is the SCC port number; valid number range is 1 to 3.

<sub-interface>

This is the SCC sub-interface number; valid number range is 2 to 4095.

Example: detects VLAN interface redirection to MPLS L2 VPN based on Martini draft

```
Host(config)# no mpls l2transport route interface VLAN 0\1\100
```

Using mpls ip default-route command

```
Host(config)# mpls ip default-route
```

It enables the distribution of labels associated with the IP default route.

Usage

```
mpls ip default-route
```

Example

```
Host(config)# mpls ip default-route
```

Using no mpls ip default-route command

```
Host(config)# no mpls ip default-route
```

It disables the distribution of labels associated with the IP default route.

Usage

```
no mpls ip default-route
```

Example

```
Host(config)# no mpls ip default-route
```

Using the vrrp command

```
Host(config)# vrrp interface
```

The SCC redundancy mechanism is based on the Virtual Router2 Redundancy Protocol (VRRP RFC-2338), which is designed to eliminate the single point of failure inherent in the static default routed environment. VRRP specifies an election protocol that dynamically assigns responsibility for a virtual router to one of the VRRP routers on a LAN. The VRRP router controlling the IP address(es) associated with a virtual router is called the Master, and forwards packets sent to these IP addresses. The election process provides dynamic fail-over in the forwarding responsibility should the Master become unavailable. Any of the virtual router's IP addresses on a LAN can then be used as the default first hop router by end-hosts.

This is beneficial because VRRP is a higher availability default path without requiring configuration of dynamic routing or router discovery protocols on every end-host.

Usage

```
vrrp interface <Ethernet | VLAN> <slot number>\<port number>[\VLAN ID] <number> ip  
<IP address> priority <priority value> [preempt-mode <on | off>]
```



Note: The virtual router IP should belong to the same subnet as the protected interface. The VRRP preempt mode default value is *off*. The **write terminal** command should not present the preempt-mode when set to its default value.

Table 7-17. vrrp command parameters

| Parameter | Description | Legal values/range |
|-------------------|-------------------------------------------|------------------------------------------|
| <Ethernet VLAN> | The interface should be Ethernet or VLAN. | Ethernet, VLAN |
| <slot number> | Slot number. | 0, 1 |
| <port number> | Physical port number. | 1 to 3 |
| <VLAN ID> | VLAN ID number. | |
| <number> | Virtual Router Group Number (VRID). | 1 to 15 |
| IP address | IP address of the virtual router. | Legal IP address |
| <priority value> | VRRP router priority value. | 1 to 254 |
| <on off> | VRRP preemptive mode. | On or off (default value is <i>off</i>) |

The following examples use two SCCs (Service Creation Cards):

Example 1 (SCC1 configuration):

```
Host(config)# vrrp interface Ethernet 0\1 1 ip 192.168.1.3 priority 200
```

Example 2 (SCC2 configuration):

```
Host(config)# vrrp interface Ethernet 0\1 1 ip 192.168.1.3 priority 100
```

The following two examples show the configuration with preemption capabilities:

Example 1 (SCC1 configuration):

```
Host(config)# vrrp interface Ethernet 0\1 1 ip 192.168.1.100 priority 200 preempt-
mode on

Host(config)# vrrp interface Ethernet 0\1 2 ip 192.168.1.101 priority 201 preempt-
mode on
```

Example 2 (SCC2 configuration):

```
Host(config)# vrrp interface Ethernet 0\1 1 ip 192.168.1.101 priority 200 preempt-
mode on

Host(config)# vrrp interface Ethernet 0\1 2 ip 192.168.1.100 priority 201 preempt-
mode on
```



Note: The system should refuse virtual router IP address that is the same as the real interface IP address.

```
Host(config)# vrrp interface Ethernet 0\1 1 ip 192.168.1.2 priority 1
Operation Error:
Virtual router IP address is equal to the real interface IP address
```



Note: The system should refuse virtual router IP address that does not belong to the same subnet as the protected interface, or belong to one of the system IP pools.

```
Host(config)# vrrp interface Ethernet 0\1 1 ip 192.168.1.2 priority 1
Operation Error:
Wrong virtual router IP address
```



Note: The system should refuse assignment of different virtual router priority for the same VRID value.

```
Host> write terminal
....
Interface ethernet 0\1 192.168.1.2 255.255.255.0 auto
Vrrp interface ethernet 0\1 1 ip 10.1.1.1 priority 99
```

```
....
Host(config)# vrrp interface ethernet 0\1 1 ip 10.1.1.1 priority 150
Operation Error:
Different virtual router priority for the same group number (VRID)
```

Using the no vrrp command

```
Host(config)# no vrrp interface
```

This command deletes the virtual router configuration in the system.

Usage

```
no vrrp interface <Ethernet | VLAN> <slot number> \ <port number> [\ <VLAN ID>]
<number>
```

Parameter(s)

Table 7-18. no vrrp command parameters

| Parameter | Description | Legal values/range |
|-------------------|-------------------------------------------|--------------------|
| <Ethernet VLAN> | The interface should be Ethernet or VLAN. | Ethernet, VLAN |
| <slot number> | Slot number. | 0, 1 |
| <port number> | Physical port number. | 1 to 3 |
| <VLAN ID> | VLAN ID number. | |
| <number> | Virtual Router Group Number (VRID). | 1 to 15 |

Example(s)

```
Host(config)# no vrrp interface Ethernet 0\1 1
```

Using the vrrp preempt command

```
Host(config)# vrrp preempt interface
```

- Executing the **vrrp preempt interface** command orders a backup system to become active (become the master) if it is configured to the highest virtual router priority.
- Executing the **vrrp preempt interface** command on a backup system, which is in a lower priority than the master, does not change the master and backup activity.

Usage

```
vrrp preempt interface <Ethernet | VLAN> <slot number>\<port number> <priority>
enable
```

Table 7-19. vrrp preempt command parameters

| Parameter | Description | Legal values/range |
|-------------------|-------------------------------------------|--------------------|
| <Ethernet VLAN> | The interface should be Ethernet or VLAN. | Ethernet, VLAN |
| <slot number> | Slot number. | 1, 1 |
| <port number> | Physical port number. | 1 to 3 |
| <priority> | VRRP router priority value. | 1 to 254 |

Example(s)

```
Host(config)# vrrp preempt interface VLAN 1\2 7 enable
```

DEBUG COMMANDS

Use the commands at the second level prompt to switch the system to the second-level debug prompt, `Host(config-debug)#`. This indicates that the user is now in the second level debug mode and has access to the commands in the debug menu.

There are three types of messages sent by the POPmaestro/SG-1 system to its logger. They are: Error Log, Event Log and Trace Log.

Error Log

An Error Log is a message sent by the system when an error that requires an operator's attention occurs. It can be either an internal error in the system's operation (usually this would be a high-level error) or an error caused from outside the system (like wrong data sent to the system).

Trace Log

A Trace Log is a message that outputs the contents of some buffer or some area in the memory, as a result of a certain event. It can be either when an error occurs, or when it is important to view a certain stream of bytes, like an important received packet.

Event Log

An Event Log is a message-documenting occurrence of an event in the system.

Configure watchdog

```
(config-debug)# watchdog-TimeValue
```

Usage

```
(config-debug)# watchdog-TimeValue <number>
```

Parameter

<number>

It is the timeout in seconds (0 = unlimited)

Example

```
(config-debug)# watchdog-TimeValue 60 (cr)
(config-debug)#
```

Configure time server

```
(config-debug)# time-server-ip
```

Usage

```
(config-debug)# time-server-ip <IP address>
```

Parameter

<IP address>

It is timer host IP address, legal IP address.

Example

```
(config-debug)# time-server-ip 10.6.1.71 (cr)
(config-debug)#
```

Configure error level commands

```
(config-debug)# error-level
```

```
(config-debug)# error-level < default | Module | Group>
```

Parameters

<default>

System's Default maximum error level

<Module>

System's error log by Module

<Group>

System's error log by Module

Example

```
(config-debug)# error-level default
```

```
(config-debug)# error-level default [ set-all <number>]
```

Parameters

set-all

Will set all modules back to default

<number>

value of default maximum error level , numbers are

Examples

```
(config-debug)# error-level default 4 output-device console
(config-debug)#
(config-debug)# error-level default 3 output-device console
(config-debug)#
```

```
(config-debug)# error-level
```

Usage

```
(config-debug)# error-level < default | Module | Group >
```

Parameters

<default>

System's Default maximum error level

<Module>

System's error log by Module

<Group>

System's error log by Module

Debug modules

```
(config-debug)# error-level Module <cr>
```

- ABM
- AbmFSM
- AbmIpPool
- AbmMIPPP
- AbmRadius
- AbmService
- AbmRadiusProxy
- EDS
- PPP
- PPPWrapper
- PPPService
- NativeIP
- L2TP
- Telnet
- CPM
- System
- DataPoller
- BSP
- ARP
- DHCP
- Router
- OSPFv2
- IPMgr
- IPinIP
- ICMP
- POPUDP
- RsmFrgm
- VRRP
- MPLS
- NetIf
- SysLogger
- Timer

Usage

```
(config-debug)# error-level Module ABM < max | min | default >
```

Parameters

< max >

Maximum level

< min >

Minimum level

< default >

Set module's levels to default

Examples

```
(config-debug)# error-level Module ABM default (cr)
(config-debug)#
(config-debug)# error-level Module ABM max 3 (cr)
(config-debug)#
```

Usage

```
(config-debug)# error-level Group AAA < max | min | default >
```

< max >

Maximum level

< min >

Minimum level

< default >

Set module's levels to default

Examples

```
(config-debug)# error-level Group AAA max 2
(config-debug)#
(config-debug)# error-level Group AAA default
(config-debug)#
```

Debug groups

```
(config-debug)# error-level Group <cr>
- AAA
- User
- IP
- Route
- System
- PPP
- Service
- NativeIP
- Interface
- L2TP
```

Examples

```
(config-debug)# event-level default 2 output-device console (cr)
(config-debug)#
(config-debug)# error-level default set-all (cr)
(config-debug)#
(config-debug)# event-level default set-all (cr)
(config-debug)#
```

Trace commands

```
(config-debug)# trace
```

Usage

```
(config-debug)# trace < <default < set-all | off | on > | Module | Group >>
```

Parameters

<default>
System's Default trace setting

<Module>
System's trace log by Module

<Group>
System's trace log by Module

Usage

```
(config-debug)# trace default < set-al | off | on>
```

Parameters

<set-all>
Set all modules back to default

< off >
without trace

< on >
with trace

Examples

```
(config-debug)# trace default on (cr)
(config-debug)#

(config-debug)# trace default off (cr)
(config-debug)#
```

Configure

```
(config-debug)# sysLog-server-ip
```

Usage

```
(config-debug)# sysLog-server-ip <IP address>
```

Parameter

<IP address >

It is the SysLogger IP address.

```
(config-debug)# exit
```

Usage

```
(config-debug)# exit
```

This command will cause exit current configuration level.

