

SunATM™ 4.0 Installation and User's Guide



THE NETWORK IS THE COMPUTER™

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900 USA
650 960-1300
Fax 650 969-9131

Part No.: 805-6552-10
March 1999, Revision A

Copyright 1999 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303-4900 U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Portions of this product may be derived from the UNIX® system, licensed from Novell, Inc., and from the Berkeley 4.3 BSD system, licensed from the University of California. UNIX is a registered trademark in the United States and in other countries and is exclusively licensed by X/Open Company Ltd. Third-party software, including font technology in this product, is protected by copyright and licensed from Sun's suppliers. RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g)(2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015(b)(6/95) and DFAR 227.7202-3(a).

Sun, Sun Microsystems, the Sun logo, SunATM, Ultra, Ultra Enterprise, SunNet Manager, and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and in other countries. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the United States and in other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK® and Sun™ Graphical User Interfaces were developed by Sun Microsystems, Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox Corporation in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a nonexclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

THIS PUBLICATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

Copyright 1999 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, Californie 94303-4900 U.S.A. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie et la décompilation. Aucune partie de ce produit ou de sa documentation associée ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Des parties de ce produit pourront être dérivées du système UNIX® licencié par Novell, Inc. et du système Berkeley 4.3 BSD licencié par l'Université de Californie. UNIX est une marque enregistrée aux Etats-Unis et dans d'autres pays, et licenciée exclusivement par X/Open Company Ltd. Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun.

Sun, Sun Microsystems, le logo Sun, SunATM, Ultra, Ultra Enterprise, SunNet Manager, et Solaris sont des marques déposées ou enregistrées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays. Toutes les marques SPARC, utilisées sous licence, sont des marques déposées ou enregistrées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays. Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

Les utilisateurs d'interfaces graphiques OPEN LOOK® et Sun™ ont été développés de Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox Corporation pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique, cette licence couvrant aussi les licenciés de Sun qui mettent en place les utilisateurs d'interfaces graphiques OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

CETTE PUBLICATION EST FOURNIE "EN L'ETAT" SANS GARANTIE D'AUCUNE SORTE, NI EXPRESSE NI IMPLICITE, Y COMPRIS, ET SANS QUE CETTE LISTE NE SOIT LIMITATIVE, DES GARANTIES CONCERNANT LA VALEUR MARCHANDE, L'APTITUDE DES PRODUITS A REPENDRE A UNE UTILISATION PARTICULIERE OU LE FAIT QU'ILS NE SOIENT PAS CONTREFAISANTS DE PRODUITS DE TIERS.



Contents

Preface Preface-xv

How This Book Is Organized Preface-xv

Using UNIX Commands Preface-xvi

Typographic Conventions Preface-xvii

Shell Prompts Preface-xvii

Ordering Sun Documents Preface-xviii

Sun Welcomes Your Comments Preface-xviii

1. Introduction to the SunATM Adapters 1-1

1.1 SunATM SBus Adapters 1-2

1.1.1 SunATM/S 155 Adapters 1-2

1.1.1.1 Features 1-3

1.1.1.2 Hardware Requirements 1-4

1.1.1.3 Software Requirements 1-4

1.1.2 SunATM/S 622 MMF Adapter 1-5

1.1.2.1 Features 1-5

1.1.2.2 Hardware Requirements 1-6

1.1.2.3 Software Requirements 1-6

1.2 SunATM PCI Adapters 1-7

1.2.1 SunATM/P 155 Adapters 1-7

1.2.1.1 Features 1-8

- 1.2.1.2 Hardware Requirements 1-8
- 1.2.1.3 Software Requirements 1-9
- 1.2.2 SunATM/P 622 MMF Adapter 1-9
 - 1.2.2.1 Features 1-10
 - 1.2.2.2 Hardware Requirements 1-10
 - 1.2.2.3 Software Requirements 1-11

2. Installing the SunATM SBus Adapters 2-1

- 2.1 Installing the SunATM/S 155 Adapters 2-1
 - ▼ SunATM/S 155 Adapter Installation 2-2
 - 2.1.1 Verifying the SunATM/S 155 Adapter Installation 2-3
 - 2.1.2 SunATM/S 155 MMF Adapter Wiring Configuration 2-4
 - ▼ To connect the SunATM/S 155 MMF adapter to the network 2-4
 - 2.1.3 SunATM/S 155 UTP5 Adapter Wiring Configuration 2-4
 - ▼ To connect the SunATM/S 155 UTP5 adapter to the network 2-5
- 2.2 Installing the SunATM/S 622 MMF Adapter 2-5
 - ▼ SunATM/S 622 MMF Adapter Installation 2-5
 - 2.2.1 Verifying the SunATM/S 622 Adapter Installation 2-8
 - 2.2.2 SunATM/S 622 MMF Adapter Wiring Configuration 2-9
 - ▼ To connect the SunATM/S 622 MMF Adapter to the network 2-9

3. Installing the SunATM PCI Adapters 3-1

- 3.1 SunATM/P Adapter Installation 3-1
- 3.2 Verifying the Installation 3-2
- 3.3 SunATM/P Adapter Wiring Configuration 3-3
 - 3.3.1 SunATM/P 155 MMF Adapter Wiring Configuration 3-3
 - ▼ To connect the SunATM/P 155 MMF adapter to the network 3-4
 - 3.3.2 SunATM/P 155 UTP5 Adapter Wiring Configuration 3-4
 - ▼ To connect the SunATM/P 155 UTP5 adapter to the network 3-5
 - 3.3.3 SunATM/P 622 MMF Adapter Wiring Configuration 3-5
 - ▼ To connect the SunATM/P 622 MMF adapter to the network 3-5

- 4. Installing the SunATM Software 4-1**
 - 4.1 Before Installing the SunATM Software 4-1
 - 4.1.1 Removing Links to Older SunATM 1.0 Device Entries 4-2
 - 4.1.2 Required Patches 4-2
 - 4.2 Installing the SunATM Software 4-3
 - 4.2.1 Adding the Software Packages Using `pkgadd` 4-3
 - 4.2.2 Checking the Package Installation Using `pkgchk` 4-5
 - 4.2.3 Checking the Package Installation Using `pkginfo` 4-5
 - 4.2.4 Removing the Software Packages Using `pkgrm` 4-6
 - 4.3 Rebooting the System and Examining Network Interfaces 4-6

- 5. Configuring the SunATM Interface 5-1**
 - 5.1 New Features in the SunATM Software 5-1
 - 5.1.1 Support for the Solaris 7 Operating Environment 5-1
 - 5.1.2 Support for Dynamic Reconfiguration 5-2
 - 5.1.3 Standardized Packaging Format 5-2
 - 5.1.4 Unified Driver support for PCI and SBus 5-2
 - 5.2 Using the `atmadmin` Configuration Program 5-3
 - 5.2.1 Starting the `atmadmin` Configuration Program 5-3
 - 5.2.2 `atmadmin` Main Menu 5-4
 - 5.2.3 `atmadmin` Navigation Commands 5-4
 - 5.2.4 System Parameter Group Menu 5-5
 - 5.2.4.1 ATM SNMP Agent Status 5-5
 - 5.2.5 Interface Configuration Menu 5-6
 - 5.2.5.1 `atmadmin` and the SunATM Configuration Files in the `/etc/opt/SUNWconn/atm` directory 5-6
 - 5.3 `atmadmin` Parameter Groups 5-7
 - 5.3.1 Physical Layer Parameter Group 5-9
 - 5.3.1.1 Framing Interface 5-9
 - 5.3.2 Signalling Parameter Group 5-10
 - 5.3.2.1 UNI Version 5-10

- 5.3.3 ILMI Parameter Group 5-11
- 5.3.4 Classical IP Parameter Group 5-11
 - 5.3.4.1 Classical IP Interface Type 5-12
 - 5.3.4.2 Hostname and IP Address 5-12
 - 5.3.4.3 Local ATM Address 5-13
 - 5.3.4.4 ATM ARP Server Address 5-14
 - 5.3.4.5 Permanent Virtual Circuit (PVC) 5-14
- 5.3.5 LAN Emulation Parameter Group 5-15
 - 5.3.5.1 Per-Instance LAN Emulation 5-16

6. Editing SunATM Configuration Files 6-1

- 6.1 Editing the `/etc/opt/SUNWconn/atm/atmconfig` File 6-1
 - 6.1.1 Changing the Framing Interface in the `/etc/opt/SUNWconn/atm/atmconfig` File 6-2
 - 6.1.2 Example of an `/etc/opt/SUNWconn/atm/atmconfig` File 6-3
- 6.2 Configuring a Classical Internet Protocol Interface 6-3
 - 6.2.1 Editing the `/etc/opt/SUNWconn/atm/aarconfig` File 6-4
 - 6.2.2 Using Variables in the `/etc/opt/SUNWconn/atm/aarconfig` File 6-7
 - 6.2.3 Sample Classical IP Configurations 6-10
- 6.3 Configuring a LAN Emulation Interface 6-12
 - 6.3.1 Editing the `/etc/opt/SUNWconn/atm/laneconfig` File 6-13
 - 6.3.2 Using Variables in the `/etc/opt/SUNWconn/atm/laneconfig` File 6-16
 - 6.3.3 Sample LAN Emulation Configurations 6-17
- 6.4 Supporting Logical Interfaces 6-17
- 6.5 Supporting Multiple Emulated LANs on a Single Interface 6-20
- 6.6 Plumbing and Unplumbing Individual ATM Interfaces 6-22

7. Classical IP and LAN Emulation Protocols 7-1

- 7.1 ATM Addresses and Address Registration 7-2
 - 7.1.1 ATM Address Registration Daemon (`ilmid`) 7-2

7.2	Classical Internet Protocol	7-3
7.2.1	ATM Address Resolution	7-3
7.2.2	ATM ARP Address Resolution Tables	7-4
7.3	LAN Emulation	7-5
7.3.1	LAN Emulation Servers	7-5
7.3.1.1	LAN Emulation Configuration Server	7-6
7.3.1.2	LAN Emulation Server	7-6
7.3.1.3	Broadcast and Unknown Address Server	7-6
7.3.2	Resolving an IP Address to an ATM Connection	7-6
7.3.3	LAN Emulation Connections	7-7
7.4	ATM and SNMP	7-8
7.4.1	SNMP and Solaris	7-9
7.4.1.1	Releases of Solaris Prior to 2.6	7-10
7.4.1.2	Solaris 2.6 and Solaris 7 and Compatible Software	7-10

A. Declarations of Conformity A-1

B. Wiring Scheme and Pin Descriptions B-1

C. SunATM Adapters Specifications C-1

C.1	SunATM/S Adapter Specifications	C-1
C.1.1	SunATM/S 155 Adapter Specifications	C-1
C.1.1.1	Physical Dimensions	C-1
C.1.1.2	Performance Specification	C-2
C.1.1.3	Power Specifications	C-2
C.1.1.4	Environmental Specifications	C-2
C.1.2	SunATM/S 622 MMF Adapter Specifications	C-4
C.1.2.1	Physical Dimensions	C-4
C.1.2.2	Performance Specifications	C-4
C.1.2.3	Power Specifications	C-4
C.1.2.4	Environmental Specifications	C-5
C.2	SunATM/P Adapter Specifications	C-6

- C.2.1 SunATM/P 155 Adapter Specifications C-6
 - C.2.1.1 Physical Dimensions C-6
 - C.2.1.2 Performance Specifications C-6
 - C.2.1.3 Power Specifications C-6
 - C.2.1.4 Environmental Specifications C-8
- C.2.2 SunATM/P 622 MMF Adapter Specifications C-8
 - C.2.2.1 Physical Dimensions C-8
 - C.2.2.2 Performance Specifications C-8
 - C.2.2.3 Power Specifications C-9
 - C.2.2.4 Environmental Specifications C-9

D. Troubleshooting and Error Messages D-1

- D.1 Troubleshooting While Starting a SunATM Interface D-1
 - D.1.1 Generic Configuration D-1
 - ▼ To Diagnose Problems D-1
 - D.1.2 Classical IP Configuration D-3
 - D.1.3 LAN Emulation Configuration D-5
 - D.1.4 Common Problems D-8
 - D.1.4.1 Are you replacing an old SunATM SBus adapter? D-8
 - D.1.4.2 Are you trying to use the `/usr/sbin/arp` command? D-9
 - D.1.4.3 Are you using a Router with Classical IP and LAN Emulation (LANE)? D-9
 - D.1.4.4 Are you trying to use the `/usr/sbin/snoop` command? D-10
 - D.1.4.5 Do you want to increase system performance by adjusting TCP/IP parameters? D-10
 - D.1.4.6 Do you want to improve the performance of NFS version 3? D-10
 - D.1.4.7 Do you want to decrease the NFS Read and Write Buffer Sizes in a Solaris 2.5.1 or earlier operating environment? D-11
 - D.1.4.8 Are you trying to mount a diskless, dataless, or autocolient system? D-12

D.1.4.9	Did the atmtest diagnostic fail?	D-12
D.2	Error Messages	D-12
D.2.1	Error Messages from s00sunatm	D-13
D.2.2	Error Messages from aarsetup and lanesetup	D-15
D.2.3	Error Messages from the Kernel Drivers	D-18
E.	Managing SunATM Interfaces with SNMP	E-1
E.1	Installing the SunATM SNMP Software	E-1
E.2	Setting Up the Management Console	E-1
E.3	Setting Up Agent Systems	E-2
F.	Application Programmers' Interface	F-1
F.1	Using the SunATM API with the Q.93B and the ATM Device Drivers	F-2
F.1.1	Q.93b Driver Interface	F-2
F.1.2	Establishing a Connection to the Q.93B Driver	F-3
F.1.3	Setting up an ATM Connection Over a Switched Virtual Circuit (SVC)	F-3
F.1.3.1	Call Setup	F-6
F.1.3.2	Release Procedure	F-7
F.1.3.3	Exception Conditions	F-7
F.2	Connecting, Sending, and Receiving Data with the ATM Device Driver	F-9
F.2.1	Raw Mode Connections	F-10
F.2.2	DLPI Mode Connections	F-10
G.	SunVTS Validation and Test Suite	G-1
G.1	ATM Adapter Test (atmtest)	G-1
G.1.1	atmtest Test Modes	G-2
G.1.2	atmtest Command Line Syntax	G-2
G.1.3	atmtest Options	G-3

Figures

FIGURE 1-1	SunATM/S 155 MMF Adapter	1-2
FIGURE 1-2	SunATM/S 155 UTP5 Adapter	1-3
FIGURE 1-3	SunATM/S 622 Adapter	1-5
FIGURE 1-4	SunATM/P 155 MMF Adapter	1-7
FIGURE 1-5	SunATM/P 155 UTP5 Adapter	1-8
FIGURE 1-6	SunATM/P 622 MMF Adapter	1-10
FIGURE 2-1	SunATM/S 155 MMF Adapter Backplate	2-4
FIGURE 2-2	SunATM/S 155 UTP Adapter Backplate	2-5
FIGURE 2-3	Installing the SunATM/S 622 MMF Adapter	2-6
FIGURE 2-4	SunATM/S 622 MMF Adapter's Backplate	2-7
FIGURE 2-5	Sun Ultra 1 Series System Back Panel	2-7
FIGURE 2-6	SunATM/S 622 MMF Adapter	2-8
FIGURE 3-1	SunATM/P 155 MMF Adapter Backplate	3-4
FIGURE 3-2	SunATM/P 155 UTP5 Adapter Backplate	3-4
FIGURE 3-3	SunATM/P 622 MMF Adapter Backplate	3-5
FIGURE 7-1	ATM Address Fields	7-2
FIGURE 7-2	Using <code>atmsnmpd</code> as a Forwarding Agent	7-9
FIGURE B-1	Designation T568B	B-1
FIGURE F-1	ATM Signalling	F-2