Example

```
(config-debug)# exit <cr>
Host>
```

```
(config-debug)# end
```

Usage

(config-debug)# end

This command will cause Return to first configuration level.

Example

```
(config-debug)# end <cr>  
(config)#
```

SG-1 VENDOR-SPECIFIC ATTRIBUTES

This appendix describes the vendor-specific attributes related to SG-1 EDS architecture.

OVERVIEW

The vendor-specific attributes are based on RFC-2865 RADIUS recommendation. The first 4 octets are the vendor id (supported vendor ID 2454, 2014). The next two octets are the vendor-type and length as recommended in the RFC.

The attributes are split into the following groups:

- User
- DHCP
- protocol
- service
- route
- vpdn
- qos
- DNS

Vendor-Specific Attributes Table

When the system sends a vendor-specific attribute to the radius, it sets the vendor type with the attribute internal number, as specified below in the vendor-specific attribute list (Table A-1).

Table A-1. Vendor-Specific Attribute List

| Num | Attribute Name | Attribute Group | Attribute Value (1 byte which takes value of 1 to 255) | First Introduced | Description |
|-----|----------------|-----------------|--------------------------------------------------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | accounting | user | 10 | V5.0 | Allows defining the session accounting operation mode and methodology. |
| 2 | orig-name | user | 12 | V5.0 | Contains the original user name as received during PPP negotiation. |
| 3 | auth-type | user | 13 | V5.0 | Contains the authentication type of the Access Request message. It provides the RADIUS with additional information regarding the purpose of the authentication. |
| 4 | action | user | 14 | V5.0 | Defines the action that should be taken by the system. |
| 5 | SSC-host | user | 15 | V5.0 | Contains the SSC (Service Selection Center) host IP address at which the user activated the service. |

| | | | | | |
|----|-----------------------|----------|----|------|---------------------------------------------------------------------------------------------------------------------------|
| 6 | service-name | user | 16 | V5.0 | Contains the information of the service name, which was given to the connected peer or the peer requested service name. |
| 7 | personal-site | user | 17 | V5.0 | Contains the personal site to which the user should be redirected. |
| 8 | mac-address | user | 18 | V7.0 | Contains the MAC address information of a connected user as learned by the DHCP relay or by the proxy RADIUS. |
| 9 | group | user | 19 | V7.0 | Contains the user group number (1 to 1000) as defined in the RADIUS server. |
| 10 | max-allowed-sessions | user | 20 | V7.0 | Defines the maximum number of sessions allowed in a single blade per username. |
| 11 | class | user | 21 | V80 | Contains the user class information, a string of maximum size of 256 characters. |
| 12 | eds-enc-key | user | 22 | V80 | The user:eds-enc-key sub-attribute contains an encryption key for EDS operation. |
| 13 | eds-cookie | user | 23 | V90 | The user:eds-cookie sub-attribute contains a user eds cookie data information, a string of maximum size of 64 characters. |
| 14 | original-url-prefix | user | 24 | V90 | The sub-attribute contains a prefix string that should be prefixed by the RDS to the user original url request. |
| 15 | dhcp-server | dhcp | 30 | V7.0 | Defines the DHCP server IP address, which the system should relay the user's DHCP requests |
| 16 | opt82-relay-remote-id | dhcp | 31 | V7.0 | Contains the received option 82 relay remote ID sub-option, while each byte information is in hexadecimal format. |
| 17 | discover-action | dhcp | 32 | V7.0 | Defines the action to be taken when a new dhcp discover message is transmitted in a connected session |
| 18 | type | protocol | 40 | V5.0 | Contains a hint of the protocol negotiated with the peer. |

| | | | | | |
|----|---------------------------|---------|----|------|-----------------------------------------------------------------------------------------------------------------------|
| 19 | service-timeout | service | 50 | V5.0 | Defines the service session timeout measured in seconds. |
| 20 | next-service-name | service | 51 | V5.0 | Defines the name of the next service to provide when a service "session timeout" expires. |
| 21 | auto-service-name | service | 52 | V5.0 | Defines the service name to be automatically provided when the user is redirected by the RDS. |
| 22 | auth-source | service | 53 | V7.0 | Defines the source name to be used when the SG-1 authorizes or authenticates a service with the RADIUS. |
| 23 | data-quota | service | 54 | V7.0 | Defines the service session data quota measured in bytes. |
| 24 | acl-data-quota | service | 55 | V7.0 | Defines the service session data quota per a specified access-list measured in bytes. |
| 25 | service-cache | service | 56 | V7.0 | Contains the service caching operation mode of the received service. |
| 26 | data-quota-used | service | 57 | V7.0 | Contains the session's used quota in bytes. |
| 27 | acl-data-quota-used | service | 58 | V7.0 | Contains the session's access list used quota in bytes. |
| 28 | acl-packet-quota | service | 59 | | Defines the service session packet quota for a time period per a specified access-list measured in number of packets. |
| 29 | acl-packet-quota-used | service | 60 | V90 | Contains the session's used packet quota along with the time period since the quota had been reset. |
| 30 | roaming | service | 61 | V10 | Defines the roaming state that should be set to a connected session |
| 31 | remote-filter-redirect-gw | route | 70 | V5.0 | Defines the remote redirection gateway for packets that do not pass the defined filters. |
| 32 | next-hop | route | 71 | V5.0 | Defines the next-hop router for the user upstream traffic. |

| | | | | | |
|----|--------------------------|-------|----|------|-----------------------------------------------------------------------------------------------------------------------------------|
| 33 | nip-pipe-next-hop | route | 72 | V7.0 | Defines the next-hop router to be used for the traffic destined to a native IP user. |
| 34 | advertise-protocol | route | 73 | V7.0 | Defines the routing protocol to be used to advertise the session IP address. |
| 35 | forward-addr | route | 74 | V90 | Defines the forwarding address. |
| 36 | acl-tcp-nat-redirect | route | 75 | V90 | Defines a destination IP address to which the system should TCP redirect all session packets. |
| 37 | tunnel-id | vpdn | 80 | V5.0 | Defines the tunnel ID, used for LAC purpose. |
| 38 | l2tp-tunnel-password | vpdn | 81 | V5.0 | Contains a password to be used to authenticate to a remote server. |
| 39 | ip-address | vpdn | 82 | V5.0 | Contains the address of the server end of the tunnel. |
| 40 | tunnel-assignment-id | vpdn | 83 | V5.0 | Indicate to the tunnel initiator the particular tunnel to which a session is to be assigned. |
| 41 | tunnel-client-ip-address | vpdn | 84 | V5.0 | Contains the address of the initiator end of the tunnel (LAC IP address). |
| 42 | nativeip | vpdn | 85 | V7.0 | Defines a session as a native IP pipe, means the session acts as tunnel for native IP traffic. |
| 43 | ip-tunnel | vpdn | 86 | V7.0 | Defines the tunneling protocol and the destination tunneling server IP address, which all the session data should be tunneled to. |
| 44 | up-mean-rate | qos | 90 | V5.0 | Specifies the average number of bits per second allowed by the user in the upstream direction. |
| 45 | down-mean-rate | qos | 91 | V5.0 | Specifies the average number of bits per second allowed by the user in the downstream direction. |
| 46 | acl-up-mean-rate | qos | 92 | V7.0 | Specifies the average number of bits per second allowed to the user in the upstream direction per a specified access list. |

| | | | | | |
|----|--------------------|-----|-----|------|------------------------------------------------------------------------------------------------------------------------------|
| 47 | acl-down-mean-rate | qos | 93 | V7.0 | Specifies the average number of bits per second allowed to the user in the downstream direction per a specified access list. |
| 48 | cos | qos | 94 | V7.0 | Defines the class of service that should be set for a specified access list. |
| 49 | acl-priority | qos | 95 | V8.0 | Specifies the Q.o.S priority that should be set for an access list. |
| 50 | ip-primary | dns | 100 | V5.0 | Defines the primary DNS server to be used by the connected peer. |
| 51 | ip-secondary | dns | 101 | V5.0 | Defines the secondary DNS server to be used by the connected peer. |

Hierarchical Attribute Mode

Most of the EDS attributes are operated in hierarchy mode. In this mode, each session includes per each attribute 3 hierarchy-operating level spaces. The first level space is the system default that is being configured, either by management or statically. The second is the user space that is initially being filled in the user authentication phase, and the third is the service space that is being re-filled on each user service change.

The user space level defines the session's "basic" configuration; whereas the service space level is "layered" above it upon a successful dynamic service change. In each level space, the system keeps a set of relevant configurations for that level. The "lifetime" of operation in a service level space is from a successful authentication of that service until a successful authentication of a new service. The "lifetime" of operation in a user level space is the entire period in which the user is authenticated for the session. The effective value of a hierarchy attribute is the most updated value in the highest level space (the highest level for which there is a value defined for the attribute).

USER GROUP

user:accounting sub-attribute

The user:accounting sub-attribute defines the session accounting operation mode and allows the operator to define per each user the accounting methodology. The attribute may be included more than once in request or accept messages. The following operation may be configured:

- **disable** – some operations like symmetric multilink, VPN, or unbilled services do not need the accounting information sent to the RADIUS. This accounting operation mode disables the sending of the accounting information.

The user:accounting sub-attribute is sent as a response to service authentication. It configures the accounting behavior on the received respond. The service default behavior is not to send any accounting records unless the respond includes the enable accounting option.

Accounting information is sent as followed:

| Authentication Response Type | Accounting Behavior |
|------------------------------------------------------------------------------------------------------------------|----------------------------------------|
| <u>Session Authentication</u> Access Accept message includes the user:accounting=disable sub-attribute | Accounting Start and Stop are disabled |
| <u>Service Authentication</u> Access Accept message does not include the user:accounting sub-attribute | Accounting On and Off are disabled |

Scenario Examples:

| Scenario | Action | RADIUS definition | Accounting Behavior |
|----------|-------------------------------------|--------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| 1 | A user is connecting to the network | User definition in the RADIUS includes the user:accounting=disable sub-attribute. | No accounting information is sent (Start and Stop). |
| | A service is selected by the user | Service definition in the RADIUS includes the user:accounting=enable sub-attribute. | Only accounting on and off for the selected service are sent. |
| 3 | A user is connecting to the network | User definition in the RADIUS does not include the user:accounting=disable sub-attribute. | Accounting Start and Stop are sent. |
| | A service is selected by the user | Service definition in the RADIUS does not include the user:accounting sub-attribute. | No accounting on and off are sent. |
| | A service is selected by the user | Service definition in the RADIUS includes the user:accounting=enable sub-attribute. | Accounting on and off are sent for the selected service. |

- **enable** – Enables the sending of the accounting information.
- **lastpacket** – This accounting operation mode enables last packet accounting session time, which is based on the last user's packet timestamps. The last packet accounting calculating method is operated in all session disconnections (such as, "clear user", idle timeout, echo, etc.). Note that this accounting method is relevant for accounting stop and accounting off RADIUS messages.
- **enable-on-ip-update** – This accounting operation mode enables the session accounting on session IP address updating regardless of the current accounting status.
- **interim-update** – Enables the sending of the interim update accounting information for a connected session. In this accounting type, the system sends an accounting message with Acct-Status-Type value 3 (Interim-Update) that includes the same attributes as being sent in accounting stop and off messages, except for the termination cause. The accounting information is being sent every time period that was configured in the received accounting attribute. The accounting:interim-update attribute is operated in hierarchy mode and supports both user and service levels. When received in service authentication it operates only in the service lifetime and sends the service accounting update. When received in user authentication it operates during the whole session lifetime and sends the session accounting update.
- **General:**

Operation Mode: Access-Accept message
 Service-Accept message

Vendor-type: 10

Vendor-length = 2 + name length + (7 | 10)

Format:

```
adc-avpair = "user:accounting=[disable | enable | lastpacket | enable-on-ip-update |
interim-update;<accounting update in seconds>]",
```

Example 1:

```
adc-avpair = "user:accounting=disable",
```

Example 2:

```
gcon-avpair = "user:accounting=interim-update;600",
```

user:orig-name sub-attribute

The **user:orig-name** sub-attribute contains the original user name as received during PPP negotiation. The sub-attribute is sent in Access Request messages, only in operation modes that overwrite or that do not send the original user name. It is used in the following modes:

- domain separator
- service authentication

General:

| | |
|-----------------|---------------------------------------------------|
| Operation Mode: | Access-Request message Service-Request message |
| Vendor-type: | 12 |
| Vendor-length = | 2 + name length + 1-128 |

Format:

```
adc-avpair = "user:orig-name=<original user name>",
```

Example:

```
adc-avpair = "user:orig-name=test",
```

user:auth-type sub-attribute

The user:auth-type sub-attribute contains the authentication type of the Access Request message sent by the system to the RADIUS. It provides the RADIUS with additional information regarding the purpose of the authentication. There are several irregular reasons for RADIUS authentication: pre-authentication, service selection, and web-authentication.

- **pre-authentication** – the system authenticates the peer using the CLI before proceeding with the user connection.
- **service selection** – the system authenticates the requested service in order to receive its definition from the RADIUS.
- **WEB authentication** – the system authenticates the received user-name and password (using the USER-PASS EDS field) with its configured RADIUS.



Note: The system sends this sub-attribute to the RADIUS and not vice versa.

General:

Operation Mode: Access Request message
 Service-Request message

Vendor-type: 13

Vendor-length = 2 + name length + (23 | 32)

Format:

adc-avpair = "user:auth-type=<pre-auth | service-selection | web-auth>",

Example:

adc-avpair = "user:auth-type=pre-auth",

user:action sub-attribute

The user:action sub-attribute defines the action that should be taken by the system.

The actions are:

- a. *Reject* – Reject an authenticated peer or disconnect a connected peer.
- b. *echo* – The system sends echo messages to the connected session. It disconnects the session after 33 seconds if it is not responding for the echo messages. In case the echo did not get any response during the session lifetime the system is not disconnecting the session and is setting the echo status to disable. The echo action is relevant only for native IP sessions and is being ignored in other session types.
- c. *macantispoof* – The system is allowing only one IP address per MAC address. The MAC is the user MAC address as learned by the DHCP relay or by the proxy RADIUS. In case the call trigger is not DHCP or proxy RADIUS the system is setting the mac-anti-spoofing status to disable.
- d. *user_space_overwrite* – The system should update the user space (and not the service space) with all attributes received when this sub-attribute appears. This action is relevant only for Service-Accept messages (service authentication response) and is being ignored when received in session authentication respond. The action may be included once in access accept messages, and the system ignores all other instances.
- e. *user_space_overwrite_on_next_service* – The system should update once the user space (and not the service space) with all next-service attributes when it is being invoked. The system then resets this sub-attribute. When the next-service parameters includes user:action=user_space_overwrite the system should ignore it. The action may be included once in access accept messages, and the system ignores all other instances. This sub-attribute is not working in hierarchy mode.

General:

Operation Mode: Access-Accept message
Service-Accept message

Vendor-type: 14

Vendor-length = 2 + 11 + (4-12)

Format:

```
adc-avpair = "user:action=<Reject | echo | macantispoof | user_space_overwrite |
user_space_overwrite_on_next_service >>",
```

Example:

```
adc-avpair = "user:action=Reject",
```

user: SSC-host sub-attribute

This vendor-specific sub-attribute contains the SSC (Service Selection Center) host IP address at which the user activated the service. By sending this information, the system gives the RADIUS a hint as to the SSC IP address. This is used for authorization purposes. This sub-attribute is used only in service authentication requests.

General:

Operation Mode: Access-Request message
Service-Request message

Vendor-type: 15

Vendor-length = 2 + name length + 7-15

Format:

```
adc-avpair = "user:SSC-host=<SSC host IP address>,"
```

Example:

```
adc-avpair = "user:SSC-host=194.90.1.15",
```

user: service-name sub-attribute

This vendor-specific sub-attribute contains the service name given to the connected peer. The user:service-name sub-attribute represents the service of the currently received attributes. It is a hint given by the RADIUS. When sent by the POPmaestro in an access-request message it implies the service requested by the user.

General:

Operation Mode: Access-Accept message
Access-Request message

Vendor-type: 16

Vendor-length = 2 + name length + (1-128)

Format:

adc-avpair = "user:service-name=<service name>,"

Example:

adc-avpair = "user:service-name=SRV1" ,

user:personal-site sub-attribute

The user:personal-site sub-attribute contains the personal site information of a connected user. The system keeps this vendor-specific Radius attribute for use in EDS query response. The personal-site information maximum size is 256 characters.

General:

Operation Mode: Access-Accept message
Service-Accept message

Vendor-type: 17

Vendor-length = 2 + name length + (1 - 256)

Format:

```
adc-avpair = "user:personal-site=<site URL>",
```

Example:

```
adc-avpair = "user:personal-site=www.walla.co.il",
```

user:mac-address sub-attribute

The user:mac-address sub-attribute contains the MAC address information of a connected user as learned by the DHCP relay or by the proxy RADIUS.

General:

| | |
|-----------------|------------------------------------------------------------------------------------------------------------------------|
| Operation Mode: | Access-Request message Service-Request message Accounting-Stop Request message Accounting-Off Request message |
| Vendor-type: | 18 |
| Vendor-length = | 2 + name length + 12 |

Format:

```
adc-avpair = "user:mac-address=<User MAC address>",
```

Example:

```
adc-avpair = "user:mac-address=00022d386dbe",
```

user:group sub-attribute

The user:group sub-attribute represents the user group number (1 to 1000), as defined in the RADIUS server. This information is sent by the system to the DHCP server via the relay remote ID Sub-option (option 82). The default user group is 1. It may be activated dynamically, enabling the change of the user:group of a connected session on the fly. This sub-attribute when received in service-access accept message, updates both the service and the session user:group.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 19 |
| Vendor-length = | 2 + name length + (1 - 4) |

Format:

```
adc-avpair = "user:group=<group number>",
```

Example:

```
adc-avpair = "user:group=12",
```

user:max-allowed-sessions sub-attribute

The user:max-allowed-sessions sub-attribute defines the maximum number of sessions allowed in a single blade per username. When the system receives this attribute in the authentication process, it checks for the number of concurrent sessions containing the authenticated user-name. If the number of sessions including the current authenticated one, exceeds the number of allowed sessions the system rejects the new incoming session, causing an immediate disconnection.

General:

Operation Mode: Access-Accept message
 Vendor-type: 20
 Vendor-length = 2 + 4 + attribute-name length

Format:

adc-avpair = "user:max-allowed-sessions=<maximum number of sessions per blade>",

Example:

adc-avpair = "user:max-allowed-sessions=1",

user:class sub-attribute

The user:class sub-attribute contains the user class information, a string of maximum size of 256 characters. It is available to be sent by the Radius server to the system in an Access-Accept or Service-Accept messages. The system sends it unmodified to the Radius server as part of the Authentication and Accounting-Requests packets. The user:class sub-attribute operates in hierarchy mode and supports both user and service levels. When received in service authentication, it operates only in the service lifetime and being reset while service is changing. When received in user authentication, it operates during the whole session lifetime.

General:

Operation Mode: Access-Accept message
 Service-Request message
 Service-Accept message
 Accounting on, off, start and stop messages, interim
 Vendor-type: 21
 Vendor-length = 2 + (1-256) + attribute-name length

Format:

adc-avpair = "user:class=<user class data>",

Example:

adc-avpair = "user:class=belong to security group",

user:eds-enc-key sub-attribute

The user:eds-enc-key sub-attribute contains an encryption key for EDS operation. The encryption key should be exactly 16 characters long, comprised solely of characters from the set ("0 - 9", "a - f", "A - F"). Every two characters in the key represent a hexadecimal byte. The bytes should be DES key legal, i.e. each containing an odd number of '1' bits. This key is being used in DES encryption and decryption of the EDS USERPASS field, and overwrites the configured EDS encryption key. The encryption key maximum length is 64 characters. This key is being used in the EDS encryption and decryption and overwrites the configured EDS encryption key. The user:eds-enc-key sub-attribute is operated in hierarchy mode and supports both user and service levels.

General:

Operation Mode: Access-Accept message
Service-Accept message

Vendor-type: 22

Vendor-length = 2 + (1-64) + attribute-name length

Format:

adc-avpair = "user:eds-enc-key=<EDS encryption key>",

Example:

adc-avpair = "user:eds-enc-key=02f804fea90102f8",

user:eds-cookie sub-attribute

The user:eds-cookie sub-attribute contains a user eds cookie data information, a string of maximum size of 64 characters. It is available to be sent by the Radius server to the system in an Access-Accept or Service-Accept messages. The system SHOULD send it unmodified to the Radius server as part of the Authentication and Accounting-Requests packets. This sub-attribute is operated in hierarchy mode and supports both user and service levels. This attribute can also be updated by the SSC (see EDS architecture document).