FIGURE F-2 Message Format F-5

FIGURE F-3 Message Flow for Normal Call Setup and Tear Down F-8

Tables

TABLE 1-1	SunATM/S 155 Architecture and Supporting Systems	1-4
TABLE 1-2	SunATM/S 622 Architecture and Supporting Systems	1-6
TABLE 1-3	SunATM/P 155 Architecture and Supporting Systems	1-9
TABLE 1-4	SunATM/P 622 Architecture and Supporting Systems	1-11
TABLE 4-1	Required Patches	4-2
TABLE 5-1	Basic Navigational Commands in <code>atmadmin</code>	5-4
TABLE 5-2	Configurable Parameters in the SunATM Software	5-7
TABLE 5-3	Predefined SunATM Variables	5-13
TABLE 6-1	<code>/etc/opt/SUNWconn/atm/atmconfig</code> Field Descriptions	6-2
TABLE 6-2	<code>/etc/opt/SUNWconn/atm/aarconfig</code> File Flag Descriptions	6-5
TABLE 6-3	<code>/etc/opt/SUNWconn/atm/aarconfig</code> File Flag Requirements and Options	6-6
TABLE 6-4	Predefined SunATM Variables	6-8
TABLE 6-5	<code>/etc/opt/SUNWconn/atm/laneconfig</code> Entry Descriptions	6-13
TABLE 6-6	<code>/etc/opt/SUNWconn/atm/laneconfig</code> Flag Descriptions	6-14
TABLE 6-7	<code>laneconfig</code> Flag Requirements and Options	6-15
TABLE 6-8	Predefined SunATM Variables	6-16
TABLE 6-9	Parameter Options for <code>atmifconfig</code>	6-22
TABLE 7-1	LAN Emulation Connections	7-8
TABLE C-1	Physical Dimensions of SunATM/S 155 Adapters	C-1

TABLE C-2	Performance Specifications of SunATM/S 155 Adapters	C-2
TABLE C-3	Power Specifications of SunATM/S 155 Adapters	C-2
TABLE C-4	Environmental Specifications of SunATM/S 155 Adapters	C-2
TABLE C-5	Physical Dimensions of SunATM/S 622 Adapters	C-4
TABLE C-6	Performance Specifications of SunATM/S 622 Adapters	C-4
TABLE C-7	Power Specifications of SunATM/S 622 Adapters	C-4
TABLE C-8	Environmental Specifications of SunATM/S 622 Adapters	C-5
TABLE C-9	Performance Specifications of SunATM/P 155 Adapters	C-6
TABLE C-10	Power Specifications of SunATM/P 155 Adapters	C-6
TABLE C-11	Environmental Specifications of SunATM/P 155 Adapters	C-8
TABLE C-12	Physical Dimensions of SunATM/P 622 Adapters	C-8
TABLE C-13	Performance Specifications of SunATM/P 622 Adapters	C-8
TABLE C-14	Power Specifications of SunATM/P 622 Adapters	C-9
TABLE C-15	Environmental Specifications of SunATM/P 622 Adapters	C-9
TABLE F-1	Messages Between the User and the Q.93B Driver	F-3
TABLE F-2	Fields in the M_PROTO mblock	F-5
TABLE F-3	qcc Functions	F-6
TABLE F-4	atm_util Function Overview	F-9
TABLE G-1	atmtest Command Line Syntax	G-2
TABLE G-2	atmtest Options	G-3

Preface

The *SunATM™ 4.0 Installation and User's Guide* provides installation instructions for the SunATM SBus and PCI adapters supported by the SunATM 4.0 software. This manual also describes how to install and configure the SunATM software.

These instructions are designed for a system administrator with experience installing similar hardware and software.

How This Book Is Organized

Chapter 1 introduces the features of the six SunATM adapters.

Chapter 2 describes how to install the SunATM/S adapters.

Chapter 3 describes how to install the SunATM/P adapters.

Chapter 4 describes how to install the SunATM 4.0 software on your system.

Chapter 5 describes some of the features in the SunATM software and how to configure the software using configuration programs.

Chapter 6 describes how to edit the SunATM configuration files to best suit your network.

Chapter 7 provides background information about the Classical Internet Protocol (IP) and the local area network (LAN) emulation protocol, which allow IP traffic to run over Asynchronous Transfer Mode (ATM) interfaces.

Appendix A contains the Declaration of Conformity certificates for each SunATM adapter.

Appendix B contains the wiring schemes and pin descriptions for the SunATM adapters.

Appendix C lists the specifications for each SunATM adapter.

Appendix D provides troubleshooting and descriptions for error messages. There is also a list of common questions.

Appendix E describes how to install and set up Simple Network Management Protocol (SNMP) agents.

Appendix F defines the SunATM application programmer's interface (API).

Appendix G provides pointers to the SunVTS™ test suite, and it describes how to run the OpenBoot™ selftest.

Using UNIX Commands

This document may not contain information on basic UNIX commands and procedures such as shutting down the system, booting the system, and configuring devices.

See one or more of the following for this information:

- *Solaris Handbook for Sun Peripherals*
- AnswerBook online documentation for the Solaris software environment
- Other software documentation that you received with your system

Typographic Conventions

TABLE P-1 Typographic Conventions

Typeface or Symbol	Meaning	Examples
AaBbCc123	Names of commands, files, and directories; on-screen computer output.	Edit your <code>.login</code> file. Use <code>ls -a</code> to list all files. % You have mail.
AaBbCc123	What you type, when contrasted with on-screen computer output.	% su Password:
<i>AaBbCc123</i>	Book titles, new words or terms, words to be emphasized	Read Chapter 6 in the <i>User's Guide</i> . These are called <i>class</i> options. You <i>must</i> be superuser to do this.
	Command-line variable; replace with a real name or value	To delete a file, type <code>rm filename</code>

Shell Prompts

TABLE P-2 Shell Prompts

Shell	Prompt
C shell	<i>machine_name%</i>
C shell superuser	<i>machine_name#</i>
Bourne shell and Korn shell	\$
Bourne shell and Korn shell superuser	#

Ordering Sun Documents

The `docs.sun.com` web site enables you to access Sun technical documentation on the web. You can browse the `docs.sun.com` archive or search for a specific book title or subject at:

`http://docs.sun.com`

Sun Welcomes Your Comments

We are interested in improving our documentation and welcome your comments and suggestions. You can email your comments to us at:

`docfeedback@sun.com`

Please include the part number of your document in the subject line of your email.

Introduction to the SunATM Adapters

This chapter introduces the six adapters supported by the SunATM 4.0 software, including:

- SunATM/S 155 MMF adapter
- SunATM/S 155 UTP5 adapter
- SunATM/S 622 MMF adapter
- SunATM/P 155 MMF adapter
- SunATM/P 155 UTP5 adapter
- SunATM/P 622 MMF adapter

The features, hardware requirements, and software requirements of these adapters are described in the following sections.

Note – All references to the SunATM/S adapter refer to SunATM/S 155 version 2.0 or SunATM/S 622 version 2.1, unless otherwise noted. All references to the SunATM/P adapter refer to SunATM/P 155 or SunATM/P 622 version 3.0, unless otherwise noted.

Some of the features, common to all SunATM adapters:

- Supports IETF RFC 1577 for Classical IP over ATM
- Supports IETF RFC 1483 for Protocol Encapsulation over ATM (except for bridge encapsulation)
- Supports ATM Forum recommendation for UNI 3.0, 3.1, or 4.0
- Supports ATM Forum recommendation for LAN Emulation 1.0
- Is compatible with relevant emerging standards (including existing ATM Forum baseline specifications and ITU-TS)
- Supports Dynamic Reconfiguration (DR)
- Redesigned packaging to comply with standard packaging efforts

1.1 SunATM SBus Adapters

1.1.1 SunATM/S 155 Adapters

The SunATM/S 155 MMF and SunATM/S 155 UTP5 adapters are single-wide SBus adapters that conform to the specifications of the Asynchronous Transfer Mode (ATM) Forum. The adapters offer 155 Mbps network bandwidth over either multimode fiber-optic cable or Category 5 unshielded twisted-pair (UTP) copper wire.

The SunATM/S 155 adapters are designed for systems that run under Solaris 2.5.1, 2.6, or 7. To use the SunATM/S 155 adapters, the system must also contain OpenBoot PROM (OBP), version 2.0 or compatible. An on-board FCode PROM provides the configuration support that identifies the SunATM/S 155 adapters to the system.

Note – For instructions on how to install the adapters, see Chapter 2 “Installing the SunATM SBus Adapters.”

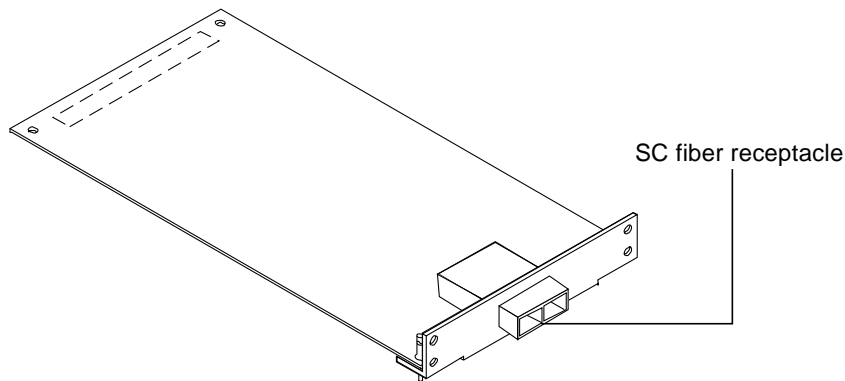


FIGURE 1-1 SunATM/S 155 MMF Adapter

Note – The SunATM/S 155 1.0 adapters are not supported by the SunATM 4.0 software.

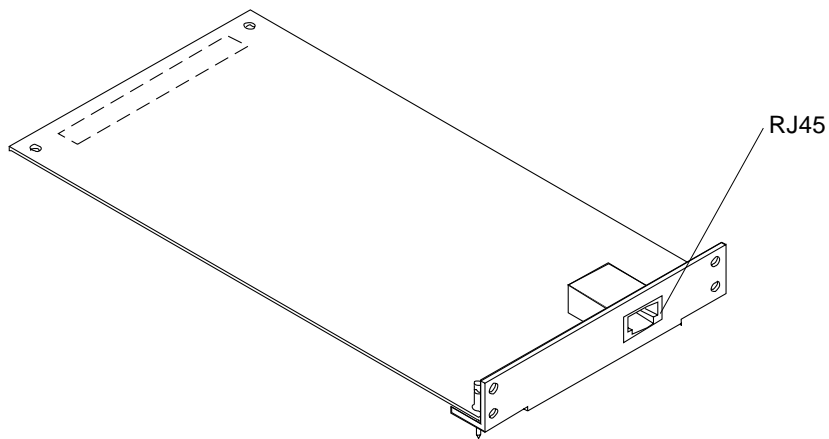


FIGURE 1-2 SunATM/S 155 UTP5 Adapter

1.1.1.1 Features

The SunATM/S 155 adapters:

- Conform to IEEE 1496
- Support 155-Mbps operation over:
 - 62.5/125 μ multimode fiber (SunATM/S 155 MMF Adapter)
 - UTP Category 5 wire (SunATM/S 155 UTP5 Adapter)
- Integrate SBus and SAR (segmentation and reassembly) functions in the SAHI (SBus to ATM host interface) ASIC
- Align SAR functionality with ATM Forum specified and International Telecommunications Union-Telecommunication Sector (ITU-TS) approved ATM Adaptation Layer (AAL) 5
- Support the SONET and SDH (Synchronous Optical NETWORK/Synchronous Digital Hierarchy) physical layer framing structures
- Support up to 1024 simultaneous virtual channels—either SVC or PVC

Note – Version 2.0 OpenBoot PROMs (or compatible) are *required* for systems using the SunATM/S 155 adapters. If lower-version boot PROMs are installed on your system, you must upgrade the boot PROMs before using SunATM/S 155 adapters. To find the OpenBoot PROM (OBP) revision version on your system, type `.version` at the `ok` prompt.

1.1.1.2 Hardware Requirements

To connect the SunATM/S 155 adapters to an ATM switch, you need the following cables:

- SunATM/S 155 MMF adapter: multimode fiber cable with an SC connector
- SunATM/S 155 UTP5 adapter: category 5 UTP with an RJ-45 connector

Refer to the manual supplied with the ATM switch for specific instructions about cable connections.

TABLE 1-1 shows the SBus based Sun-4u, Sun-4m, and Sun-4d architecture systems that support the SunATM/S 155 adapters.

TABLE 1-1 SunATM/S 155 Architecture and Supporting Systems

Platform Architecture	System Type
Sun-4u	Sun Ultra™ 1 Creator Series Sun Ultra 1 Series Sun Ultra 2 Series Ultra Enterprise™ 1 Ultra Enterprise 2 Ultra Enterprise 3000/3500 Ultra Enterprise 4000/4500 Ultra Enterprise 5000/5500 Ultra Enterprise 6000/6500 Ultra Enterprise 10000
Sun-4m	SPARCstation™ Classic SPARCstation LX SPARCstation 4 SPARCstation 5 SPARCstation 10 SPARCstation 10SX SPARCstation 20 Series SPARCstation 600 Series
Sun-4d	SPARCserver™ 1000 and 1000E SPARCcenter™ 2000 and 2000E

1.1.1.3 Software Requirements

The SunATM/S 155 adapters are supported on systems running under Solaris 2.5.1, 2.6, or 7.

The SunATM 4.0 CD-ROM that shipped with the SBus adapter contains the required driver software that must be installed to connect a SunATM/S 155 adapter to a network.

1.1.2 SunATM/S 622 MMF Adapter

The SunATM/S 622 MMF Adapter is a double-wide adapter that conforms to the specifications of the Asynchronous Transfer Mode (ATM) Forum. The adapter offers 622 Mbps network bandwidth over a multimode fiber-optic cable.

The SunATM/S 622 MMF Adapter is designed for operation in systems that run Solaris 2.5.1, 2.6, or 7. To use the SunATM/S 622 adapter, the system must also contain an OpenBoot™ PROM (OBP), version 2.0 or compatible. An on-board FCode PROM provides the configuration support that identifies the adapter to the system.

Note – For instructions on how to install the adapter, see Chapter 2 “Installing the SunATM SBus Adapters”.