Operation Mode: Access-Accept message
Service-Request message

Vendor-type: 23

Vendor-length = 2 + (1-64) + attribute-name length

Format:

adc-avpair = "user:eds-cookie=<user eds cookie>",

Example:

adc-avpair = "user:eds-cookie=rt123456",

user:original-url-prefix sub-attribute

The user:original-url-prefix sub-attribute contains a string that should be prefixed by the RDS to the user original requested url when redirecting the user to its personal site. This sub-attribute indicates the RDS that the user original url should be concatenated on the tail of the personal site URL when redirecting the user. The original-url-prefix information maximum size is 64 characters. This sub-attribute is operated in hierarchy mode and supports both user and service levels with one exception. The hierarchy operation mode is not relevant in case a service authentication response includes user:personal-site attribute without user:original-url-prefix. In this case the system should assume there is no original-url-prefix although the user space includes such information.



Note: This feature is implemented in the RDS as well.

Example:

1. The user's session attributes are
 - a.user:personal-site=www.cnn.com
 - b.user:original-url-prefix=?url=
 - c....
2. The user tries to connect to www.yahoo.com

The user is being redirected to:www.cnn.com?url=www.yahoo.com

Operation Mode: Access-Accept message
Service-Accept message

Vendor-type: 24

Vendor-length = up to 64 + attribute-name length

Format:

```
adc-avpair = "user:original-url-prefix=<prefixed string>",
```

Example:

```
adc-avpair = "user:original-url-prefix=?url=",
```

DHCP GROUP

dhcp:dhcp-server sub-attribute

The dhcp:dhcp-server attribute defines the DHCP server IP address, which the system should relay the user's DHCP requests. It may be activated dynamically, enabling the change of DHCP server IP of a connected user on the fly. This dynamic capability is allowed when the agent-id override mode is enabled. This sub-attribute when received in service-access accept message, updates both, the service and the session user:group.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 30 |
| Vendor-length = | 2 + name length + (7 + 15) |

Format:

```
adc-avpair = "dhcp:dhcp-server=<DHCP Server IP>",
```

Example:

```
adc-avpair = "dhcp:dhcp-server=194.90.1.5",
```

dhcp:opt82-relay-remote-id sub-attribute

The dhcp:opt82-relay-remote-id attribute contains the received option 82 relay remote ID sub-option, while each byte information is in hexadecimal format.

General:

Operation Mode: Access-Accept message
Service-Accept message

Vendor-type: 31

Vendor-length = 2 + name length + (1 - 255)

Format:

adc-avpair = "dhcp:opt82-relay-remote-id=<Option 82 relay remote ID sub-option>",

Example:

adc-avpair = "dhcp:opt82-relay-remote-id=01844400660300",

dhcp:discover-action sub-attribute

The dhcp:discover-action attribute defines the action to be taken when a new dhcp discover message is transmitted in a connected session. The discover actions are:

- a. update - The system is not disconnecting the session on a new dhcp discover message and updates it with the new configured IP address.
- b. normal - The system disconnects the session on a new dhcp discover message, which is the system normal behavior.

General:

Operation Mode: Access-Accept message
Service-Accept message

Vendor-type: 32

Vendor-length = 2 + 6 + name length

Format:

adc-avpair = "dhcp:discover-action=<normal | update>",

Example:

adc-avpair = "dhcp:discover-action=update",

PROTOCOL GROUP

protocol:type sub-attribute

There is a need in Access Request messages to receive a hint of the protocol negotiated with the peer. The protocol:type sub-attribute fulfills this need and enables the operator to manage the connections. The system sends this sub-attribute to the RADIUS when detecting a connection-negotiating multilink.



Note: In multi-link calls, this sub-attribute is sent per each link.

General:

| | |
|-----------------|------------------------|
| Operation Mode: | Access-Request message |
| Vendor-type: | 40 |
| Vendor-length = | 2 + name length + 3 |

Format:

```
adc-avpair = "protocol:type=[mlp]" ,
```

Example:

```
adc-avpair = "protocol:type=mlp" ,
```

SERVICE GROUP

service:service-timeout

This sub-attribute defines the service session timeout measured in seconds. When a service session timeout event occurs the system activates the next-service-name as defined in the next-service-name sub-attribute (section 2.3.2). A service:service-timeout attribute with a zero value indicates the POPmaestro that the connected session should be logged off. The attribute may be included only once in access-accept message. When appearing more than once the system will consider the last attribute appearance.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 50 |
| Vendor-length = | 2 + name length + (1 - 10) |

Format:

```
adc-avpair = "service:service-timeout=<timeout in seconds>" ,
```

Example:

```
adc-avpair = "service:service-timeout=500" ,
```


service:next-service-name

This vendor specific sub-attribute defines the name of the next service to use when a service "session timeout" expires. The system then checks if the authentication base is of service type and a next service is configured for the session. In that case it activates the next service. It would do so in the same way it would activate a new service received directly from a service selection center. If no next service name is configured the system activates a static service called "GServiceTimeout". If no service timeout is configured, the system immediately activates the received next service. The attribute may be included only once in access-accept message. When appearing more than once the system will consider the last attribute appearance.

**Note:**

1. After activating the next-service the system has no configured next service. This is true unless the activated next-service includes the next-service-name sub-attribute as well.
2. The service timeout sub-attribute is defined per link.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 51 |
| Vendor-length = | 2 + name length + (1 - 128) |

Format:

adc-avpair = "service:next-service-name=<service name>",

Example:

adc-avpair = "service:next-service-name=SRV12",

service:auto-service-name

The service:auto-service-name sub-attribute contains the service name to be automatically activated when the user is redirected by the RDS. The system keeps the auto-service-name information for use in EDS query response. The system deletes the auto-service-name information after sending the query response, since it should be used only once. The attribute may be included only once in access-accept message. When appearing more than once the system will consider the last attribute appearance.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 52 |
| Vendor-length = | 2 + name length + (1-128) |

Format:

adc-avpair = "service:auto-service-name=<service name>",

Example:

adc-avpair = "service:auto-service-name=SRV12",

service:auth-source

This sub-attribute defines the source name to be used when the POPmaestro authorizes or authenticates a service with the RADIUS. The POPmaestro performs a RADIUS access request when a service is activated. The service:auth-source attribute defines the source name to be used as the user-name in this request. The authentication source can have one of the following values: user, service, or CLI. When the value is not service, the POPmaestro includes the requested service name in the RADIUS access request message (see user:service-name attribute). The authentication source default value is service. The attribute may be included only once in access-accept message. When appearing more than once, the system will consider the last attribute appearance.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 53 |
| Vendor-length = | 2 + name length + (3 - 7) |

Format:

```
adc-avpair = "service:auth-source=<service | user | CLI>" ,
```

Example:

```
adc-avpair = "service:auth-source=CLI" ,
```

service:data-quota

This sub-attribute defines the service session data quota measured in bytes. The POPmaestro monitors the session to track the data quota usage. When a service runs out of quota (session quota termination event occurs), the system activates the next-service-name if defined (see next-service-name definition). This is true in case that the service authentication base is set to be service. Otherwise the POPmaestro activates reauthorization according to the configured service:auth-base value (see service:auth-base definition). A service:data-quota attribute with a zero value indicates the POPmaestro that the connected session has no data credit left and should be logged off. The POPmaestro then disconnects the session. When a session is disconnected or when the service is exchanged, the POPmaestro includes the service:data-quota attribute in the accounting message with the remainder data credit. The attribute may be included only once in access-accept message. When appearing more than once the system will consider the last attribute appearance.

General:

| | |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Operation Mode: | Access-Accept message Access-Request message Service-Accept message Service-Request message Accounting-Request messages |
| Vendor-type: | 54 |
| Vendor-length = | 2 + name length + (1 - 10) |

Format:

```
adc-avpair = "service:data-quota=<data quota in bytes>" ,
```

Example:

```
adc-avpair = "service:data-quota=5000000" ,
```

service:data-quota-used

The service:data-quota-used contains the session's used quota in bytes. It is being sent only if a quota has been established for the session. It is being sent in authorization requests and Accounting-Stop and Accounting-Off messages.

General:

| | |
|-----------------|-----------------------------------------------------------------------------|
| Operation Mode: | Access-Request message Accounting-Stop message Accounting-Off message |
| Vendor-type: | 57 |
| Vendor-length: | 1 - 19 + attribute-name length |
| Values: | Min = 0; Max = $2^{63} - 1$ |

Format:

```
adc-avpair = "service:data-quota-used=<used data quota in bytes>" ,
```

Example:

```
adc-avpair = "service:data-quota-used=5000000" ,
```

service:acl-data-quota

The service:acl-data-quota sub-attribute defines the service session data quota per a specified access-list measured in bytes. This attribute enables to define different data quota per each access-list of a session. The POPmaestro monitors the session to track the data quota usage for the defined access lists. When a service runs out of quota (access-list quota termination event occurs) the POPmaestro activates the next-service-name if defined (see next-service-name definition). This is true in case that the service authentication base is set to be service. Otherwise the POPmaestro activates reauthorization according to the configured service:auth-base value (see service:auth-base definition). The authorization requests include the service:acl-data-quota with the remainder quota. A service:accessl-list-data-quota attribute with a zero value indicates to the POPmaestro that the connected session has no access-list data credit left. The POPmaestro then redirects all the session packets that match the specific access-list, to the redirection server if one is configured. Otherwise, it discards the packets. When a session is disconnected or when the service is exchanged, the POPmaestro includes all the service:acl-data-quota attributes in the accounting messages with the remainder data credit. The attribute may be included more than once in request or accept messages.

General:

| | |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Operation Mode: | Access-Accept message Access-Request message Service-Accept message Service-Request message Accounting-Request messages |
| Vendor-type: | 55 |
| Vendor-length = | 2 + name length + (1 - 10) |
| Values: | Min = -1; Max = $2^{63} - 1$ |

Format:

```
adc-avpair = "service:acl-data-quota=<access-list number>;<data quota in bytes>",
```

Example:

```
adc-avpair = "service:acl-data-quota=101;5000000",
```

service:service-cache

The service:service-cache sub-attribute contains the service caching operation mode of the received service. The system when receiving this attribute along with the service definition clears the service cached information if exists. The system ignores this attribute when received in session authentication respond.

General:

| | |
|-----------------|------------------------|
| Operation Mode: | Service-Accept message |
| Vendor-type: | 56 |
| Vendor-length: | 24 |

Format:

```
adc-avpair = "service:service-cache=<off>",
```

Example:

```
adc-avpair = "service:service-cache=off",
```

service:acl-data-quota-used

The service:acl-data-quota-used attribute contains the session's used quota in bytes. It is being sent only if a quota has been established for a specific session access list. It is being sent in authorization requests and Accounting-Stop and Accounting-Off messages.

General:

| | |
|-----------------|-----------------------------------------------------------------------------------------------------------------|
| Operation Mode: | Access-Request message Service Request message Accounting-Stop Accounting-Off, interim request message |
| Vendor-type: | 58 |
| Vendor-length: | 1 - 19 + attribute-name length |
| Values: | Min = 0; Max = 2 ⁶³ - 1 |

Format:

adc-avpair = "service:acl-data-quota-used=[access-list name;]<used data quota in bytes>",

Example:

adc-avpair = "service:acl-data-quota-used=video;5000000",

service:acl-packet-quota

The service:acl-list-packet-quota sub-attribute defines the service session packet quota for a time period per a specified access-list measured in number of packets (total packets of both upstream and downstream user's traffic that were permitted by the filter). This attribute enables to define different packet quota per each access-list of a session.

- a. The system monitors the session to track the packet quota usage in a certain time period for the defined access lists.
- b. When a service runs out of packet quota within the configured period (access-list quota termination event occurs) the system activates the next-service-name if defined (see next-service-name definition). This is true in case that the service authentication base is set to be service. Otherwise the system activates reauthorization according to the configured service:auth-source value (see service:auth-source definition).
- c. When a service runs out of packet quota after the configured period the system resets the quota for its initial value.
- d. All authorization requests include the service:acl-packet-quota with the remainder quota in the current interval.
- e. A service:accessl-list-packet-quota attribute with a zero value indicates to the SG-1 that the connected session has no access-list packet credit left.
- f. The SG-1 then redirects all the session packets that match the specific access-list, to the redirection server if one is configured. Otherwise, it discards the packets.
- g. The attribute may be included more than once in request or accept messages. The system does not consider packets that were denied.

General:

| | |
|-----------------|---------------------------------------------------|
| Operation Mode: | Access-Request message Service-Request message |
| Vendor-type: | 59 |
| Vendor-length: | 1 - 19 + attribute-name length |
| Values quota: | Min = -1; Max = $2^{23} - 1$ |
| Time: | Min = 0; Max = $2^{32} - 1$ |

Format:

ADC-avpair = "service:acl-packet-quota=<access-list name>;<packet quota>;<quota period in seconds>",

Example:

ADC-avpair = "service:acl-packet-quota=mail;100;300",

service:acl-packet-quota-used

This sub-attribute contains the session's used packet quota along with the time period since the quota had been reset.

- The service:acl-packet-quota-used attribute is being sent only if quota have been established for the session.
- The attribute is being sent in SG-1 authorization requests, Accounting-Stop, Accounting-Off and interim messages.
- It should include the access-list name for access-list-packet-quota attributes.
- The sent time period is the time since the last packet arriving on that ACL.

General:

| | |
|-----------------|-----------------------------------------------------------------------------------------------------------|
| Operation Mode: | Access-Request message Service-Request message Accounting-Stop, off and interim request messages |
| Vendor-type: | 60 |
| Vendor-length: | 1 - 19 + attribute-name length |
| Values quota: | Min = 0; Max = $2^{31} - 1$ |
| Time: | Min = 0; Max = $2^{32} - 1$ |

Format:

adc-avpair = "**service:acl-packet-quota-used**=*access-list name*; <*used packet quota*>; <*time period since last ACL packet*>",

Example:

adc-avpair = "**service:acl-packet-quota-used**=mail;100;245",

service:roaming

This sub-attribute defines the roaming state that should be set to a connected session. It overwrites the system-roaming configuration. The attribute is operated in hierarchy mode and supports both user and service levels.

General:

| | |
|-----------------|--------------------------------------------------|
| Operation Mode: | Access-Request message Service-Accept message |
| Vendor-type: | 61 |
| Vendor-length: | 1 - 197+ attribute-name length |

Format:

adc-avpair = "**service:roaming**=<*disable*>",

Example:

adc-avpair = "**service:roaming**=*disable*",

ROUTE GROUP

route:remote-filter-redirect-gw

This vendor specific sub-attribute defines the remote redirection gateway for redirecting the packets that did not pass the defined filters. It also works dynamically and allows changing the user-redirected gateway on the fly. Upon receipt of this sub-attribute the system tunnels the user data to the remote Redirection Gateway using the IP in IP tunnel protocol. When a redundant RDS is configured the system uses it, if the first prioritized RDS is not responding.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 70 |
| Vendor-length = | 2 + name length + (7-15) |

Format:

```
adc-avpair = "route:remote-filter-redirect-gw=<gateway IP address>[,<redundant
gateway IP address]>" ,
```

Example:

```
adc-avpair = "route:remote-filter-redirect-gw=192.168.1.23,192.168.1.24" ,
```

route:next-hop

This sub-attribute defines the next-hop router to be used for the user connection upstream traffic. It may be activated dynamically, enabling the change of the next-hop router of a connected user on the fly.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 71 |
| Vendor-length = | 2 + name length + (7-15) |

Format:

```
adc-avpair = "route:next-hop=<Next-hopIP address>" ,
```

Example:

```
adc-avpair = "route:next-hop=192.168.1.23" ,
```

route:nip-pipe-next-hop

The route:nip-pipe-next-hop attribute defines the next-hop router to be used for the traffic destined to a native IP user. The system when receiving this attribute directs the traffic through the native IP PIP to the defined next-hop. The attribute may be sent in Access-Accept message and operated during the session lifetime and it overwrites the current native IP pipe next-hop of the user session. It may be included once in access accept messages, and the

system ignores all other instances. Note that the next hop must be directly connected otherwise the packets will be discarded.



Note: The attribute is relevant only for native IP traffic over Ethernet.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 72 |
| Vendor-length = | 2 + attribute length + (1-15) |

Format:

adc-avpair = "route:nip-pipe-next-hop=<Next-hop IP address>",

Example:

adc-avpair = "route:nip-pipe-next-hop=192.168.1.23",

route:advertise-protocol attribute

The route:advertise-protocol sub-attribute defines the routing protocol to be use to advertise the session IP address. The system when receiving this attribute advertises the session IP address via the received protocol regardless to the system default routing definition.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 73 |
| Vendor-length = | 4-5 + attribute length |

Format:

adc-avpair = "route:advertise-protocol=<ripv2 | ospf>",

Example:

adc-avpair = "route:advertise-protocol=ospf"

route:forward-addr attribute

The route:forward-addr sub-attribute defines the forwarding address. This address is indicating that packets for the external destination (user IP address) should be forwarded to the specified forwarding address by the downstream router (not to the system). This attribute may be received dynamically during the session lifetime – It supports hierarchy operation mode. It may be included once in access accept messages, and the system ignores all other instances. This attribute is valid only when route:advertise-protocol attribute is defined, otherwise the system should ignore it. The forward-addr default value is 0.0.0.0. This attribute is supported only for OSPF advertising protocol.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 74 |
| Vendor-length = | 2 + 7-15 + attribute length |

Format:

adc-avpair = "route:forward-addr=<IP address>",

Example:

adc-avpair = "route:forward-addr=192.168.1.4"

route:acl-tcp-nat-redirect attribute

The route:acl-tcp-nat-redirect attribute defines a destination IP address to which the system should TCP redirect all session packets. In this case the system should perform NAT redirection for all TCP packets that meet the access-list definition (replacing the destination IP for upstream traffic and replacing it back for the downstream traffic). The network address translation is performed on the last destination of an upstream packet belonging to this access list flow. This sub-attribute is operated in hierarchy mode and supports both user and service levels.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 75 |
| Vendor-length = | 2 + 7-15 + attribute length |

Format:

adc-avpair = "route:acl-tcp-nat-redirect=<access list name>;<IP address>",

Example:

adc-avpair = "route:acl-tcp-nat-redirect=SMTP;192.168.1.4"

VPDN GROUP**vpdn:tunnel-id attribute**

This attribute defines the tunnel ID, used for LAC purpose. This attribute is mandatory for opening a tunnel session.

General:

| | |
|-----------------|----------------------------|
| Operation Mode: | Access- Accept message |
| Vendor-type: | 80 |
| Vendor-length = | 2 + name length + (1 - 64) |

Format:

adc-avpair = "vpdn:tunnel-id=<username>",

Example:

adc-avpair = "vpdn:tunnel-id=test",

vpdn:l2tp-tunnel-password attribute

This Attribute contains a password to be used to authenticate to a remote server. This attribute is mandatory for opening a tunnel session.

General:

Operation Mode: Access- Accept message
Vendor-type: 81
Vendor-length = 2 + name length + (1-64)

Format:

adc-avpair = "vpdn:l2tp-tunnel-password=<password>",

Example:

adc-avpair = "vpdn:l2tp-tunnel-password=test",

vpdn:ip-address attribute

This attribute indicates the address of the server end of the tunnel. This attribute is mandatory for opening a tunnel session. Attribute type is string.

General:

Operation Mode: Access- Accept message
Vendor-type: 82
Vendor-length = 2 + name length + (7-15)

Format:

adc-avpair = "vpdn:ip-address=<x.x.x.x>",

Example:

adc-avpair = "vpdn:ip-address=192.168.4.3",

vpdn:tunnel-assignment-id attribute

This attribute is used to indicate to the tunnel initiator the particular tunnel to which a session is to be assigned. The system when receiving this attribute checks if the assigned tunnel exists, if not the tunnel is created. Termination of the last session in the tunnel causes the closing of the tunnel. This attribute is optional, while not used a new tunnel will be open per each session. The values used for tunnel-assignment-id are 1 to 10,000. When receiving a tunnel-assignment-id greater than 10,000 the system ignores it and sends a warning message to the log.

General:

Operation Mode: Access-Accept message
Vendor-type: 83
Vendor-length = 2 + (7-15)+attribute-name length

Format:

adc-avpair = "vpdn:tunnel-assignment-id=<id>",

Example:

adc-avpair = "vpdn:tunnel-assignment-id=2",

vpdn:tunnel-client-ip-address

This attribute contains the address of the initiator end of the tunnel (LAC IP address). It enables the operator to distinguish between users that accessed the network from different access servers.