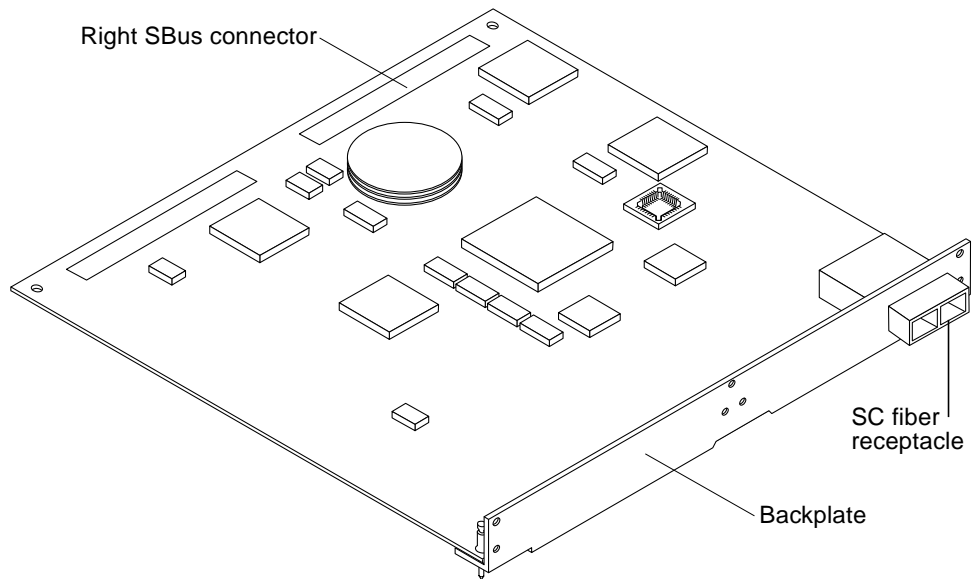


FIGURE 1-3 SunATM/S 622 Adapter

1.1.2.1 Features

The SunATM/S 622 MMF adapter:

- Conforms to IEEE 1496
- Supports 622 Mbps operation over 62.5/125 μ multimode fiber

- Integrates SBus and SAR (segmentation and reassembly) in the SAHI (SBus to ATM Host Interface) ASIC
- Aligns SAR functionality with ATM Forum specified and International Telecommunications Union-Telecommunication Sector (ITU-TS) approved ATM Adaptation Layer (AAL) 5
- Supports the SONET and SDH (Synchronous Optical Network/Synchronous Digital Hierarchy) physical layer framing structure
- Supports up to 1024 simultaneous virtual channels—either SVC or PVC

1.1.2.2 Hardware Requirements

To connect the SunATM/S 622 adapter to an ATM switch, you will need a multimode fiber cable with an SC connector.

Refer to the manual supplied with the ATM switch for the specific instructions about cable connections.

TABLE 1-2 shows the Sun-4u, Sun-4m, and Sun-4d architecture systems that support the SunATM/S 622 adapter.

TABLE 1-2 SunATM/S 622 Architecture and Supporting Systems

Platform Architecture	System Type
Sun-4u	Sun Ultra 1 Creator Series Sun Ultra 1 Series Sun Ultra 2 Series Ultra Enterprise 1 Ultra Enterprise 2 Ultra Enterprise 3000/3500 Ultra Enterprise 4000/4500 Ultra Enterprise 5000/5500 Ultra Enterprise 6000/6500 Ultra Enterprise 10000
Sun-4m	SPARCstation 20 Series
Sun-4d	SPARCserver 1000 and 1000E SPARCcenter 2000 and 2000E

1.1.2.3 Software Requirements

The SunATM/S 622 adapter and the SunATM 4.0 software are supported on systems running Solaris 2.5.1, 2.6, or 7.

The SunATM 4.0 CD-ROM that shipped with the adapter contains the software that must be installed to connect a SunATM/S 622 adapter to a network.

1.2 SunATM PCI Adapters

1.2.1 SunATM/P 155 Adapters

The SunATM/P 155 MMF adapter and the SunATM/P 155 UTP5 adapter are seven-inch PCI adapters that conform to the specifications of the Asynchronous Transfer Mode (ATM) Forum. The adapters offer 155 Mbps network bandwidth over either multimode fiber-optic cable or Category 5 unshielded twisted-pair (UTP) copper wire.

The SunATM/P 155 adapters are designed for operation in systems that run under Solaris 2.5.1 Hardware: 4/97, 8/97, or 11/97, Solaris 2.6, or Solaris 7. An on-board FCode PROM provides the configuration support that identifies the SunATM/P 155 adapters to the system.

Note – For instructions on how to install the adapters, see Chapter 3 “Installing the SunATM PCI Adapters.”

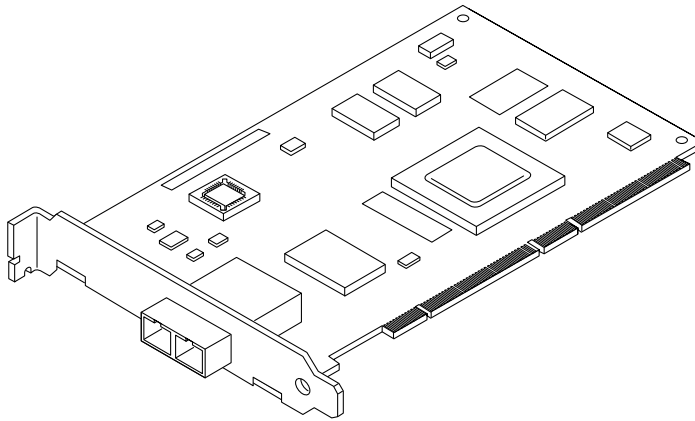


FIGURE 1-4 SunATM/P 155 MMF Adapter

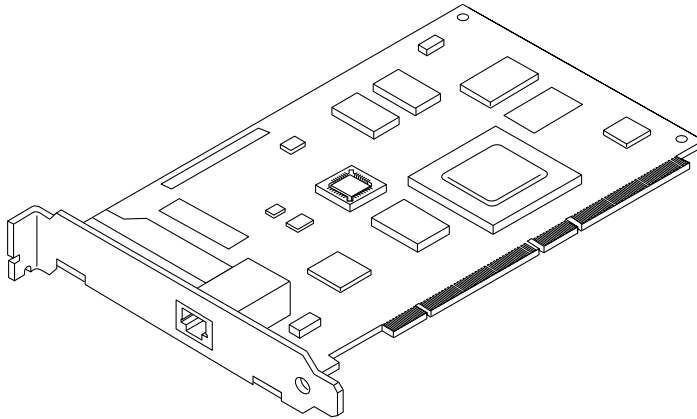


FIGURE 1-5 SunATM/P 155 UTP5 Adapter

1.2.1.1 Features

The SunATM/P 155 adapters:

- Support 155-Mbps operation over:
 - 62.5/125 μ multimode fiber (SunATM/P 155 MMF) or
 - UTP Category 5 wire conforming with the TIA/EIA-568-A standard (SunATM/P 155 UTP)
- Integrate PCI and SAR (segmentation and reassembly) functions in an ASIC implemented in standard CMOS
- SAR function aligned with ATM Forum specified and International Telecommunications Union-Telecommunication Sector (ITU-TS) approved ATM Adaptation Layer (AAL) 5
- Support 32- and 64-bit bus master interface
- Support both 33 MHz and 66 MHz clock speeds
- Support the SONET and SDH (Synchronous Optical Network/Synchronous Digital Hierarchy) physical layer framing structure
- Are compatible with relevant emerging standards (including existing ATM Forum baseline specifications and ITU-TS)

1.2.1.2 Hardware Requirements

To connect the SunATM/P 155 adapters to an ATM switch, you need the following cables:

- SunATM/P 155 MMF adapter: multimode fiber cable with an SC connector
- SunATM/P 155 UTP5 adapter: category 5 UTP with an RJ-45 connector

Refer to the manual supplied with the ATM switch for specific instructions about cable connections.

TABLE 1-3 shows the Sun-4U platforms that support SunATM/P 155 adapters.

TABLE 1-3 SunATM/P 155 Architecture and Supporting Systems

Platform Architecture	System Type
Sun-4u	Sun Ultra 30 Sun Ultra 60 Ultra 5 Ultra 10 Ultra Enterprise 250 Ultra Enterprise 450 Ultra Enterprise 3000/3500 Ultra Enterprise 4000/4500 Ultra Enterprise 5000/5500 Ultra Enterprise 6000/6500

1.2.1.3 Software Requirements

The SunATM/P 155 adapters are supported on systems running under Solaris 2.5.1 Hardware: 4/97, 8/97, 11/97, Solaris 2.6, or Solaris 7.

The SunATM 4.0 CD-ROM that shipped with the PCI adapter contains the required driver software that must be installed to connect a SunATM/P 155 adapter to a network.

1.2.2 SunATM/P 622 MMF Adapter

The SunATM/P 622 MMF adapter is a seven-inch adapter that conforms to the specifications of the Asynchronous Transfer Mode (ATM) Forum. The adapter offers 622 Mbps network bandwidth over a multimode fiber-optic cable.

The SunATM/P 622 MMF adapter is designed for operation in systems that run Solaris 2.5.1 Hardware: 4/97, 8/97, 11/97, Solaris 2.6, or Solaris 7. An on-board FCode PROM provides the configuration support that identifies the adapter to the system.

Note – For instructions on how to install the adapter, see Chapter 3 “Installing the SunATM PCI Adapters.”

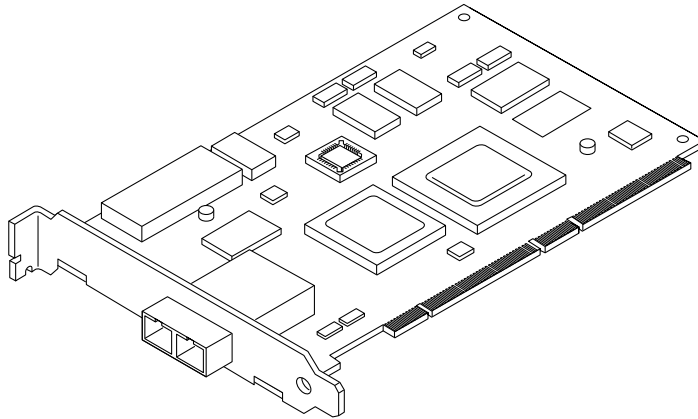


FIGURE 1-6 SunATM/P 622 MMF Adapter

1.2.2.1 Features

The SunATM/P 622 MMF adapter:

- Supports 622 Mbps operation over 62.5/125 μ multimode fiber
- Integrates PCI and SAR (Segmentation And Reassembly) functions in an ASIC implemented in standard CMOS
- SAR function aligned with ATM Forum specified and International Telecommunications Union-Telecommunication Sector (ITU-TS) approved ATM Adaptation Layer (AAL) 5
- Supports 32- and 64-bit bus master interface
- Supports both 33 MHz and 66 MHz clock speeds
- Supports SONET and SDH (Synchronous Optical NETWORK/Synchronous Digital Hierarchy) physical layer framing structure
- Is compatible with relevant emerging standards (including existing ATM Forum baseline specifications and ITU-TS)

1.2.2.2 Hardware Requirements

To connect the SunATM/P 622 adapter to an ATM switch, you will need a multimode fiber cable with an SC connector.

Refer to the manual supplied with the ATM switch for specific instructions about cable connections.

TABLE 1-4 shows the Sun-4u systems that support the SunATM/P 622 adapter.

TABLE 1-4 SunATM/P 622 Architecture and Supporting Systems

Platform Architecture	System Type
Sun-4u	Sun Ultra 30 Sun Ultra 60 Ultra 5 Ultra 10 Ultra Enterprise 250 Ultra Enterprise 450 Ultra Enterprise 3000/3500 Ultra Enterprise 4000/4500 Ultra Enterprise 5000/5500 Ultra Enterprise 6000/6500

1.2.2.3 Software Requirements

The SunATM/P 622 adapter and the SunATM 4.0 software are supported on systems running Solaris 2.5.1 Hardware: 4/97, 8/97, 11/97, Solaris 2.6, or Solaris 7.

The SunATM 4.0 CD-ROM that shipped with the adapter contains the software that must be installed to connect a SunATM/P 622 adapter to a network.

Installing the SunATM SBus Adapters

This chapter describes how to install the following three SunATM SBus adapters:

- SunATM/S 155 MMF adapter
- SunATM/S 155 UTP5 adapter
- SunATM/S 622 MMF adapter

Refer to your system's service and installation manuals for additional information about installing SBus adapters.

2.1 Installing the SunATM/S 155 Adapters

Note – The SunATM 4.0 software is supported on systems running Solaris 2.5.1, 2.6, or 7.

Up to two SunATM/S 155 adapters are supported per SBus. For example, on desktop machines that have only one SBus per system (even though there may be multiple SBus slots), no more than two SunATM/S 155 adapters are supported.

An extender plate (part number 560-1977-01) is included with the SunATM/S 155 adapter. In some older systems, you must attach this extender plate to the SBus adapter before installing the adapter. Refer to the hardware installation or service manual that shipped with your system for information about installing SBus adapters.

▼ SunATM/S 155 Adapter Installation

Refer to the hardware installation or service manual that shipped with your system for information about installing SBus adapters.



Caution – Make sure you use a wrist strap. It provides grounding for static electricity between your body and the system chassis. If you do not wear a wrist strap, the system components can be damaged by harmful electrical discharge.

Note – The SunATM/S 155 adapters are shipped with a rubber plug that keeps the connector free of dust during shipping and storage. Before installing the adapter, you must remove the plug from the SC fiber receptacle.

1. **Power off your system according to the instructions in your system administration guide and open the system unit.**
2. **Attach the adhesive copper strip of the wrist strap to the metal casing of the power supply. Wrap the other end twice around your wrist, with the adhesive side against your skin.**
3. **Holding the SBus adapter by the edges, unpack it and place it on an antistatic surface.**
4. **Remove the SBus filler panel from the slot in which you want to insert the SBus adapter.**
5. **Holding the SBus adapter by the edges, align the adapter edge connector with the SBus slot. Slide the adapter face plate into the small slot at the end of the SBus opening.**
6. **Applying even pressure at both corners of the adapter, push the SBus adapter until it is firmly seated in the slot.**



Caution – Do not use excessive force when installing the adapter into the SBus slot. You may damage the adapter's connector. If the adapter does not seat properly when you apply even pressure, remove the adapter and carefully reinstall it again.

7. **Detach the wrist strap and close the system unit.**

2.1.1 Verifying the SunATM/S 155 Adapter Installation

After you have installed the SunATM/S 155 adapter, but *before* booting the system, verify installation by executing the `show-devs` command.



Caution – Do not change the SBus slot in which a SunATM/S 155 adapter is installed once the system has been booted. Solaris remembers the location of each SBus adapter that has been installed. Switching SBus slots will tell the operating system that you removed your original SunATM/S 155 adapter and added a second adapter to the system. Refer to the `path_to_inst` man page for more information.

1. Power on the system and when the banner appears, press the Stop-A keys to interrupt the boot process. This will take you to the `ok` prompt.
2. Use the `show-devs` command to display the device information.

The `show-devs device path` command displays all devices known to the system directly beneath a given level in the device hierarchy. The SunATM/S 155 adapters are identified by the driver name `ba`.

The `show-devs` command used by itself shows the entire device tree. The abbreviated examples below display the `show-devs` output from a SPARCstation 10 system.

Note – If the system returns the `/SUNW,ba@3,0` entries in responses to both commands, the system recognizes the SunATM/S 155 adapter plugged into SBus slot 3.

```
ok show-devs /iommu/sbus
/iommu@f,e0000000/sbus@f,e0001000/SUNW,ba@3,0
. . .
```

```
ok show-devs
. . .
/iommu@f,e0000000/sbus@f,e0001000/SUNW,ba@3,0
. . .
```

Note – If you do not see the `ba` device listed, check that the adapter is properly seated and, if necessary, reinstall the adapter.

3. Reboot the system.

2.1.2 SunATM/S 155 MMF Adapter Wiring Configuration

The SunATM/S 155 MMF adapter is shipped with the SC connector already keyed. As you hold the SBus adapter with the connector pointed toward you, “transmit” is on the left and “receive” is on the right.