General:

| | |
|-----------------|--------------------------|
| Operation Mode: | Access-Request message |
| Vendor-type: | 84 |
| Vendor-length = | 2 + name length + (7-15) |

Format:

```
adc-avpair = "vpdn:tunnel-client-ip-address-=<tunnel client ip address>",
```

Example:

```
adc-avpair = "vpdn:tunnel-server-client-ip-address=192.168.3.5",
```

vpdn:nativeip sub-attribute

This attribute defines a session as a native IP pipe, meaning the session acts as a tunnel for native IP traffic. It is currently used for PPP sessions only. It may be activated dynamically, enabling the change of a connected PPP session Native IP status on the fly.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 85 |
| Vendor-length = | 2 + name length + 3 |

Format:

```
gcon-avpair = "vpdn:nativeip=<ppp>",
```

Example:

```
gcon-avpair = "vpdn:nativeip=ppp",
```

vpdn:ip-tunnel sub attribute

The vpdn:ip-tunnel sub attribute defines the tunneling protocol and the destination tunneling server IP address, which all the session data should be tunneled to. It may be activated dynamically, enabling to change the session tunneling protocol and destination on the fly. The attribute may be included once in access accept messages, and the system ignores all other instances.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 86 |
| Vendor-length = | 2 + 7-15 + attribute-name length |

Format:

```
adc-avpair = "vpdn:ip-tunnel=<gre;IP address[,redundant IP address]|ip-in-ip;IP  
address [,redundant IP address]>",
```

Example 1:

```
adc-avpair = "vpdn:ip-tunnel=gre;192.168.1.34",
```

Example 2:

```
adc-avpair = "vpdn:ip-tunnel=gre;192.168.1.34,192.168.4.1",
```

QOS GROUP

qos:up-mean-rate

The qos:up-mean-rate sub-attribute specifies the average number of bits per second allowed by the user in the upstream direction. It is sent in an Access-Accept message and it overwrites the current upstream rate allocated to the user. This attribute may be activated during a session lifetime.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 90 |
| Vendor-length = | 2 + name length + (1-128) |

Format:

```
gcon-avpair = "qos:up-mean-rate=<up mean rate in Kbits>",
```

Example:

```
gcon-avpair = "qos:up-mean-rate=128",
```

qos:down-mean-rate

The qos:down-mean-rate sub-attribute specifies the average number of bits per second allowed by the user in the downstream direction. It is sent in an Access-Accept message and it overwrites the current downstream rate allocated to the user. This attribute may be activated during a session lifetime.

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 91 |
| Vendor-length = | 2 + name length + (1-64) |

Format:

```
gcon-avpair = "qos:down-mean-rate=<down mean rate in Kbits>",
```

Example:

```
gcon-avpair = "qos:down-mean-rate=256",
```

qos:acl-up-mean-rate sub attribute

The qos:acl-up-mean-rate sub-attribute specifies the average number of bits per second allowed to the user in the upstream direction per a specified access list. It is sent in Access-Accept message and it overwrites the current access list upstream rate of the user. This attribute may be operated during the session lifetime (EDS).

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 92 |
| Vendor-length: | 1-128 + attribute-name length |

Format:

adc-avpair = "qos:acl-up-mean-rate=<access list name>;<up mean rate in Kbits>",

Example:

adc-avpair = "qos:acl-up-mean-rate=acl1;128",

qos:acl-down-mean-rate sub attribute

The qos:acl-down-mean-rate sub-attribute specifies the average number of bits per second allowed to the user in the downstream direction per a specified access list. It is sent in Access-Accept message and overwrites the current access list downstream rate of the user. This attribute may be operated during the session lifetime (EDS).

General:

| | |
|-----------------|-------------------------------------------------|
| Operation Mode: | Access-Accept message Service-Accept message |
| Vendor-type: | 93 |
| Vendor-length: | 1-128 + attribute-name length |

Format:

adc-avpair = "qos:acl-down-mean-rate=<access list name>;<down mean rate in Kbits>",

Example:

adc-avpair = "qos:acl-down-mean-rate=acl1;256",

qos:cos

The qos:cos sub-attribute defines the class of service that should be set for a specified access list. The system sets the defined COS value for all the session's packets that matched the access list. This attribute may be included more than once in request or accept messages. The attribute is being ignored and not being saved by the system, whenever a specified access list does not exist in the session's definitions.

General:

| | |
|-----------------|------------------------------|
| Operation Mode: | Access-Accept message |
| Vendor-type: | 94 |
| Vendor-length: | 1-16 + attribute-name length |

Format:

adc-avpair = "qos:cos=<access-list name>;<COS value>",

| Parameter | Description | Values |
|--------------------|------------------|-----------------------------------|
| <access list name> | Access list name | Up to 16 alphanumeric characters. |
| <COS> | Class of service | |
| Mandatory | Number - 0 to 63 | |

Example:

The system sets DIFFSERV field to 12 of all the packets that passed access-list video.

```
Filter-Id = "video out permit 192.168.1.0 255.255.255.0 12",
adc-avpair = "qos:cos=video;12"
```

qos:acl-priority sub attribute

The qos:acl-priority sub-attribute specifies the Q.o.S priority that should be set for an access list within a session. A default access list priority is 0. The system precedes the high priority packets on the low priority ones. The priority is being activated only if the qos:down-mean-rate or/and qos:up-mean-rate are defined for the connected session, otherwise it is being ignore.



Note: If the higher priority traffic is exceeding the session bandwidth limitation, then starvation of lower priority traffic may occur. To prevent lower priority traffic starvation, the higher priority access list bandwidth limitation should be smaller that the session one.

General:

Operation Mode: Access-Accept message
 Service-Accept message

Vendor-type: 95

Vendor-length: 1-128 + attribute-name length

Format:

```
adc-avpair = "qos:acl-priority=<access list name>;<priority number>,"
```

| Parameter | Description | Values |
|--------------------|----------------------------------------------------------------------------|-----------------------------------|
| <access list name> | Access list name | Up to 16 alphanumeric characters. |
| <Priority value> | Access list priority, while 0 is the lowest priority and 10 is the highest | |
| Mandatory | Number - 0 to 10 | |

Example:

```
adc-avpair = "qos:acl-priority=acl1;1",
```

DNS GROUP

dns:ip-primary

The dns:ip-primary attribute defines the primary DNS server to be used by the connected peer.

General:

| | |
|-----------------|--------------------------|
| Operation Mode: | Access-Accept message |
| Vendor-type: | 100 |
| Vendor-length = | 2 + name length + (7-15) |

Format:

```
adc-avpair = "dns:ip-primary=<Primary DNS IP>",
```

Example:

```
adc-avpair = "dns:ip-primary=194.90.1.5",
```

dns:ip-secondary

The dns:ip-secondary attribute defines the secondary DNS server to be used by the connected peer.

General:

| | |
|-----------------|--------------------------|
| Operation Mode: | Access-Accept message |
| Vendor-type: | 101 |
| Vendor-length = | 2 + name length + (7-15) |

Format:

```
adc-avpair = "dns:ip-secondary=<Primary DNS IP>",
```

Example:

```
adc-avpair = "dns:ip-secondary=194.90.1.5",
```

REDIRECTION SERVER

The Redirection Server (RDS) is an ADC's product that redirects all peers' Http requests to their personal-sites as pre-defined in the Radius server. The RDS uses ADC's EDS (Enhanced Dynamic Services) policy to redirect the connected peers and it actually acts as a sophisticated SSC.

The RDS is usually located at the ISP/Carrier network but it is not mandatory. When the RDS is located outside the access subnet the SG-1 uses an IP in IP tunnel to transfer the redirected data.

Using the system event logging information includes the Http URL requests.

Usage

<Date>|<Source IP>|<Requested URL>|<URL type>

Parameter(s)

<Date>

The date should be in the following format: DD/MM/YYYY HH:MM

<Source IP>

The source IP address of the requestor

<Requested URL>

The redirected URL string

<URL type>

May be PS or DEFAULT

Example:

01/03/2001 21:54|192.168.5.8|http://www.yahoo.com|PS

Using password command

The password command modifies the login password:

- It is located at the "configure terminal" menu.
- Technician user may change all passwords. The operator user may change its own password only.
- The password is encrypted using DES algorithm. Write terminal command displays the encrypted values.

Usage

password <user type> <password>

Parameter(s)

<user type>

It is the user type to change the password by operator or technician.

<password>

It is the new password, 6 to 64 alphanumeric characters.

Example:

```
RDSHost> configure terminal <cr>
RDSHost(config)# password operator test11
```

Using default-redirection-site command

- The default-redirection-site command defines the site to which the user will be directed when the personal-site information is not available.
- Initial value for this field is "http://<RDS IP address>/RDS.html". When RDS IP address is not configured the initial value is "http://RDS.html".
- The command located at the "configure terminal" menu.

Usage

default-redirection-site <URL name>

Parameter:

<URL name>

Full URL name to use as default redirection site, it is 1 to 255 alphanumeric characters.

Example:

```
RDSHost> configure terminal <cr>
RDSHost(config)# default-redirection-site http://www.yahoo.com
```

ORUP COMMANDS

Using ORUP (Original Requested URL Prefix)

Usage

<user personal site><ORUP field><original URL request>

Parameter:

<ORUP field>

This is the ORUP field value.

Example

- The ORUP field value is: **?url=**
- The user tries to connect to www.yahoo.com
- The user personal site is: www.cnn.com

The user is being redirected to:
www.cnn.com?url=www.yahoo.com

Using orup-active command

- The orup-active command activates the ORUP (Original Requested URL Prefix) functionality.
- The command located at the “configure terminal” menu.

Usage

orup-active

Example:

```
RDSHost> configure terminal <cr>  
RDSHost(config)# orup-activate
```

Using no orup-active command

- The no orup-active command de-activates the ORUP (Original Requested URL Prefix) functionality.
- The command located at the “configure terminal” menu.

Usage

no orup-active

Example:

```
RDSHost> configure terminal <cr>  
RDSHost(config)# no orup-activate
```

SERVICE NAME COMMANDS

Using service-name command

- The service-name command defines the service to be operated for all sessions.
- The command located at the “configure terminal” menu.

Usage

service-name <service name>

Example:

```
RDSHost> configure terminal <cr>
RDSHost(config)# service-name srv17
```

Using no service-name command

- The no service-name command disables the operating of a service for all the sessions.
- The command located at the “configure terminal” menu.

Usage

no service-name

Example:

```
RDSHost> configure terminal <cr>
RDSHost(config)# no service-name
```

Using event-level command

- This command defines the event level to log and the media to use for logging.
- The command is located at the “configure terminal” menu.

Usage

event-level <NUM> **output-device** <Logging media>

Parameter(s)

<NUM>

It is the event level number. It is a number between 0-1000 Initial value 0.