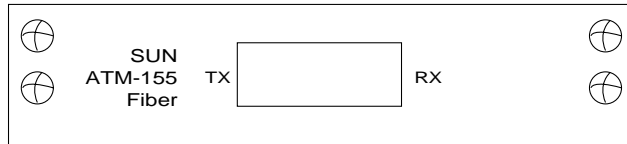


FIGURE 2-1 SunATM/S 155 MMF Adapter Backplate

▼ To connect the SunATM/S 155 MMF adapter to the network

1. **Connect one end of the multimode fiber cable into the fiber receptacle on the SBus adapter and connect the other end to the ATM switch (FIGURE 2-1).**

Refer to the installation or user’s manual supplied with the switch for additional information.

2. **After connecting the adapter to the ATM switch, install the SunATM software.**
See Chapter 4 “Installing the SunATM Software,” for more installation instructions.

2.1.3 SunATM/S 155 UTP5 Adapter Wiring Configuration

The SunATM/S 155 UTP5 adapter is shipped with the RJ45 connector already keyed for “transmit” (pair 2, pins 1 and 2) and “receive” (pair 4, pins 7 and 8) in accordance with the EIA/TIA (T568B) wiring scheme (see Appendix B “Wiring Scheme and Pin Descriptions”).

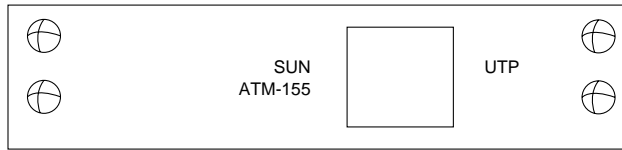


FIGURE 2-2 SunATM/S 155 UTP Adapter Backplate

▼ To connect the SunATM/S 155 UTP5 adapter to the network

1. Plug one end of the Category 5 UTP network cable into the RJ45 receptacle on the SBus adapter and connect the other end to the ATM switch (FIGURE 2-2).

Refer to the installation or user's manual supplied with the switch for additional information

2. After connecting the adapter to the ATM switch, install the SunATM software. See Chapter 4 "Installing the SunATM Software," for more installation instructions..

2.2 Installing the SunATM/S 622 MMF Adapter

Note – The SunATM 4.0 software is supported on systems running Solaris 2.5.1, 2.6, or 7.



Caution – Only one SunATM/S 622 adapter is supported per SBus. For example, on desktop machines that have only one SBus per system (even though there may be multiple SBus slots), no more than one SunATM/S 622 adapter is supported.

▼ SunATM/S 622 MMF Adapter Installation

Refer to the hardware installation or service manual that shipped with your system for information about installing SBus adapters.



Caution – Make sure you use a wrist strap. It provides grounding for static electricity between your body and the system chassis. If you do not wear a wrist strap, the system components can be damaged by harmful electrical discharge.

Note – The SunATM/S 622 MMF adapter is shipped with a rubber plug that keeps the connector free of dust during shipping and storage. Before installing the adapter, you must first remove the plug from the SC fiber receptacle (FIGURE 2-6).

Because of the large size of the SunATM/S 622 adapter's backplate (see FIGURE 2-4), you may need to follow these additional steps to correctly install the adapter.

1. **Power off your system according to the instructions in your system administration guide and open the system unit.**
2. **Attach the adhesive copper strip of the wrist strap to the metal casing of the power supply. Wrap the other end twice around your wrist, with the adhesive side against your skin.**
3. **Holding the SBus adapter by the edges, unpack it and place it on an antistatic surface.**
4. **Remove the SBus filler panel from the slot in which you want to insert the SBus adapter.**
5. **Insert the SunATM/S 622 adapter at an angle into the system by hooking the adapter under the top tabs on the system's back panel (FIGURE 2-3).**

Be sure the SC fiber receptacle goes through the slot on the back panel.

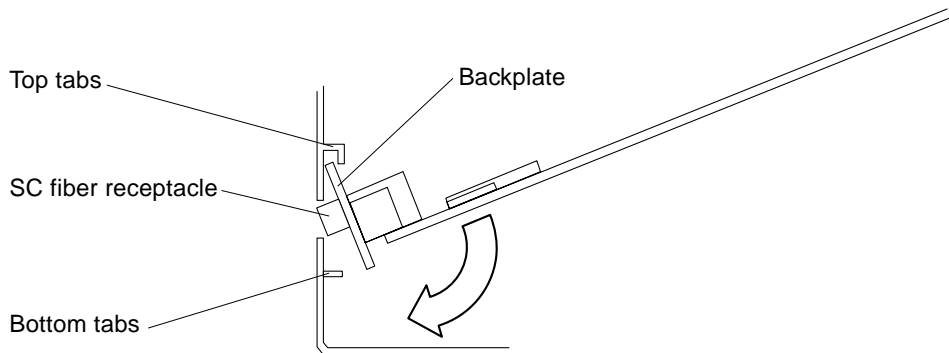


FIGURE 2-3 Installing the SunATM/S 622 MMF Adapter

6. **Align the adapter's SBus connectors with the system's SBus sockets, and gently lower the SBus adapter over the SBus sockets.**

7. Carefully slide the adapter to the left and to the right until the system's bottom tabs (see FIGURE 2-3) slide into the bottom recesses (see FIGURE 2-4) of the adapter's backplate.

Once the bottom tabs are in place, you should be able to lower the adapter easily into the system.

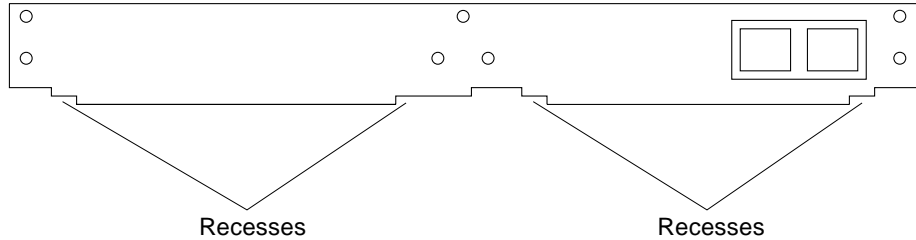


FIGURE 2-4 SunATM/S 622 MMF Adapter's Backplate

Do not force the adapter, or you may damage the pins on the connectors. If you are installing the adapter into a Sun Ultra 1 system, press only the right connector into the top SBus socket. On all other supported systems, press the two connectors into the two SBus sockets.

Note – Refer to your system's installation or service manual for more information about installing SBus adapters.

8. Detach the wrist strap and close the system unit.



Caution – If you are installing the SunATM/S 622 adapter into a Sun Ultra 1 Creator series or a Sun Ultra 1 series system, you must install the adapter in the top SBus slot (slot 1) of the system (see FIGURE 2-5).

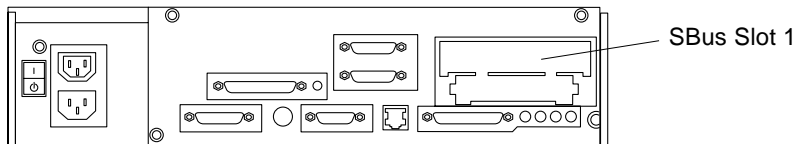


FIGURE 2-5 Sun Ultra 1 Series System Back Panel

Note – If you are installing the SunATM/S 622 adapter into a Sun Ultra 1 system, you will not be able to connect the adapter's two SBus connectors into two SBus sockets. Instead, you must install the right connector into the top SBus socket. The adapter will operate properly in a Sun Ultra 1 system if only the right connector is installed (FIGURE 2-6).

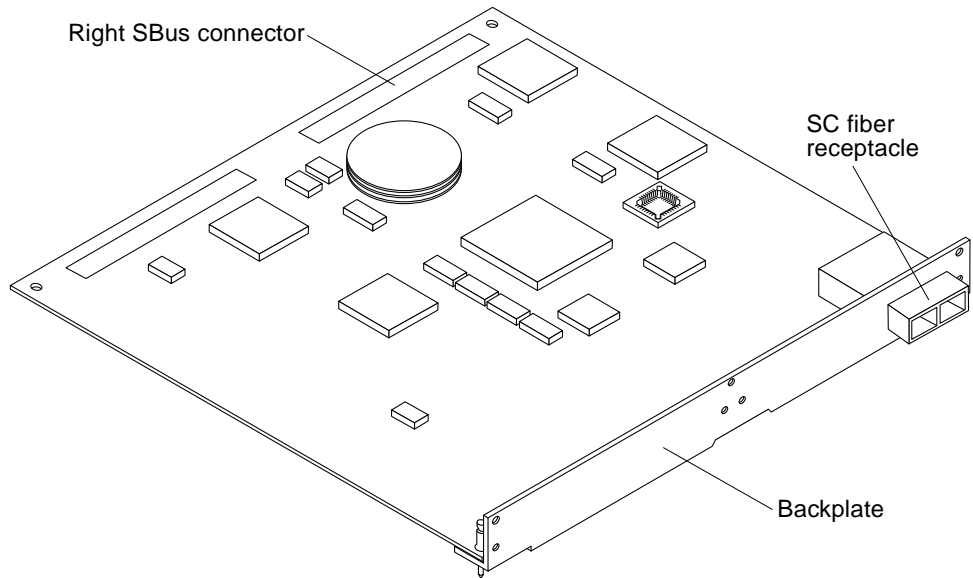


FIGURE 2-6 SunATM/S 622 MMF Adapter

2.2.1 Verifying the SunATM/S 622 Adapter Installation

After you have installed the SunATM/S 622 adapter, *but before booting the system*, verify installation by executing the `show-devs` command at the OpenBoot prompt.



Caution – Do not change the SBus slot in which a SunATM/S 622 MMF adapter is installed once the system has been booted. Solaris remembers the location of each SBus adapter that has been installed. Switching SBus slots will tell the operating system that you removed your original SunATM/S 622 adapter and added a second adapter to the system. Refer to the `path_to_inst` man page for more information.

1. **Power on the system and when the banner appears, press the Stop-A keys to interrupt the boot process. This will take you to the `ok` prompt.**

2. Use the `show-devs` command to display the device information.

The `show-devs device path` command displays all devices known to the system directly beneath a given level in the device hierarchy. The `show-devs` command used by itself shows the entire device tree. The SunATM/S 622 adapters 2.1 are identified by the driver name `ba`.

The abbreviated example below shows the `show-devs` output from a Sun Ultra Enterprise 3000 server with an OpenBoot PROM level 3.2.3.

```
ok show-devs /sbus
. . .
/sbus@3,0/SUNW,ba@2,0
. . .
```

Note – If you do not see the `ba` device listed, check that the adapter is properly seated and, if necessary, reinstall the adapter.

3. Reboot the system.

2.2.2 SunATM/S 622 MMF Adapter Wiring Configuration

The SunATM/S 622 MMF adapter is shipped with the SC fiber receptacle already keyed (FIGURE 2-4). As you hold the adapter with the receptacle pointed toward you, “transmit” is on the left and “receive” is on the right.

▼ To connect the SunATM/S 622 MMF Adapter to the network

1. Connect one end of the multimode fiber cable into the fiber receptacle on the adapter and connect the other end to the ATM switch.

Refer to the installation and user’s manual supplied with the switch for additional information.

2. After connecting the adapter to the ATM switch, install the SunATM software.

See Chapter 4 “Installing the SunATM Software,” for more installation instructions.



Caution – Do not boot the operating system until the installation is tested and verified. See Section 2.2.1 “Verifying the SunATM/S 622 Adapter Installation” for more information.

Installing the SunATM PCI Adapters

This chapter describes how to install the following three SunATM PCI adapters:

- SunATM/P 155 MMF adapter
- SunATM/P 155 UTP5 adapter
- SunATM/P 622 MMF adapter

Refer to your system's service and installation manuals for additional information about installing PCI adapters.

3.1 SunATM/P Adapter Installation

Note – The SunATM 4.0 software is supported on systems running Solaris 2.5.1 Hardware 4/97, 8/97, 11/97, Solaris 2.6, or Solaris 7.

Note – The SunATM/P adapters do not support JTAG. Remove the jumpers on the PCI I/O card when using SunATM/P on an Ultra Enterprise 3x00, 4x00, 5x00, or 6x00 system.

Note – Refer to your system installation or service manual for detailed instructions for the following tasks.

1. **Power off your system according to the instructions in your system administration guide and open the system unit.**

2. Attach the adhesive copper strip of the wrist strap to the metal casing of the power supply. Wrap the other end twice around your wrist, with the adhesive side against your skin.
3. Holding the PCI adapter by the edges, unpack it and place it on an antistatic surface.

Note – The SunATM/P MMF adapters are shipped with a rubber plug that keeps the SC connector free of dust. Remove this plug before installing the adapter into the system.

4. Remove the PCI filler panel from the slot in which you want to insert the SunATM PCI adapter.
5. Holding the PCI adapter by the edges, align the adapter edge connector with the PCI slot. Slide the adapter face plate into the small slot at the end of the PCI opening.
6. Applying even pressure at both corners of the adapter, push the PCI adapter until it is firmly seated in the slot.



Caution – Do not use excessive force when installing the adapter into the PCI slot. You may damage the adapter's PCI connector. If the adapter does not seat properly when you apply even pressure, remove the adapter and carefully reinstall it again.

7. Screw the adapter into position.
8. Detach the wrist strap and close the system unit.

3.2 Verifying the Installation

After you have installed the SunATM/P adapter, but before you boot your system, perform the following tasks to verify the installation. Refer to the Solaris Handbook for Sun Peripherals manual or your Solaris documentation for the detailed instructions.



Caution – Do not change the PCI slot in which a SunATM/P adapter is installed once the system has been booted. Solaris remembers the location of each PCI adapter that has been installed. Switching PCI slots will tell the operating system that you removed your original SunATM/P adapter and added a second adapter to the system. Refer to the man page about `path_to_inst` for more information.

1. Power on the system and, when the banner appears, press the Stop-A keys to interrupt the boot process and to get to the `ok` prompt.
2. Use the `show-devs` command to list the system devices.

You should see lines in the list of devices, similar to the example below, specific to the SunATM/P adapter: The SunATM/P adapters are identified by the hardware name `ma`.

```
ok show-devs
...
/pci@1f,4000/SUNW,ma@1
...
```

The `SUNW,ma@1` entry indicates that the SunATM adapter is seated in PCI slot 1.

Note – If you do not see the `ma` device listed, check that the adapter is properly seated and, if necessary, reinstall the adapter.

3. Reboot the system.

3.3 SunATM/P Adapter Wiring Configuration

3.3.1 SunATM/P 155 MMF Adapter Wiring Configuration

The SunATM/P 155 MMF adapter is shipped with the SC connector already keyed. As you hold the PCI adapter with the connector pointed toward you, “transmit” is on the left and “receive” is on the right.

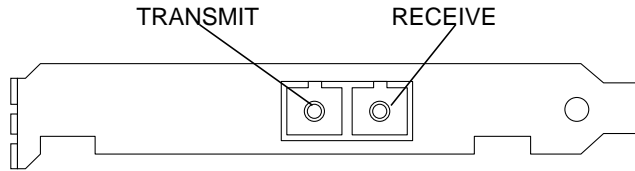


FIGURE 3-1 SunATM/P 155 MMF Adapter Backplate

▼ To connect the SunATM/P 155 MMF adapter to the network

1. **Connect one end of the multimode fiber cable into the fiber receptacle on the PCI adapter and connect the other end to the ATM switch.**

Refer to the installation or user's manual supplied with the ATM switch for additional information.

2. **After connecting the adapter to the ATM switch, install the SunATM software.**

See Chapter 4 "Installing the SunATM Software," for further installation instructions.

3.3.2 SunATM/P 155 UTP5 Adapter Wiring Configuration

The SunATM/P 155 UTP5 Adapter is shipped with the RJ45 connector already keyed for "transmit" (pair 2, pins 1 and 2) and "receive" (pair 4, pins 7 and 8) in accordance with the EIA/TIA (T568B) wiring scheme (see Appendix B "Wiring Scheme and Pin Descriptions").

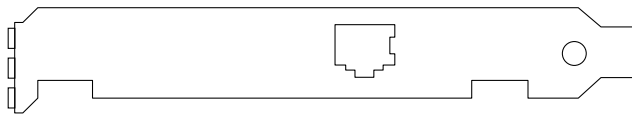


FIGURE 3-2 SunATM/P 155 UTP5 Adapter Backplate

▼ To connect the SunATM/P 155 UTP5 adapter to the network

1. **Plug one end of the Category 5 UTP network cable into the RJ45 receptacle on the PCI adapter and connect the other end to the ATM switch.**

Refer to the installation or user's manual supplied with the ATM switch for additional information.

2. **After connecting the adapter to the ATM switch, install the SunATM software.**

See Chapter 4 "Installing the SunATM Software," for further installation instructions.

3.3.3 SunATM/P 622 MMF Adapter Wiring Configuration

The SunATM/P 622 MMF adapter is shipped with the SC fiber receptacle already keyed. As you hold the adapter with the receptacle pointed toward you, "transmit" is on the left and "receive" is on the right.

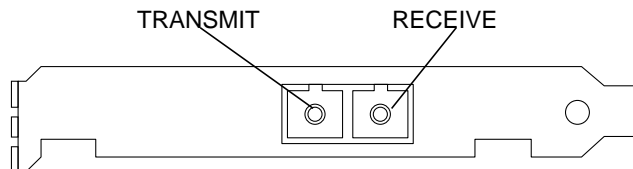


FIGURE 3-3 SunATM/P 622 MMF Adapter Backplate

▼ To connect the SunATM/P 622 MMF adapter to the network

1. **Connect one end of the multimode fiber cable into the fiber receptacle on the adapter and connect the other end to the ATM switch.**

Refer to the installation or user's manual supplied with the ATM switch for additional information.

2. **After connecting the adapter to the ATM switch, install the SunATM software.**

See Chapter 4 "Installing the SunATM Software," for further installation instructions.

Installing the SunATM Software

After you have installed the SunATM software, but *before* you reboot your system, you will need to configure the SunATM software. See *Chapter 5 “Configuring the SunATM Interface,”* for instructions on how to use the `atmadmin` configuration program.

4.1 Before Installing the SunATM Software

Before installing the SunATM software, remove any previous version of the SunATM software from your system. If you attempt to add the new software packages over existing SunATM packages, the installation will fail.

1. **Become superuser (root).**
2. **Check for any SunATM software packages by using the `pkginfo` command:**

```
# /usr/bin/pkginfo | grep SUNwatm
```

If you find any SunATM packages, you must remove them.

- **Remove any existing SunATM software packages by using the `pkgrm` command:**

```
# /usr/sbin/pkgrm SUNwatm SUNwatma SUNwatmu
```

4.1.1 Removing Links to Older SunATM 1.0 Device Entries

When you remove older versions of the SunATM software with the `pkgrm` utility, certain device entries in the `/dev` and `/devices` directories will not be removed. If you are installing the SunATM 4.0 software packages on a system in which a SunATM 1.0 SBus adapter had been installed, remove these device entries. The SunATM 1.0 hardware used the `sa` driver (which is not supported in the SunATM 4.0 software), and previous versions of the SunATM software created the `/dev/sa` and `/devices/pseudo/clone@0:sa` device entries.

To remove these SunATM 1.0 device entries, type:

```
# rm /dev/sa
# rm /devices/pseudo/clone@0:sa
```

4.1.2 Required Patches

There are software patches required for some versions of the Solaris operating environment on specific platforms. If you are installing the SunATM software on Solaris 2.5.1 or 2.6, you must install the appropriate patch prior to driver installation.

TABLE 4-1 Required Patches

Operating Environment	Patch Number	Platforms That Require Patch
Solaris 2.5.1	104003-02	SPARCstation 4 and SPARCstation 5
Solaris 2.6	105786-07*	All

* If you are unable to install this patch, the system administrator must add the following line to the `/etc/system` file, then reboot the system.

```
set ba:ba_dozerocopy=0
```

4.2 Installing the SunATM Software

4.2.1 Adding the Software Packages Using `pkgadd`

1. Become superuser (`root`).
2. Insert the SunATM CD-ROM into the CD-ROM player connected to your system.
 - If your system is running the Volume Management software, it should automatically mount the CD on the `/cdrom/sunatm_4_0` directory.
 - If your system is not running the Volume Management software, type the following to mount the CD-ROM:

```
# mkdir /cdrom
# mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /cdrom
```

3. Add the SunATM software packages.
 - If your system is running the Volume Management software, type:

```
# /usr/sbin/pkgadd -d /cdrom/sunatm_4_0/Product SUNWatm SUNWatmu
SUNWatma
```

- If your system is not running the Volume Management software, and you have mounted the CD-ROM as described in Step 2, type:

```
# /usr/sbin/pkgadd -d /cdrom/Product SUNWatm SUNWatmu SUNWatma
```

Note – For basic ATM functionality, the `SUNWatm` package is the only required software package. The `SUNWatmu` package contains the man pages and the files required to configure an ATM Simple Network Management Protocol (SNMP) management system. The `SUNWatma` package provides the SunATM interim Application Programmers' Interface (API) libraries and header files.

The SunATM packages will be installed in the following directories:

- SunATM Device Drivers and Utilities (SUNWatm) go into /kernel/mod, /kernel/mod/sparcv9, /kernel/drv, /etc/init.d, /platform/SUNW,Ultra-4FT/kernel/drv, /kernel/drv/sparcv9, /etc/opt/SUNWconn/atm, /etc/rc2.d, and /etc/opt/SUNWconn/bin.
- SunATM Runtime Support Software (SUNWatmu) goes into /opt/SUNWconn/atm and /opt/SUNWconn/man.

Note – Man pages contained in the SUNWatmu package go into /opt/SUNWconn/atm/man and will have symbolic links in /opt/SUNWconn/man. (Add /opt/SUNWconn/man to your system's \$MANPATH environment variable.) Interim API examples will go into /opt/SUNWconn/atm/examples.

- SunATM Interim API (SUNWatma) will go into /opt/SUNWconn/atm/include, /opt/SUNWconn/atm/lib, /opt/SUNWconn/include and /opt/SUNWconn/lib.

1. Eject the SunATM CD-ROM.

- If your system is running the Volume Management software and a window interface, click on the Eject Disk button on the /cdrom/sunatm_4_0 File Manager.
- If your system is running the Volume Management software without a window interface, type:

```
# cd /  
# eject cdrom
```

- If you are not running the Volume Management software, and you mounted the CD-ROM as described in Step 2, type:

```
# cd /  
# umount /cdrom  
# eject /dev/dsk/c0t6d0s0
```

Note – For more information about the Volume Management software, refer to the Solaris documentation.

2. Configure your SunATM interface(s).

You must complete the network configuration of your SunATM interface before you can use the interface. An interactive program, `/etc/opt/SUNWconn/bin/atmadmin`, is included with the SunATM software, and it may be used to configure your SunATM interfaces. See Chapter 5 “Configuring the SunATM Interface,” for instructions on how to use the `atmadmin` configuration program.



Caution – You *must* configure the SunATM software before rebooting your system.

3. Reboot your system and check the network.

See Section 4.3 “Rebooting the System and Examining Network Interfaces,” for more information.

4.2.2 Checking the Package Installation Using `pkgchk`

Once the package is installed, you can use the `pkgchk` command to see if the installation was complete:

```
# /usr/sbin/pkgchk SUNWatm
```

You can specify multiple packages at the command line by separating the package names with a space. If you do not specify a package identifier, the entire contents of the machine are checked.

4.2.3 Checking the Package Installation Using `pkginfo`

Check the ATM software installation by using the `pkginfo` command:

```
# /usr/bin/pkginfo | grep SUNWatm
system      SUNWatm  SunATM Device Drivers
application SUNWatma SunATM Interim Api Support Software
application SUNWatmu SunATM Runtime Support Software
```

4.2.4 Removing the Software Packages Using `pkgrm`

If you are superuser, you can remove one or more packages with the following command:

```
# /usr/sbin/pkgrm SUNWatm SUNWatma SUNWatmu
```

In this example, `pkgrm` removes the `SUNWatm` (SunATM Device Drivers and Utilities), `SUNWatma` (SunATM Interim API Support Software), and `SUNWatmu` (SunATM Runtime Support Software) packages.

4.3 Rebooting the System and Examining Network Interfaces

1. Reboot the system using the `boot -r` command.

The `-r` option is required by the Solaris software environment when installing new hardware. Use the `-v` option to display the boot messages, so you can see that the SunATM adapters are recognized correctly.

To start the boot process from the `ok` prompt, type:

```
ok boot -rv
```

Note – In Solaris 2.5.1, 2.6, or 7 operating environments, use `boot -r` whenever the physical configuration of the system is changed. Refer to the `boot(1M)` man page for more information.

2. Execute `ifconfig -a` and `netstat -i` commands to examine the state of all network interfaces.

You can also use the `/usr/sbin/ping` command to see if a network interface is active.

The following are examples of `ifconfig -a`, `ping`, and `netstat -i` output. Refer to the `ifconfig(1M)`, `ping(1M)`, and `netstat(1M)` man pages for more information.

```
example% /sbin/ifconfig -a
lo0: flags=849 <UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
      inet 127.0.0.1 netmask ff000000
ba0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 9180
      inet 129.144.130.9 netmask ffffffff broadcast 129.144.130.255
      ether 8:0:20:75:89:ff
```

```
example% /usr/sbin/ping zelda
zelda is alive
```

```
example% netstat -i
Name Mtu  Net/Dest      Address      Ipkts  Ierrs  Opkts  Oerrs  Collis  Queue
lo0   8232  loopback     localhost    1      0      1      0      0      0
ba0   9180  umtv20-130-n zardoz      5875   0      382812  0      0      0
```


Configuring the SunATM Interface

This chapter describes the new features in the SunATM software and how to configure the software using the `atmadmin` configuration program.

5.1 New Features in the SunATM Software

The SunATM software contains several new features. This section provides a brief overview of these new features.

5.1.1 Support for the Solaris 7 Operating Environment

The SunATM 4.0 software is compatible with the Solaris 7 operating environment. A major addition to the Solaris 7 operating environment is software support for 64-bit-capable hardware, such as Sun's Ultra platforms. SunATM 4.0 includes both 32-bit and 64-bit versions of its kernel drivers and modules. The 32-bit binaries will be installed in `/kernel/drv` and `/kernel/mod`; 64-bit binaries will be installed in `/kernel/drv/sparcv9` and `/kernel/mod/sparcv9`. The system will automatically select the appropriate drivers, depending on how it is booted.

Refer to the Solaris documentation that shipped with your system for more information.

5.1.2 Support for Dynamic Reconfiguration

The SunATM 4.0 driver includes a new utility, `/etc/opt/SUNWconn/bin/atmifconfig`, that allows individual interfaces to be configured and unconfigured without rebooting the system. This utility supports the Dynamic Reconfiguration features in the Solaris operating environment.

Refer to Section 6.6 “Plumbing and Unplumbing Individual ATM Interfaces” or the `atmifconfig(1M)` man page for more information on this utility.

5.1.3 Standardized Packaging Format

The packaging and installation format has been modified to comply with a standardized format for communications software. The SunATM software will be installed in two main locations: `/etc/opt/SUNWconn/atm` and `/opt/SUNWconn/atm`. In addition, symbolic links are created for items in bin directories under the atm subtree to the `/etc/opt/SUNWconn/bin` directory. This is also true for the man pages, installed in `/opt/SUNWconn/atm/man` and linked to `/opt/SUNWconn/man`.

These locations will be used for all communications software, allowing you to add a single location to your PATH and MANPATH variables and locate the software for many different products at that location.

One major implication of this change is that the location of the SunATM configuration files has changed. Those files are now located in `/etc/opt/SUNWconn/atm`. The SunATM 4.0 software installation process will automatically relocate existing configuration files to the new location, if necessary.

5.1.4 Unified Driver support for PCI and SBus

The SunATM 4.0 software brings together support for both PCI and SBus in a single device driver, the `ba` driver.

If you have an SBus adapter and are upgrading from SunATM 2.1 software, there were several new features in the PCI software release that are now available for SBus.

Those features include:

- Support for the ATM Forum’s UNI and ILMI 4.0 specifications.
- Automatic LECS Address detection via ILMI, for LAN Emulation.

5.2 Using the `atmadmin` Configuration Program

The SunATM configuration program, `atmadmin`, is an interactive command-line interface. The program contains a hierarchy of menus, which divide the configuration into six main parameter groups: system, physical layer, signalling, ILMI, Classical IP, and LAN Emulation. All but the system parameter group are specific to individual SunATM interfaces, so you must configure the parameters in these groups separately for each interface.

If you prefer, you can enter and change the SunATM configuration information by editing the SunATM configuration files directly. See Chapter 6 “Editing SunATM Configuration Files,” for a description of the configuration files’ contents and formats.

Note – See the Glossary for descriptions of the ATM and SunATM terms used in this chapter. Chapter 6 “Editing SunATM Configuration Files,” and Chapter 7 “Classical IP and LAN Emulation Protocols,” also provide more information about ATM protocols and the SunATM implementation of these protocols.

5.2.1 Starting the `atmadmin` Configuration Program

The `atmadmin` program is installed with the `SUNWatm` software package in the `/etc/opt/SUNWconn/bin` directory. The program must be run as superuser (`root`). It can be run in any local or remote shell on the SunATM system.

```
# /etc/opt/SUNWconn/bin/atmadmin
```

5.2.2 atmadmin Main Menu

After you start the `atmadmin` configuration program, you see the `atmadmin` Main Menu. From this menu you can either go to the system parameter group menu (see Section 5.2.4 “System Parameter Group Menu”), or you can enter the SunATM interface you want to configure. The following is screen example is from a system with one interface named `ba0`.

```
Welcome to the SunATM Admin Program.
The following interfaces are installed in your system:
    ba0
    [S] Modify System Parameters
    [X] Exit
    [?] Help

Enter interface name or option: ba0
```

After selecting an interface, you will then see the Interface Configuration menu (see Section 5.2.5 “Interface Configuration Menu”).

5.2.3 atmadmin Navigation Commands

TABLE 5-1 lists the basic commands that let you navigate through the menu hierarchy.

TABLE 5-1 Basic Navigational Commands in `atmadmin`

Command	Action
<code>m</code>	Return to the <code>atmadmin</code> main menu
<code>p</code>	Return to the previous menu
<code>x</code>	Exit <code>atmadmin</code>
<code>?</code>	Provide more information about the options on this menu

5.2.4 System Parameter Group Menu

The system parameter group contains parameters that are not specific to an interface; they apply to the entire system. The following example shows the system parameter group menu.

```
Modifying system-wide parameters;
Currently configured as an ATM SNMP agent, using UDP port 161
The SNMP agent options are:
    [A] ATM SNMP agent
    [N] not an agent
    [U] UDP Port

    [P] Previous Menu
    [M] Main Menu
    [X] Exit
    [?] Help

Enter selection:
```

5.2.4.1 ATM SNMP Agent Status

You can configure your SunATM system as an ATM SNMP agent. The SunATM SNMP daemon, `atmsnmpd`, always runs on an ATM host. If you do not run your system as an SNMP agent, the daemon does not bind to a UDP port.