<Logging media>

The media to use for logging, console: sys-Logger none, Initial value **none**

Example:

```
RDSHost> configure terminal <cr>
RDSHost(config)# Event-level 1 output-device log-file
```

TFTP COMMANDS

Using copy-TFTP command

- The copy-TFTP command enables the upgrading of new RDS software.
- The new upgraded software is affecting after reloading the system.
- It is located at the “main” menu.
- It upgrades only standard RDS pack files.
- Download process reports the following errors:
 - a. Download timeout
 - b. File not found
 - c. Download hardware problem
 - d. Download pack file is corrupted

Usage

copy-TFTP flash <IP address> <File name>

Parameter(s):

<IP address>

This is the IP address of the TFTP server.

<File name>

This is the file name to download. Valid name: 1 to 255 alphanumeric characters.

Example 1: Successful software download

```
RDSHost> copy-TFTP flash 192.168.1.4 RDS.pack <cr>
      Download in progress ...

      Pack loaded successfully
      In order to use new software, please reload the system

RDSHost>
```

Example 2: Unsuccessful software download

```
RDSHost> copy-TFTP flash 192.168.1.4 test.doc <cr>
      Download in progress ...

      Pack loaded unsuccessfully due to: Download pack file is corrupted

RDSHost>
```

Using copy-TFTP flash def-redirection-page command

- The copy-TFTP flash def-red-page command enables the updating of the default redirection page (RDS.HTML).
- It is located at the “ADC_Mode” menu.
- Download process reports the following errors:
 - a. Download timeout
 - b. File not found
 - c. Download hardware problem

Usage

copy-TFTP flash def-redirection-page <IP address> <File name>

Parameter(s)

<IP address>

This is the IP address of the TFTP server.

<File name>

This is the file name to download. Valid name: 1 to 255 alphanumeric characters.

Example 1: Successful software download

```
RDSHost> copy-TFTP flash def-redirection-page 192.168.1.4 RDS.pack <cr>
      Download in progress ...

      File loaded successfully
RDSHost>
```

Example 2: Unsuccessful software download

```
RDSHost> copy-TFTP flash def-redirection-page 192.168.1.4 test.doc <cr>
      Download in progress ...

      File loaded unsuccessfully due to: Download pack file is corrupted
RDSHost>
```

Using syslog-server-ip command

- The syslog-server-ip command defines the IP address of the syslog server.
- It is located at the “configure terminal” menu.
- SysLog server IP initial value is 0.0.0.0.

Usage

syslog-server-ip <sysLog server IP>

Parameter

<sysLog server IP>

This is the IP address of the syslog server.

Example:

```
RDSHost> syslog-server-ip 192.168.1.8 <cr>
RDSHost>
```

SHOW COMMANDS

Using show version command

- The show version command displays the software and hardware versions.
- It is located at the “main” menu.

Usage

show version <software | hardware | pack>

Parameter(s)

<software>

It shows the current system software version.

<hardware>

It shows the current system hardware version.

<pack>

It shows the current system software pack.

Example 1:

```
RDSHost> show version hardware <cr>
Module Num Part No. Serial No. Version
-----
RDS      1 780-002 1234567 V1.0_RDS_1000_128
RDSHost>
```

Example 2:

```
RDSHost> show version software <cr>
Module      Num  Version
-----
EDS          1  V1.0_RDS October 29 2001 19:49:43
Kernel      2  V2.4   October 29 2001 15:00:10
```

Example 3:

```
RDSHost> show version pack <cr>
Pack Image Name: V1.0_RDS_pack
Version: V1.0_RDS October 29 2001 19:53:37
Size: 1537790 byte
Pack Components:
Kernel Version: V2.4 October 29 2001 15:00:10
EDS.php Version: V1.0 October 29 2001 19:49:43
RDSHost>
```

Using show configuration command

- The show configuration command displays RDS configuration.
- It is located at the “main” menu.

Usage**show configuration****Example:**

```
RDSHost>
RDSHost> show config <cr>
# version: V1.0_RDS Apr 25 2002 15:44:56
# Last saved: 25 May 2002 10:10:30

event-level 1 output-device sys-logger
default-redirection-site http://192.168.1.5/RDS.html
interface Ethernet 1 192.168.1.4 255.255.255.03
ip default-gateway 192.168.1.1

RDSHost>
```

Using show system command

- The show system command displays the system information of the RDS.
- It is located at the “main” menu.
- The command presents the following information:
 - a. System time
 - b. Http hits statistics
 - c. RDS current CPU usage

Usage**show system**

Example:

```
RDSHost>
RDSHost> show system
Up-time: 0 Hours, 38 Minutes, 23 Seconds
CPU usage: 50 %
Average Http hits per minute:8681
Accumulated Http hits on port 80:19993456
Accumulated Http hits on EDS port:39986912
RDSHost>
```

ETHERNET COMMANDS**Using interface Ethernet command**

- The Ethernet command is used for configuring the RDS IP address.
- It is located at the “configure terminal” menu.
- The change is affecting immediately.

Usage

interface Ethernet 1 <RDS IP address> <RDS mask> [**mode** <Ethernet mode>]

Parameter(s)

<RDS IP address>

This is the IP address of the Ethernet interface of the Redirection Server.

<RDS mask>

This is the mask of the Ethernet interface of the Redirection Server.

<Ethernet mode>

This is the Ethernet mode and it is one of the following: Auto, 10H, 10F, 100H, 100F, 1000H(negotiate), 1000F (negotiate). The default mode is Auto.

Example:

```
RDSHost>
RDSHost> config termin <cr>
RDSHost(config)> interface Ethernet 1 192.168.2.5 255.255.255.0 <cr>
RDSHost(config)>
```

Using no interface Ethernet command

- The command is located at the “configure terminal” menu.
- It deletes the configured interface.
- The change is affecting immediately.

Usage

no interface Ethernet <Interface number>

Parameter

<Interface number>

It is the interface number 1.

Example:

```
Host(config)# no interface Ethernet 1 <cr>
Host(config)#
```

The system should warn when configuring the Ethernet interface with subnet, which contradicts the default gateway.

Example 1:

```
Host(config)# interface Ethernet 1 192.168.2.5 255.255.255.0 <cr>
      Operation Warning:
      The default gateway is out of subnet
Host(config)# end
Host# write terminal
Interface Ethernet 1 192.168.2.5 255.255.255.0
...
```

DEFAULT GATEWAY COMMANDS

Using ip default-gateway command

- The command is located at the “configure terminal” menu.
- It configures the system default gateway.
- The change is affecting immediately.

Usage

ip default-gateway <gateway IP address>

Parameter

<gateway IP address>

It is the IP address of the Default gateway.

Using no ip default-gateway command

- The command is located at the “configure terminal” menu.
- It deletes the system default gateway.
- The change is affecting immediately.

Usage

no ip default-gateway

Example:

```
Host(config)# no ip default-gateway <cr>
Host(config)#
```

When assigning a default-gateway that is not in the subnet of the Ethernet interface, the system should ignore the command and indicates the reason.

Format

Operation Error:

Default-gateway is out of subnet.

Example 1:

```
Host(config)# ip default-gateway 194.90.2.1 <cr>
      Operation Error:#
      The default gateway is out of subnet
```

USING RELOAD COMMAND

- The reload command restarts the RDS.
- The command is located at the “main” menu.

Usage

reload <non-graceful>

Parameter

<non-graceful>

It is for reset the Machine (RDS).

Example:

```
RDSHost>
RDSHost> reload non-graceful
```

WRITE COMMANDS

Using write terminal command

- The write terminal command displays the running RDS configuration.
- It is located at the “main” menu.

Usage

write terminal

Example 1:

```
RDSHost>
RDSHost> write terminal <cr>
# version: V1.0_RDS Apr 25 2002 15:44:56
password operator r^dfit
password technician ^-e3t
event-level 1 output-device sys-logger
default-redirection-site http://www.yahoo.com
interface Ethernet 1 192.168.1.4 255.255.255.0
ip default-gateway 192.168.1.1
sysLog-server-ip 192.168.1.3
RDSHost>
```

Example 2: Write terminal command performed before configuring the RDS

```
RDSHost> write terminal <cr>
# version: V1.0_RDS Apr 25 2002 15:44:56
# version: V1.0_RDS Apr 25 2002 15:44:56
password operator r^dfit
password technician ^-e3t
event-level 0 output-device none
default-redirection-site http://RDS.html
sysLog-server-ip 0.0.0.0
RDSHost>
```


Using write memory command

- The write memory command writes the running RDS configuration to the NV memory.
- It is located at the “main” menu.

Usage

write memory

Example:

```
RDSHost>  
RDSHost> write memory <cr>  
RDSHost>
```

USING POWEROFF COMMAND

- The poweroff command brings the system down in a secure way.
- It is located at the “main” menu.

Usage

poweroff

Example:

```
RDSHost>  
RDSHost> poweroff  
RDSHost> system is shutting down...
```

SNMP COMMANDS

Using access-list SNMP-permit command

- The access-list SNMP-permit command enables SNMP access command.
- It is located at the “configure terminal” menu.

Usage

access-list SNMP-permit <IP address> <source mask>

Parameter(s)

<IP address>

It is the Permitted source IP address.

<source mask>

It is the Permitted source mask.

Example:

```
RDSHost(config)> access-list SNMP-permit 192.168.1.0 255.255.255.0 <cr>
RDSHost(config)> end
RDSHost> write terminal <cr>
...
access-list SNMP-permit 192.168.1.0 255.255.255.0
...
```

Using no access-list SNMP-permit command

- The no access-list SNMP-permit command deletes SNMP access definitions.
- It is located at the “configure terminal” menu.

Usage

no access-list SNMP-permit <IP address>

Parameter

<IP address>

It is the permitted source IP address/Network.

Example:

```
RDSHost(config)> no access-list SNMP-permit 192.168.1.0 255.255.255.0 <cr>
RDSHost(config)> end
RDSHost>
```

Using SNMP-server community command

- The SNMP-server community command defines get and set SNMP community.
- It is located at the “configure terminal” menu.
- The default community is “public”.

Usage

SNMP-server community <get | set> <community string>

Parameter

<community string>

It is an Alpha community numeric string.

Example:

```
RDSHost(config)> SNMP-server community set test <cr>
RDSHost(config)> end
RDSHost> write terminal <cr>
...
SNMP-server community set test
...
```

USING RESET CONFIGURATION COMMAND

- The reset configuration command deletes the RDS stored configuration.
- It is located at the “ADC” menu.