Note – See Section 7.4 “ATM and SNMP” on page 7-8 for more information about `atmsnmpd` options.

5.2.5 Interface Configuration Menu

Once you select a SunATM interface, you will see the `atmadmin` Interface Configuration menu. From this menu you can proceed to the interface parameter group sub-menus, which are described in Section 5.3 “`atmadmin` Parameter Groups.” You can use these sub-menus to change the SunATM interface configuration parameters.

```
Modifying ba0
  [Y] Physical Layer
  [U] UNI Signalling
  [I] ILMI Address Registration
  [C] Classical IP
  [L] LAN Emulation

  [P] Previous Menu
  [M] Main Menu
  [X] Exit
  [?] Help

Enter selection:
```

5.2.5.1 `atmadmin` and the SunATM Configuration Files in the `/etc/opt/SUNWconn/atm` directory

The `atmadmin` program first attempts to read the current configuration information from the `/etc/opt/SUNWconn/atm/atmconfig`, `/etc/opt/SUNWconn/atm/aarconfig`, and `/etc/opt/SUNWconn/atm/lanconfig` files. If no configuration information is found, or if the files do not exist, the default values listed in TABLE 5-2 are applied to the installed interfaces.



Caution – When saving configuration information, `atmadmin` *overwrites* the existing SunATM configuration files in the `/etc/opt/SUNWconn/atm` directory. Therefore, any comments or other changes you manually made to the files will be lost.

5.3 atmadmin Parameter Groups

The atmadmin configuration program contains a series of menus where you can input or alter the configuration of specific SunATM software parameters. These menus, or parameter groups, are described in this chapter:

- Physical Layer Parameter Group—page 5-9
- Signalling Parameter Group—page 5-10
- ILMI Parameter Group—page 5-11
- Classical IP Parameter Group—page 5-11
- LAN Emulation Parameter Group—page 5-15

TABLE 5-2 summarizes the configurable parameters in each parameter group. Although the parameter list appears rather lengthy, you only need to use the default values for most standard configurations. The large number of parameters offer the flexibility to support special case configurations, and to allow interoperability with equipment from other vendors.

Note – In most cases, you will only need to configure the parameters that do not have default values.

TABLE 5-2 Configurable Parameters in the SunATM Software

Group	Parameters	Possible Values	Default Values	Required?
System	SNMP Agent Status	agent or not_agent	not_agent	Yes
	SNMP Agent UDP port	0 <= n <= 65355	161 or 1000	For SNMP Agent
Physical Layer	Framing Interface	SONET or SDH	SONET	Yes
Signalling	UNI Version	3.0, 3.1, 4.0, or none	No default	Yes
ILMI	ILMI Status	Enabled or Disabled	Enabled	Yes
Classical IP	Hostname/IP Address	Valid hostname and IP address	No default	For Classical IP
	Interface Type	Client, Server, or Standalone	No default	For Classical IP

TABLE 5-2 Configurable Parameters in the SunATM Software (Continued)

Group	Parameters	Possible Values	Default Values	Required?
	Local ATM Address	Valid ATM address	\$myaddress	For Classical IP Clients or Servers
	ARP Server	Valid ATM address	\$localswitch_server	For Classical IP Clients
	PVC	32 <= n < 1024	32	For Classical IP Standalones
	Destination hostname or IP address	Valid hostname and IP address	No default	For Classical IP Standalones
LAN Emulation	Instance Number	0 <= n <= 999	No default	For LAN Emulation
Per-Instance Parameters	Hostname/IP Address	Valid hostname and IP address	No default	For LAN Emulation
	Local ATM Address	Valid ATM address	\$myaddress	For LAN Emulation
	LECS Indicator	No LECS or LECS Present	LECS Present	For LAN Emulation
	LECS ATM Address	Valid ATM address	ILMI value or the well-known LECS address	For LAN Emulation, lecs_present
	LES ATM Address	Valid ATM address	No default	For LAN Emulation, no_lecs
	Emulated LAN Name	Character string	No default	For additional instance on a physical interface
	Additional Hostnames?	Yes or no	No	For LAN Emulation
Per-Additional Hostname	Minor Instance Number	0 <= n <= 8190 ¹	None	For LAN Emulation, additional IP
	Hostname/IP Address	Valid hostname and IP address	No default	For LAN Emulation, additional IP

1. The maximum value is 255 for versions of Solaris earlier than 2.6. For Solaris 2.6 and compatible, the value can go to a maximum of 8190.

5.3.1 Physical Layer Parameter Group

The physical layer parameter group contains only the framing interface parameter. The following example shows the physical layer parameter menu.

```
Modifying ba0; Current framing interface is SONET
The framing interfaces that may be configured are:
    sonet
    sdh

    [P] Previous Menu
    [M] Main Menu
    [X] Exit
    [?] Help

Enter selection:
```

5.3.1.1 Framing Interface

The framing interface defines the encapsulation method used for ATM cells as they are sent onto the wire. The default framing interface is SONET, but the SunATM software also supports the SDH interface. Your switch product information should indicate whether your switch uses either the SONET or the SDH interface. If the switch uses the SDH interface, you will need to select SDH from the physical parameter group menu.

5.3.2 Signalling Parameter Group

The signalling parameter group contains only the UNI version parameter. The following example shows the signalling parameter menu.

```
Modifying ba0; Current UNI Version is 3.0
The UNI versions that may be configured are:
    3.0
    3.1
    4.0
    [N] No Signalling Enabled

    [P] Previous Menu
    [M] Main Menu
    [X] Exit
    [?] Help

Enter selection:
```

5.3.2.1 UNI Version

The SunATM software supports three versions of the ATM Forum's User Network Interface (UNI) Specification: versions 3.0, 3.1, and 4.0. You may choose not to enable signalling, but in order to support either Classical IP or LAN Emulation (or both), you must select one of the three UNI versions.

5.3.3 ILMI Parameter Group

If your ATM switch does not support the Interim Local Management Interface (ILMI), you can turn off the ILMI address registration on your SunATM interface from the ILMI configuration menu. The following example shows the ILMI configuration menu.

```
Modifying ba0; Currently ILMI is enabled
  [E] Enable ILMI
  [D] Disable ILMI

  [P] Previous Menu
  [M] Main Menu
  [X] Exit
  [?] Help

Enter selection:
```

5.3.4 Classical IP Parameter Group

Classical Internet Protocol (Classical IP), specified by RFC 1577, is one way of supporting the TCP/IP and UDP/IP protocols in an ATM environment. In Classical IP, an ATM ARP server is used to resolve IP addresses to ATM addresses, replacing the traditional ARP protocol. In this configuration, each host must register with the ARP server when the ATM interface is brought up. For more information on the Classical IP protocols, see Section 7.2 “Classical Internet Protocol.”

One reason ATM ARP is used instead of the traditional ARP is that ATM does not support broadcast (a network capability providing transmission from one point to all points on a network). Because Classical IP over ATM does not support broadcast, you cannot use the `ypbind -broadcast` UNIX command to automatically locate the NIS server (`ypserver`) on a Classical IP ATM subnet.

If you are planning to run NIS over your ATM network, you must specify the list of NIS servers (`ypservers`) using the `ypinit -c` command. See the `ypinit(1M)` man page for details of setting up the `ypserver`. Be sure that the IP addresses of the `ypservers` are listed in the `/etc/hosts` file.

The Routing Information Protocol (RIP) also uses the broadcast feature of IP, so it is not supported under the Classical IP environment. In the Solaris operating environment, RIP is implemented by the daemon `in.routed`.

Classical IP alone also does not support the multicast packet delivery system. If you are using Classical IP, you must explicitly add the routes to the routers in the ATM subnet. You may also specify one router as the default router to provide connectivity outside of the ATM subnet. See the `route(1M)` man page for information on using the `route` command to add specific router entries and to add a default router.

You can use the Classical IP parameter group menu to define the Classical IP configuration of a SunATM interface.

```
Modifying ba0; Current Configuration:
  Arp Client
  IP = atm_cip
  ATM = $myaddress
  ARPSRV = $localswitch_server
    [N] No Classical IP Enabled
    [C] Client
    [S] Arp Server
    [T] Standalone
    [I] Hostname or IP Address
    [L] Local ATM Address
    [A] ATM ARP Server Address

    [P] Previous Menu
    [M] Main Menu
    [X] Exit
    [?] Help

Enter Selection:
```

5.3.4.1 Classical IP Interface Type

The SunATM software allows you to configure your interface as either a Classical IP ARP server or a client. In addition, you can connect two systems back-to-back, in a standalone configuration, using a Permanent Virtual Circuit (PVC). These three modes are options on the Classical IP parameter menu.

5.3.4.2 Hostname and IP Address

Regardless of the Classical IP Interface Type, you must assign an IP address and hostname to the interface. If you enter a hostname that appears in the `/etc/hosts` file, or if NIS, NIS+, or DNS is enabled and the hostname is resolvable over it, you are not prompted to enter an IP address. Instead, the resolution is performed

automatically. If the hostname cannot be resolved, you are prompted to enter an IP address. If you must enter an IP address, or if the address is only available through NIS, NIS+, or DNS, the SunATM software updates the `/etc/hosts` file.

A valid IP hostname is no more than 80 characters. A valid IP address is a set of four decimal numbers in the range of 0 to 255, separated by dots (for example, 149.144.130.9).

5.3.4.3 Local ATM Address

The local ATM address is the 20-byte ATM address associated with a specific Classical IP instance. You must assign an ATM address to each Classical IP client and server, but you do not need to assign an ATM address on standalone (back-to-back) configurations. The following section describes ATM address formats and some of the SunATM software defined address variables.

ATM Address Formats and Variables

ATM addresses, like Network Service Access Point (NSAP) addresses, are 20 octets long, with each octet made up of 1 or 2 hexadecimal digits. The ATM address is divided into three fields: the End System Identifier field, the Selector field, and the Network Prefix field. The End System Identifier (ESI) field is a unique 6 octet value, which can be the IEEE hardware MAC address conventionally associated with every network interface. The Selector field is one octet long. The 13 octets that make up the rest of the ATM address are called the Network Prefix. This field should be derived from the ATM switch fabric to which the interface is connected. Every ATM switch fabric is configured with a 13 octet prefix.

To simplify references to ATM addresses in the SunATM software, several system-defined variables are built into the software. Variables are referenced with the `$` operator, as in UNIX shell scripts. TABLE 5-3 summarizes the system-defined SunATM ATM address variables.

TABLE 5-3 Predefined SunATM Variables

Variable	Description
<code>prefix</code>	The 13-byte prefix associated with the local switch.
<code>mac</code>	The 6-byte medium access control (MAC) address associated with the local host or interface.
<code>sel</code>	The default 1-byte selector for the local interface.
<code>macsel</code>	The concatenation of <code>\$mac:\$sel</code> .

TABLE 5-3 Predefined SunATM Variables (Continued)

Variable	Description
myaddress	Concatenation of \$prefix:\$mac:\$sel, resulting in the default address for the local interface.
sunmacselN	Concatenation of one of a series of reserved MAC addresses and \$sel to create a block of reserved ATM ARP server addresses. N should be a decimal number in the range 0 - 199.
localswitch_server	Concatenation of \$prefix, a unique reserved MAC address, and \$sel. When used as a server address, restricts server access to clients connected to the local switch only.

Note – The \$prefix variable, and any other variables that use it (including \$myaddress and \$localswitch_server), may not be used on interfaces that are not running ILMI.

ATM addresses are represented by 20 colon-separated octets, with each octet made up of 1 or 2 hexadecimal digits. You can combine variables representing portions of an ATM address with other variables and/or octets to make up a complete address. For example, \$prefix:aa:bb:cc:dd:ee:ff:\$sel represents a valid ATM address.

5.3.4.4 ATM ARP Server Address

If you configured the Classical IP instance as a client, you must also enter the address of the ARP server. This parameter, like the local ATM address, must be a 20-byte ATM address. See “ATM Address Formats and Variables” on page 13 for a discussion of ATM address formats and variables.

5.3.4.5 Permanent Virtual Circuit (PVC)

The Permanent Virtual Circuit parameter applies only to standalone configurations. It identifies the PVC which will be used to communicate between the two systems connected back-to-back. Both systems must use the same PVC value. The PVC parameter must be a decimal integer between 32 and 1023.

5.3.5 LAN Emulation Parameter Group

LAN Emulation, standardized by the ATM Forum's LAN Emulation 1.0 specification, is another way of providing TCP/IP and UDP/IP support over an ATM interface. Address resolution information is provided by a series of LAN Emulation services. When a LAN Emulation interface is brought up it must register with these LAN Emulation services (known as "joining the LAN"). This registration process and the address resolution process are described in Section 7.3 "LAN Emulation" on page 7-5.

Unlike Classical IP, the LAN Emulation protocol provides a broadcast service to the upper layer protocols. Therefore, the multicast and RIP limitations described in Section 5.3.4 "Classical IP Parameter Group", do not affect LAN Emulation interfaces.

The SunATM software allows a single ATM interface to join up to sixteen emulated local area networks (ELANs), provided that this action is allowed by the switch and LAN Emulation (LANE) services. Each ELAN joined is represented by a unique lane instance (e.g. lane0 or lane1).

Note – A requirement for supporting this feature is that the adapter card be assigned multiple MAC addresses, which the SunATM/S 2.1 and SunATM/P 3.0 adapters support. This feature *does not* work with the older SunATM/S 2.0 adapters. Use the `atmgetmac (1M)` command with the `count` option to find the number of MAC addresses assigned to your SunATM adapter.

After you configure LAN Emulation parameters, you are asked to choose an existing (previously configured) LAN Emulation (lane) instance or to create a new one in the LAN Emulation Instance menu. The following is an example of this menu.

```
The following lane instances are configured on ba0:
    lane0
    lane1
    [C] Create new lane instance
    [D] Delete lane instance

    [P] Previous Menu
    [M] Main Menu
    [X] Exit
    [?] Help

Enter lane instance or option: lane0
```

5.3.5.1 Per-Instance LAN Emulation

The Per-Instance LAN Emulation Parameters menu allows you to configure the per-instance LAN Emulation parameters.

```
Modifying lane0; Current Configuration:
```

```
IP = atm_lane
ATM = $myaddress
LECS Present
LECS_Address = well-known address
no additional IP hostnames
  [I] Hostname or IP Address
  [L] Local ATM Address
  [C] LECS Present
  [N] No LECS
  [A] LECS ATM Address
  [E] Emulated LAN Name
  [H] Additional Hostnames

[P] Previous Menu
[M] Main Menu
[X] Exit
[?] Help
```

```
Enter Selection:
```

Hostname or IP Address

If IP traffic runs over a LAN Emulation instance, assign a hostname and corresponding IP address to the instance. If you enter a hostname that appears in the `/etc/hosts` file, or if NIS, NIS+, or DNS is enabled and the hostname is resolvable over it, you are not prompted to enter an IP address. Instead, the resolution is performed automatically. If the hostname cannot be resolved, you are prompted to enter an IP address. If you must enter an IP address, or if the address is only available through NIS, NIS+, or DNS, the SunATM software updates the `/etc/hosts` file.

A valid hostname is no more than 80 characters. A valid IP address is a set of four decimal numbers in the range of 0 to 255, separated by dots (for example, 149.144.130.9).

Local ATM Address

The local ATM address is the 20-byte ATM address associated with this LAN Emulation instance. See “ATM Address Formats and Variables” for more information about ATM address formats and variables.

Each `lane` instance must be assigned a unique ATM address. Each SunATM 2.1 or 3.0 adapter has been assigned 16 unique MAC addresses; if you use the variable `$myaddress` for each `lane` instance, the SunATM software will automatically distribute those MAC addresses to the `lane` instances associated with each physical interface.

LECS Indicator

Most LAN Emulation Services include a LAN Emulation Configuration Server (LECS), which is the first server contacted when bringing up a LAN Emulation client. The LECS provides the ATM address of the LAN Emulation Server (LES), as well as other configuration information about the emulated LAN. However, some LAN Emulation services do not include an LECS, and the LES must be contacted directly. With the LECS Indicator parameter, you specify which service should be contacted first in your configuration. The possible values for this parameter are displayed as individual options on the LAN Emulation Instance menu.

Note – If the value of this parameter is `NO LECS`, you must specify a value for the LES ATM Address parameter.

LECS ATM Address

By default, the SunATM software attempts to obtain the LECS address using ILMI, as specified in the LAN Emulation specification. If this is not successful, the “well-known” ATM address, also specified by the ATM Forum, is used.

If your LECS uses a different ATM address (not the well-known address), and does not make that address available via ILMI, specify it using this parameter. If applicable, any of the ATM address variables described in “ATM Address Formats and Variables” may be used. Use variable `$prefix`, in particular.

LES ATM Address

This parameter is required if the value of the LECS Indicator parameter is `no_LECS`. There is no “well-known” address for the LES, so an ATM address must be specified for the LES since there is not an LECS present to provide one. This parameter is a

standard ATM address. If any of the SunATM address variables described in Section 5.3.4.3 under “ATM Address Formats and Variables” (\$prefix in particular) are applicable, they can be used.

Emulated LAN Name

If multiple Emulated LANs (ELANs) are present, you can enter a character string in the Emulated LAN Name parameter. The LAN Emulation client uses this parameter to tell the LAN Emulation services which ELAN it wishes to join. By default, if a SunATM LAN Emulation client does not specify an ELAN name, it tells the services to assign it to the default (or only) ELAN.

Note – If you have multiple LAN Emulation instances configured on a physical interface, only one instance can join the default (unspecified) ELAN. You must specify an ELAN name for all other instances.

Additional Hostnames

The SunATM software supports logical interfaces in the SunATM LAN Emulation environment. Logical interfaces allow you to assign multiple IP addresses to a single LAN Emulation interface. A logical interface name consists of three parts: the device name (in the case of SunATM LAN Emulation, `lane`); the major number, which corresponds to the `lane` instance number; and the minor number, which distinguishes the logical interfaces on a single `lane` instance. The format of a LAN Emulation logical interface name is `laneN:X`, where *N* is the major number and *X* is the minor number (for example, `lane0:2`).

The SunATM software associates each logical interface with a unique hostname and IP address. All logical interfaces on a given physical interface are associated with the same ATM and MAC addresses.

The hostname displayed in the LAN Emulation instance menu corresponds to the minor instance 0. The additional IP Address parameter indicates if any additional hostnames are assigned to the instance. Select this parameter to modify or create additional hostnames. You must enter or modify each additional IP hostname in the same manner as other IP hostname and address pairs (see “ATM Address Formats and Variables” for more details), and associate it with a minor number between 0 and 255.

Editing SunATM Configuration Files

This chapter describes how to configure SunATM interfaces by editing the configuration files.

You are not required to edit these configuration files by hand. You can use the `atmadmin` configuration program, described in Section 5.2 “Using the `atmadmin` Configuration Program,” to configure the SunATM files. From the program’s command-line interface, you can change most of the SunATM parameters.



Caution – When it saves configuration information, `atmadmin` *overwrites* the existing SunATM configuration files in the `/etc/opt/SUNWconn/atm/` directory. Therefore, any comments or other changes you manually made to the files will be lost.

6.1 Editing the `/etc/opt/SUNWconn/atm/atmconfig` File

The `/etc/opt/SUNWconn/atm/atmconfig` file is a generic file that must appear on every SunATM system. It provides general configuration information used by the SunATM setup utilities to bring up ATM interfaces at boot time.

The file consists of one or more entries per interface. An entry contains the following fields:

TABLE 6-1 /etc/opt/SUNWconn/atm/atmconfig Field Descriptions

Field	Description
Interface	Physical interface, baN.
UNI/Framing	Version of the UNI specification used for signalling, 3.0, 3.1, or 4.0; or, for entries with only two fields, the Framing Interface, SONET or SDH.
CIP_Host	Hostname used for Classical IP.
LANE_Instance	Instance number for a LAN Emulation interface; LAN Emulation interfaces will be called laneN where N is the LAN Emulation instance number. The LANE instance number must be between 0 and 999. Note: The LANE instance number is not necessarily the same as the physical instance number.
LANE_Host	Hostname used for LAN Emulation

The Interface and UNI fields are required for all interfaces. The CIP_Host field is required for interfaces that run Classical IP, and the LANE_Instance and LANE_Host fields are required for interfaces that run LAN Emulation. If a field is not used, it is represented by a hyphen.

Modifications to individual interfaces will take effect when the interface is plumbed. This will happen either at boot time or when you use the `atmifconfig` utility to plumb interfaces. If you modify an existing (already running) interface, you must first unplumb it with the `atmifconfig` utility. Refer to the `atmifconfig(1m)` man page or “Plumbing and Unplumbing Individual ATM Interfaces” on page 22 for more information.

6.1.1 Changing the Framing Interface in the /etc/opt/SUNWconn/atm/atmconfig File

The framing interface defines the encapsulation method used for ATM cells as they are sent onto the wire. The default framing interface is SONET, but the SunATM software also supports the SDH interface. Your switch product information should indicate which interface your switch uses.

Previous versions of the SunATM software allowed you to choose a framing interface for the entire system (by setting a variable in the `/etc/system` file). In the SunATM software, the system variable can be used to allow backwards compatibility, but the preferred method is to select the framing interface for each

interface and to add an entry in the `/etc/opt/SUNWconn/atm/atmconfig` file. An entry in `/etc/opt/SUNWconn/atm/atmconfig` overrides a variable set in `/etc/system` for a particular interface. If there is no value in either `/etc/system` or `/etc/opt/SUNWconn/atm/atmconfig`, the default framing interface is SONET.

Framing entries in `/etc/opt/SUNWconn/atm/atmconfig` should appear on individual lines, with two fields. The first field indicates the interface, `baN`, where `N` is the instance number (for example: `ba0`). The second is either `SDH` or `SONET`, depending on the desired setting.

6.1.2 Example of an `/etc/opt/SUNWconn/atm/atmconfig` File

The following sample `atmconfig` file creates this configuration:

- A LAN Emulation interface `lane0`, supporting UNI 3.1, on the `ba0` interface.
- An interface that supports both Classical IP and LAN Emulation on `ba1`, using UNI 4.0. The LAN Emulation interface name is `lane1`.
- A Classical IP interface, supporting UNI 3.0, on `ba2`, which uses the SDH framing interface.

#Interface	UNI/Framing	CIP_Host	LANE_Instance	LANE_Host
#-----				
ba0	3.1	-	0	atm0
ba1	4.0	atm1	1	atm2
ba2	3.0	atm3	-	-
ba2	SDH			

6.2 Configuring a Classical Internet Protocol Interface

Classical Internet Protocol (Classical IP), specified by RFC 1577, is one way of supporting the TCP/IP and UDP/IP protocols in an ATM environment. In Classical IP, an ATM ARP server is used to resolve IP addresses to ATM addresses, replacing the traditional ARP protocol. In this configuration, each host must register with the ARP server when the ATM interface is brought up. For more information on the Classical IP protocols, see Section 7.2 “Classical Internet Protocol.”

ATM ARP is used instead of the traditional ARP because ATM does not support broadcast (a network capability providing transmission from one point to all points on a network). Because Classical IP over ATM does not support broadcast, you cannot use the `ypbind` UNIX command with the `-broadcast` option to automatically locate the NIS server (`ypserver`) on a Classical IP ATM subnet.

If you are planning to run NIS over your ATM network, use the `ypinit -c` command to specify the list of NIS servers (`ypservers`). See the `ypinit(1M)` man page for details of setting up the `ypserver`. Be sure that the IP addresses of the `ypservers` are listed in the `/etc/hosts` file.

The Routing Information Protocol (RIP) also uses the broadcast feature of IP, so it is not supported under the Classical IP environment. In the Solaris operating environment, RIP is implemented by the daemon `in.routed`.

If you are using Classical IP only, you must explicitly add the routes to the routers in the ATM subnet. You can also specify one router as the default router to provide connectivity outside of the ATM subnet. See the `route(1M)` man page for information on using the `route` command to add specific router entries and to add a default router.

6.2.1 Editing the `/etc/opt/SUNWconn/atm/aarconfig` File

The `/etc/opt/SUNWconn/atm/aarconfig` file is a generic file that must appear on every SunATM system which is supporting Classical IP interfaces. It allows you to specify IP to ATM address translation, permanent virtual circuits (PVCs) to destinations, and the address of the ATM ARP server. The environment allows for a mix of PVCs and switched virtual circuits (SVCs).

Each time the `/etc/opt/SUNWconn/atm/aarconfig` file is modified, run the ATM ARP setup program, `aarsetup`, which is in the `/etc/opt/SUNWconn/bin` directory.

Every node, or client, has both an IP address and either an ATM address or a virtual circuit identifier (VCI). See Section 7.2.1 “ATM Address Resolution”, for ATM addressing scheme information.

In the IP-ATM address table shown in the `/etc/opt/SUNWconn/atm/aarconfig` file:

- `Interface` is the last part of the device name (`ba0`, for example).
- `Hostname` is either an IP address in “dot” notation or the name of a host that should be locally available unless a non-ATM network connection also exists.
- `ATM Address` consists of 20 octets with each octet represented by a one- or two-digit hexadecimal number and separated by colons.
- The `VCI` field is a positive decimal integer.

- An unused field is denoted by a hyphen.

TABLE 6-3 on page 6-6 lists the flags, including configuration flags, and the options they provide.

TABLE 6-2 /etc/opt/SUNWconn/atm/aarconfig File Flag Descriptions

Flag	Description
l	Represents the ATM address of the local interface on ARP clients or systems not using an ARP server for ATM address resolution, and can be used to assign an ATM address to the host. <i>Hostname</i> should not appear; <i>ATM Address</i> should be provided if, and only if, SVCs are used. If you provide an <i>s</i> entry to use an ARP server (see below), you must also provide an <i>ATM Address</i> (a server is meaningful only in an SVC environment). See TABLE 6-3.
L	Represents the ATM address of the local interface on an ARP server. <i>Hostname</i> should not appear; <i>ATM Address</i> is required. See TABLE 6-3.
s	Specifies a connection to the ATM ARP server. Either <i>ATM Address</i> or <i>VCI</i> (in the case of a PVC connection) should appear, but not both. <i>Hostname</i> should not appear. The <i>s</i> entry is required on all clients that need to communicate with the server for ATM address resolution. See TABLE 6-3.
t	Represents an IP to ATM address/VCI entry. <code>aarsetup</code> adds these entries into the local table. Any <i>t</i> entries on the server must contain <i>ATM Address</i> and may also contain <i>VCI</i> if PVC communication between the server and client is desired. In addition, there are some cases when a <i>t</i> entry may be useful on an ARP client system. If a client wants to communicate with another system over PVCs, the PVC to be used is provided in a <i>t</i> entry containing <i>VCI</i> ; or if a client wishes to cache frequently used addresses to avoid frequent ARP requests, a <i>t</i> entry containing <i>ATM Address</i> may be provided. See TABLE 6-3.

TABLE 6-2 /etc/opt/SUNWconn/atm/aarconfig File Flag Descriptions (Continued)

Flag	Description
t cont.	Note: If your naming service (NIS+ or DNS) server is an ATM host, you must provide the hostname to IP address resolution for the hosts included in t entries, either by using the IP address in the <i>Hostname</i> field of the t entry, or by adding an entry to the local /etc/hosts file.
a	Represents an address that may have access to this host. If no a entries appear in the aarconfig file, access to the host is unrestricted. Including a entries allows access to be restricted to known hosts. As an alternative to listing individual addresses, the ATM address field may contain a prefix, followed by the wildcard \$anymacsel, which matches any 7-byte ESI/Selector combination following the given prefix. This allows access by any host connected to the switch that is specified by the given prefix. <i>Hostname</i> and <i>VCI</i> should not appear; <i>ATM Address</i> is required. See TABLE 6-3.
m	Notifies the system that the entire ATM address, including the network prefix, must be configured manually on this interface. If your interface is connected to a switch that does not support ILMI, you must include this option in your /etc/opt/SUNWconn/atm/aarconfig or /etc/opt/SUNWconn/atm/lanconfig file. Note that you may not use the variables \$myaddress, \$prefix, and \$localswitch_server (which use the switch prefix obtained from the switch via ILMI) if ILMI is not running.

Note – Although SunATM supports PVC connections to a server for ARP traffic, RFC 1577 does not specify this case. For interoperability with other implementations, connections to the server should use SVCs.

Note – For two hosts to communicate over PVCs, corresponding PVC connections must also be established in the ATM switch fabric.

TABLE 6-3 describes the required, optional, and illegal fields for each flag type. If a field is unused, it is represented by a hyphen.

TABLE 6-3 /etc/opt/SUNWconn/atm/aarconfig File Flag Requirements and Options

Interface *	Host	ATM Address	VCI	Flags	*
required	illegal	SVC only	illegal	l	local information
required	illegal	required	illegal	L	local information on server
required	illegal	required	illegal	a	access list entry

*or – Means one or the other required, but using both is also legal.

**xor – Means one or the other required, but using both is illegal.

TABLE 6-3 /etc/opt/SUNWconn/atm/aarconfig File Flag Requirements and Options

Interface *	Host	ATM Address	VCI	Flags	*
required	required	or*	or*	t	permanent table entry
required	illegal	xor**	xor**	s	server address/PVC
required	illegal	illegal	illegal	m	manual address registration

*or – Means one or the other required, but using both is also legal.
**xor – Means one or the other required, but using both is illegal.

Note – Group entries in the `aarconfig` file in a designated order: the local (*I* or *L*) entry first, followed by any other flags in any order. You only need to maintain the ordering within entries for each physical interface; for example, all of the `ba0` entries can appear first, and then all of the `ba1` entries, etc.

6.2.2 Using Variables in the /etc/opt/SUNWconn/atm/aarconfig File

Because the prefix portion of an ATM address specifies the ATM switch, a number of hosts specified in an `aarconfig` file can have ATM addresses that share the same prefix. To simplify setting up the `aarconfig` file, you can define variables that contain part of an ATM address.

A variable's name is an identifier consisting of a collection of no more than 32 letters, digits, and underscores. The value associated with the variable is denoted by a dollar sign followed immediately by the variable name.

Note – Use variables in the ATM address field only. They are not valid in any of the other fields in an entry.