Usage

reset configuration

HTTP COMMANDS

Using http-proxy-server port command

- The http-proxy-server port command defines the supported Http-proxy server posts in the system.
- The command is located in the “configure terminal” menu.
- The total available http-proxy-server ports are 3.

Usage

http-proxy-server port <port number>

Parameter

<port number>

It is the supported HTTP proxy port number. Legal port number (1 – 65,000).

Example:

```
RDSHost(config)> http-proxy-server port 8080 <cr>
RDSHost(config)> http-proxy-server port 8090 <cr>
RDSHost(config)> end
RDSHost> write terminal <cr>
...
http-proxy-server port 8080
http-proxy-server port 8090
...
```

Using no http-proxy-server port command

- The no http-proxy-server port command deletes configured Http-proxy server posts in the system.
- The command is located in the “configure terminal” menu.

Usage

no http-proxy-server port <port number>

Parameter

<port number>

It is the supported HTTP proxy port number. Legal port number (1 – 65,000).

Example:

```
RDSHost(config)> no http-proxy-server port 8080 <cr>
RDSHost(config)> no http-proxy-server port 8090 <cr>
RDSHost(config)> end
RDSHost> write terminal <cr>
...
```

NAME SERVER

Using ip primary-name-server command

- The ip primary-name-server command defines the primary DNS server IP address the RDS should use for the Http proxy functionality.
- The command is located in the “configure terminal” menu.

Usage

ip primary-name-server <IP address>

Parameter

<IP address>

It is the DNS IP address.

Example:

```
RDSHost(config)> ip dns server 192.168.1.4 <cr>
RDSHost> write terminal <cr>
...
ip dns server 192.168.1.4
...
```

Using no ip primary-name-server command

- The no ip primary-name-server command deletes the DNS server IP address definition (set the OBI to 0.0.0.0).
- The command is located in the “configure terminal” menu.

Usage

no ip primary-name-server

Example:

```
RDSHost(config)> no ip primary-name-server <cr>
RDSHost> write terminal <cr>
...
```

Using ip remote-proxy command

- The ip remote-proxy command defines the Http proxy and port to be used in the Http proxy functionality.
- The command is located in the “configure terminal” menu.

Usage

ip remote-proxy <IP address> <port number>

Parameter

<IP address>

It is the DNS IP address.

<port number>

It is the DNS port number.

Example:

```
RDSHost(config)> ip remote-proxy 192.168.1.4 8080 <cr>
RDSHost> write terminal <cr>
...
ip remote-proxy 192.168.1.4 8080
...
```

Using no ip remote-proxy command

- The no ip remote-proxy command deletes the Http proxy definition.
- The command is located in the “configure terminal” menu.

Usage

no ip remote-proxy

Example:

```
RDSHost(config)> no ip remote-proxy <cr>
RDSHost> write terminal <cr>
...
```

USING HOSTNAME COMMAND

- The hostname command defines the system hostname name.
- The command is located in the “configure terminal” menu.
- The default hostname is “RDShost”.

Usage

hostname <host name>

Parameter

<host name>

Example:

```
RDSHost(config)> hostname rds_test <cr>
RDSHost> write terminal <cr>
...
hostname rds_test
...
```

USING EDS-URL-IDENTITY COMMAND

- The eds-url-identity command defines the eds name to be used in the EDS URL.
- The command is located in the “configure terminal” menu.
- The default eds identity is the system primary IP address.

Usage

eds-url-identity <eds url name>

Example 1:

```
RDSHost(config)> eds-url-identity redirect_test
RDSHost> write terminal <cr>
...
eds-url-identity redirect_test
```

USING NO EDS-URL-IDENTITY COMMAND

- The no eds-url-identity command deletes the eds name configuration to be used in the EDS URL. The system uses its default value (system primary IP address).
- The command is located in the “configure terminal” menu.

Usage

no eds-url-identity

Example 1:

```
RDSHost(config)> no eds-url-identity redirect_test
RDSHost> write terminal <cr>
...
```

IP-IN-IP COMMANDS

Using remote-ip-in-ip command

- The remote-ip-in-ip command defines the remote endpoints in which the system should return the traffic through an ip-in-ip tunnel.
- The command is located in the “configure terminal” menu.
- The maximum number of remote ip-in-ip endpoints is 32.



When the remote-ip-in-ip is not defined, the system returns the traffic to the users without ip-in-ip encapsulation (as done today).

Usage

remote-ip-in-ip <IP address>

Parameter

<IP address>

It is the remote ip-in-ip endpoint IP address.

Example:

```
RDSHost(config)> remote-ip-in-ip 192.168.1.4 <cr>
RDSHost(config)> remote-ip-in-ip 192.168.1.5 <cr>
RDSHost(config)> remote-ip-in-ip 192.168.1.6 <cr>
RDSHost> write terminal <cr>
...
remote-ip-in-ip 192.168.1.4
remote-ip-in-ip 192.168.1.4
remote-ip-in-ip 192.168.1.4
...
```

Using no remote-ip-in-ip command

- The no remote-ip-in-ip command deletes the remote endpoint definition.
- The command is located in the “configure terminal” menu.

Usage

no remote-ip-in-ip <IP address>

Parameter

<IP address>

It is the remote ip-in-ip endpoint IP address.

Example:

```
RDSHost(config)> no remote-ip-in-ip 192.168.1.4 <cr>
RDSHost(config)> no remote-ip-in-ip 192.168.1.5 <cr>
RDSHost(config)> no remote-ip-in-ip 192.168.1.6 <cr>
RDSHost> write terminal <cr>
...
```

SHOW USER COMMANDS

Using show users command

- The command displays connected users information.
- The command is located at ADC menu.

Usage

show users

Output format:

Default of Linux w command.

Using show proc command

- The command display running processes information.
- The command is located at ADC menu.

Usage

show proc

Output format:

Default of Linux ps ax command.

Using show memory command

- The command display memory information.
- The command is located at ADC menu.

Usage

show memory

Output format:

Default of Linux cat /proc/meminfo command.

Using show cpu command

- The command display cpu information.
- The command is located at ADC menu.

Usage**show cpu****Output format:**

Default of Linux cat /proc/cpuinfo command.

USING DEBUG PROTOCOL COMMAND

- The command enables the sniffing of tcp, udp, ether, fddi, ip, arp, rarp, decent, lat, sca, moprc, mopdl, icmp, igrp, nd.
- The command is located at ADC menu.

Usage**debug protocol** [[<protocol name>] [source | dest <IP>]]**Parameter(s)**

<protocol name>

It is the Sniffed protocol, and can be one of the following protocols: tcp, udp, ether, fddi, ip, arp, rarp, decent, lat, sca, moprc, mopdl, icmp, igrp, nd.

<IP>

It is the IP address.

Output format:

Default of Linux tcpdump command.

USING REST WEB COMMAND

- The command resets RDS http demon.
- The command is located at ADC menu.

Usage**reset web**

USING DATE COMMAND

- The command sets RDS date.
- The command is located at ADC menu.

Usage

date <time> <date>

Parameter(s)

<time>

It is Time of the day in H24:MM:SS format.

<date>

It is the Date in DD/MM/YYYY format.

Example:

```
RDSHost(ADC)> date 18:07:30 10/02/2002 <cr>
RDSHost(ADC)>
```

PRODUCT SUPPORT

ADC Customer Service Group provides expert pre-sales support and training for all of its products. Technical support is available 24 hours a day, 7 days a week by contacting the ADC Technical Assistance Center.

| | |
|--------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Sales Assistance: 800.366.3891 | Quotation Proposals, Ordering and Delivery General, and Product Information |
| Systems Integration: 800.366.3891 | Complete Solutions (from concept to installation), Network Design and Integration Testing, System Turn-Up and Testing, Network Monitoring (upstream or downstream), Power Monitoring and Remote Surveillance, Service/Maintenance Agreements, and Systems Operation |
| ADC Technical Assistance Center: 800.366.3891 Email: wsd.support@adc.com | Technical Information, System/Network Configuration, Product Specification and Application, Training (product-specific), Installation and Operation Assistance, and Troubleshooting and Repair/Field Assistance |
| Online Technical Support: | www.adc.com/Knowledge_Base/index.jsp |
| Online Technical Publications: | www.adc.com/documentationlibrary/technicalpublications/ |
| Product Return Department: 800.366.3891 Email: repair.return@adc.com | ADC Return Material Authorization (RMA) number and instructions must be obtained before returning products. |

GLOSSARY

A

ACL – Access Control List

ATM – Asynchronous Transfer Mode

APS – Automatic Protection System

C

CBR – Constant Bit Rate

CLEI – Common Language Equipment Identifier

CLI – Command Line Interface

D

DS3 – Digital Signal, Level 3

E

EMS – Element Management System

F

FRF.5 – Frame Relay/ATM Network Interworking Implementation

FRF.8 – Frame Relay-to-ATM Service Interworking

FTP – File Transfer Protocol

I

IDSL – ISDN Digital Subscriber Line

IP – Internet Protocol

L

LCP – Link Control Protocol

LAN – Local Area Network

M

MIB – Management Information Base

O

OC3 – Optical Carrier Level 3 (155.52 Mbps)

P

PNNI – Private Network to Node Interface

PVC – Permanent Virtual Circuit

PPP – Point-to-Point Protocol

S

SCC – Service Creation Cards

SG-1 – Service Gateway System (ADC product)

SNMP – Simple Network Management Protocol

SPVC – Semi-Permanent Virtual Circuit

T

TFTP – Trivial File Transport Protocol

V

VRRP – Virtual Router Redundancy Protocol

Certification and Warranty

FCC Class A Compliance

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

Limited Warranty

Product warranty is determined by your service agreement. Refer to the *ADC Warranty/Software Handbook* for additional information, or contact your sales representative or Customer Service for details.

Modifications

The FCC requires the user to be notified that any changes or modifications made to this device that are not expressly approved by ADC voids the user's warranty.

All wiring external to the products should follow the provisions of the current edition of the National Electrical Code.

Safety Standards Compliance

This equipment has been tested and verified to comply with the applicable sections of the following safety standards:

- GR 63-CORE - Network Equipment-Building System (NEBS) Requirements
- GR 1089-CORE - Electromagnetic Compatibility and Electrical Safety
- Binational Standard, UL-60950 3rd Edition/CSA1459 C22.2 No. 60950-00: Safety of Information Technology Equipment

For technical assistance, refer to **"Appendix C: Product Support"** on page C-1.

World Headquarters

ADC Telecommunications, Inc.
PO Box 1101
Minneapolis, MN 55440-1101 USA

For Technical Assistance

Tel: 800.366.3891

SG-1 Service Gateway System User Manual

Document Number: SG1-UM-8500-03



1374864