You can use a colon to concatenate multiple variables to represent a single ATM address expression. Thus, if one variable, `v1`, is set to `11:22` and another variable, `v2`, is set to `33:44`, the sequence `$v1:$v2` represents `11:22:33:44`. You can include hexadecimal numbers with variables in the expression. The expression `45:$v1:$v2` would have the value `45:11:22:33:44`.

Use the following format to define variables in the `aarconfig` file:

```
set VARIABLE = EXPRESSION
```

where *VARIABLE* is the name of a variable and *EXPRESSION* is an expression concatenating one or two-digit hexadecimal numbers or the values of variables that have been previously defined. The equal sign is optional, but separate the variable and expression either by white space (spaces or tabs), an equal sign, or both.

Several predefined variables are built into the SunATM software. These variables are summarized in TABLE 6-4.

Note – You can not use the `$prefix` variable or any other variables that use it (including `$myaddress` and `$localswitch_server`) on interfaces that are not running ILMI.

TABLE 6-4 Predefined SunATM Variables

Variable	Description
<code>prefix</code>	The 13-byte prefix associated with the local switch.
<code>mac</code>	The 6-byte medium access control (MAC) address associated with the local host or interface.
<code>sel</code>	The default 1-byte selector for the local interface.
<code>macsel</code>	The concatenation of <code>\$mac:\$sel</code> .
<code>myaddress</code>	The concatenation of <code>\$prefix:\$mac:\$sel</code> , resulting in the default address for the local interface.
<code>anymac</code>	A wild card representing any 6-byte ESI. Should only be used in <i>a</i> entries.
<code>anymacsel</code>	A wild card representing any 7-byte ESI and Selector combination. Should only be used in <i>a</i> entries.
<code>?</code>	A wild card matching one or two hexadecimal digits within any colon-separated field. For example, <code>\$prefix:\$anymac:?</code> is equivalent to both <code>\$prefix:\$anymac:??</code> and <code>\$prefix:\$anymacsel</code> . However, it is <i>not</i> the same as <code>\$prefix:\$anymacsel:0?</code> , which requires that the first digit of the selector byte is a 0. This wild card should only be used in <i>a</i> entries.
<code>sunmacselN</code>	The concatenation of one of a series of reserved MAC addresses and <code>\$sel</code> to create a block of reserved ATM ARP server addresses. <i>N</i> should be a decimal number in the range 0-199.
<code>localswitch_server</code>	The concatenation of <code>\$prefix</code> , a unique reserved MAC address, and <code>\$sel</code> . When used as a server address, restricts server access to clients connected to the local switch only.

In most network configurations, the ATM address assigned to the local interface is `$myaddress`; using this variable in the `l` entry makes it possible to use identical `aarconfig` files on all Classical IP clients using a given server.

The `sunmacselN` variables can be used in conjunction with a prefix, as well as with known server addresses that are not bound to a particular system. As an example, consider the case where a server that supports 50 clients fails. If the ATM address of the server is specific to that particular server, you must change the `s` entry on all 50 clients to switch to a backup server. However, if the ATM address used for that server is `$prefix:$sunmacsel3`, this address is not only guaranteed to be unique, since it uses reserved medium access control (MAC) addresses, you can also simply assign that address to the backup server on the same switch by changing the `l` entry to an `s` entry on one system and bringing up a new server with no changes to the clients.

Note – The `sunmacselN` variables do not include a prefix since a client and server may be on different switches and thus have different local prefix values.

In the case of a single-switch network, you can use `localswitch_server` as a well-known server address that includes the prefix associated with the local switch and a MAC address. It will restrict server access to clients on the local switch and provide a unique ATM address among all ATM clients connected to that switch. Thus, any host with a network prefix other than that of the local switch will be refused a connection to the ARP server if the ARP server's address is `$localswitch_server`.

Several rules apply to the use of variables in the `aarconfig` file:

- Two variables cannot follow each other in an expression without an intervening colon. Thus `$v1:$v2` is legal while `$v1$v2` is not.
- Fields in each line in the `aarconfig` file are separated by white space. Therefore variables should not be separated from the rest of an ATM address with whitespace. For example, `$v1: $v2` is illegal.
- Once a variable is defined by a set command, it may not be redefined later in the same configuration file.
- The reserved variable names cannot be set. These names include `prefix`, `mac`, `sel`, `macsel`, `myaddress`, `anymac`, `anymacsel`, `sunmacselN` (where `N` is a number between 0 and 199), and `localswitch_server`.

Note – The ESI portion of `localswitch_server` and the `sunmacselN` variables is a reserved MAC address. The hexadecimal values of the reserved addresses are:

```
localswitch_server  08:00:20:75:48:08
sunmacselN base    08:00:20:75:48:10
```

To calculate the ESI portion for a `sunmacselN` address, simply add the value of *N* (converted to a hexadecimal number) to the `sunmacselN` base address. For example, the ESI portion of `sunmacsel20` would be `08:00:20:75:48:10 + 0x14 = 08:00:20:75:48:24`.

6.2.3 Sample Classical IP Configurations

The following examples demonstrate entries in the `/etc/opt/SUNWconn/atm/aarconfig` file for several typical network configurations.

Although some of the examples show only one sample `aarconfig` file, similarly configured files must appear on each system. Example 2 shows the files for each of the three systems in the configuration.

1. SVC-only: Clients use the default address and access to the ARP server is restricted to clients on the local switch only.

- a. The `/etc/opt/SUNWconn/atm/aarconfig` file on a client:

Interface	Host	ATM Address	VCI	Flag
ba0	-	\$myaddress	-	l
ba0	-	\$localswitch_server	-	s

- b. The `/etc/opt/SUNWconn/atm/aarconfig` file on the server:

Interface	Host	ATM Address	VCI	Flag
ba0	-	\$localswitch_server	-	L

2. PVC-only: *hosta* is connected to *hostb* and *hostc* over PVCs. There is no ARP server.

a. `/etc/opt/SUNWconn/atm/aarconfig` on *hosta*:

Interface	Host	ATM Address	VCI	Flag
ba0	-	-	-	l
ba0	hostb	-	100	t
ba0	hostc	-	101	t

b. on *hostb*:

Interface	Host	ATM Address	VCI	Flag
ba0	-	-	-	l
ba0	hosta	-	100	t
ba0	hostc	-	102	t

c. on *hostc*:

Interface	Host	ATM Address	VCI	Flag
ba0	-	-	-	l
ba0	hosta	-	101	t
ba0	hostb	-	102	t

3. SVC with no ARP server: *hosta* uses SVCs to connect to *hostb* and *hostc*. All hosts are connected to the same switch; there is no ARP server.

Interface	Host	ATM Address	VCI	Flag
ba0	-	\$myaddress	-	l
ba0	hostb	\$prefix:08:00:20:d5:08:a8:00	-	t
ba0	hostc	\$prefix:08:00:20:21:20:c3:00	-	t

4. PVC/SVC mix: *hosta* uses a SVC to connect to *hostb*, and a PVC to connect to *hostc*. *hostb* is not on the local switch; there is no ARP server.

Interface	Host	ATM Address	VCI	Flag
ba0	-	\$myaddress	-	l
ba0	hostb	45:00:00:00:00:00:00:00:0f:00:00:00:00:08:00:20:d5:08:a8:00	-	t
ba0	hostc	-	100	t

5. ARP server with Access Restrictions: Hosts are connected to an ATM ARP server that resolves addresses. Access is restricted to the local switch subnet and one additional switch subnet.

a. /etc/opt/SUNWconn/atm/aarconfig on *hosta*:

Interface	Host	ATM Address	VCI	Flag
ba0	-	\$myaddress	-	l
ba0	-	\$prefix:\$sunmacsel0	-	s

b. /etc/opt/SUNWconn/atm/aarconfig on server:

Interface	Host	ATM Address	VCI	Flag
ba0	-	\$prefix:\$sunmacsel0	-	L
ba0	-	\$prefix:\$anymacsel	-	a
ba0	-	45:00:00:00:00:00:00:00:0f:00:00:00:00:\$anymacsel	-	a

6. Manual address configuration: Hosts are connected to a switch that does not support ILMI.

a. /etc/opt/SUNWconn/atm/aarconfig on server:

```
set prfx = 45:00:00:00:00:00:00:00:0f:00:00:00:00
Interface Host ATM Address VCI Flag
ba0 - $prfx:$sunmacsel0 - L
ba0 - - - - m
```

b. /etc/opt/SUNWconn/atm/aarconfig on client:

```
set prfx = 45:00:00:00:00:00:00:00:0f:00:00:00:00
Interface Host ATM Address VCI Flag
ba0 - $prfx:$macsel - l
ba0 - $prfx:$sunmacsel0 - s
ba0 - - - - m
```

6.3 Configuring a LAN Emulation Interface

LAN Emulation, standardized by the ATM Forum's LAN Emulation 1.0 specification, is another way of providing TCP/IP and UDP/IP support over an ATM interface. Address resolution information is provided by a series of LAN

Emulation services. When a LAN Emulation interface is brought up, it must “join the LAN,” that is, it must register with these services. This process, and the address resolution process is described in Section 7.3 “LAN Emulation.”

Unlike Classical IP, the LAN Emulation protocol provides a broadcast service to the upper layer protocols. Therefore, the multicast and broadcast limitations described in Section 6.2 “Configuring a Classical Internet Protocol Interface,” do not affect LAN Emulation interfaces.

6.3.1 Editing the `/etc/opt/SUNWconn/atm/lanecfg` File

The `/etc/opt/SUNWconn/atm/lanecfg` file contains the required configuration information for each interface that uses LAN Emulation. One entry is required for each SunATM interface.

Each time you modify the `/etc/opt/SUNWconn/atm/lanecfg` file, run the LAN Emulation setup program (`lanesetup`). `lanesetup` is in the `/etc/opt/SUNWconn/atm/bin` directory.

Each `/etc/opt/SUNWconn/atm/lanecfg` entry follows this format:

Interface	MAC Address/ ELAN Name	ATM Address	VCI	Flag
-----------	---------------------------	-------------	-----	------

These entry fields are described in TABLE 6-5.

TABLE 6-5 `/etc/opt/SUNWconn/atm/lanecfg` Entry Descriptions

Field	Description
Interface	Refers to the LAN Emulation interface, <code>laneN</code> .
MAC Address/ELAN Name	This field is for the 6-byte MAC address of the interface, or, in the case of an <code>n</code> entry, the name of the emulated LAN to join.

TABLE 6-5 /etc/opt/SUNWconn/atm/laneconfig Entry Descriptions

Field	Description
ATM Address	This field is for the 20-byte ATM address. The <code>\$myaddress</code> variable assigns the local switch prefix, local MAC address, and default selector.
VCI	The VCI field is a positive decimal integer identifying a Permanent Virtual Circuit. Place a dash in this field if VCI is not used.
Flag	This field identifies the type of entry. For a complete description of the <code>laneconfig</code> flags, see TABLE 6-6.

TABLE 6-6 describes the flags used in the `/etc/opt/SUNWconn/atm/laneconfig` file.

TABLE 6-6 /etc/opt/SUNWconn/atm/laneconfig Flag Descriptions

Flag	Description
l	This flag designates a local address entry. There must be an <code>l</code> entry for each interface running LAN Emulation. The interface and the ATM address must be included; the MAC address may be included (default is the MAC address assigned to the board).
t	This flag designates a table entry for the local MAC-ATM address resolution table. If you wish to avoid the address resolution process for a frequently accessed system, for instance, you may include a <code>t</code> entry for that system; you may also create PVCs with a <code>t</code> entry. The interface, MAC address, and either ATM address or VCI (for SVC or PVC connection, respectively) must be included.
n	This flag allows you to specify the name of an emulated LAN to join. By default, the SunATM implementation will use the name provided by the LECS. If you wish to specify a different name, or if your LECS requires that a user include a name in its requests, a name may be provided with this flag. Interface is required; the name should be entered in the second field.
M	Notifies the system that a larger MTU size will be used in the ELAN which this lane instance will join. The default MTU size is 1500 bytes. With the <code>M</code> flag, you can set the MTU size to be either 4 Kbytes (4528 bytes) or 9 Kbytes (9218 bytes).
a	Represents an address that may have access to this host. If no <code>a</code> entries appear in the <code>laneconfig</code> file, access to the host is unrestricted. Including <code>a</code> entries allows access to be restricted to known hosts. As an alternative to listing individual addresses, the ATM address field may contain a prefix, followed by the wildcard <code>\$anymacsel</code> , which matches any 7-byte ESI/Selector combination following the given prefix. This allows access by any host connected to the switch specified by the given prefix. <i>Mac Address</i> and <i>VCI</i> should not appear; <i>ATM Address</i> is required. See TABLE 6-7.

TABLE 6-6 /etc/opt/SUNWconn/atm/laneconfig Flag Descriptions (Continued)

Flag	Description
c	This flag allows an alternate LECS address to be specified. By default, the SunATM software uses the well-known address specified in the LAN Emulation standard. If, however, your LECS has a different address, or you wish to connect to the LECS over a PVC, you may provide the alternate ATM address or VCI in a c entry. If you wish to make a PVC connection, the VCI must be 17, as required by the LAN Emulation standard. The interface and ATM address or VCI must be included.
s	This flag specifies the LES address or VCI, and instructs the system to contact the LES directly, and to use default subnet configuration information. This flag should be used if your subnet does not have an LECS. Without this entry, the system first connects to the LECS, which provides the LES address and configuration information.
m	Notifies the system that the entire ATM address, including the network prefix, must be configured manually on this interface. If your interface is connected to a switch that does not support ILMI, you must include this option in your /etc/opt/SUNWconn/atm/aarconfig or /etc/opt/SUNWconn/atm/laneconfig file. Note that the variables \$myaddress, \$prefix and \$localswitch_server (which use the switch prefix obtained from the switch via ILMI) may not be used if ILMI is not running.

TABLE 6-7 describes the required, optional, and illegal fields for each flag type.

TABLE 6-7 laneconfig Flag Requirements and Options

Interface	MAC Address/ELAN Name	ATM Address	VCI	Flag
required	optional	required	illegal	l
required	required	xor ¹	xor ¹	t
required	Emulated LAN name	illegal	illegal	n
required	MTU Size in bytes	illegal	illegal	M
required	illegal	required	illegal	a
required	illegal	xor ¹	xor	c
required	illegal	xor ¹	xor	s
required	illegal	illegal	illegal	m

¹xor means that you can use either the ATM Address field or the VCI field, but not both.

Note – Designate unused fields in the /etc/opt/SUNWconn/atm/laneconfig file with a dash.

6.3.2 Using Variables in the `/etc/opt/SUNWconn/atm/laneconfig` File

You can use some of the predefined variables from `/etc/opt/SUNWconn/atm/aarconfig` file in the `/etc/opt/SUNWconn/atm/laneconfig`. These variables are listed in TABLE 6-8. For a complete description of how to use these variables, see Section 6.2.2 “Using Variables in the `/etc/opt/SUNWconn/atm/aarconfig` File.”

Note – Using predefined or user-defined variables in the MAC address field of local ('l') entries is not supported. Variables may be used in the MAC address field of other entry types, such as in table ('t') entries.

Note – You can not use the `$prefix` variable or any other variables that use it (including `$myaddress`), on interfaces that are not running ILMI.

TABLE 6-8 Predefined SunATM Variables

Variable	Description
<code>prefix</code>	The 13-byte prefix associated with the local switch.
<code>mac</code>	The 6-byte MAC address associated with the local host or interface.
<code>sel</code>	The default 1-byte selector for the local interface.
<code>macsel</code>	The concatenation of <code>\$mac:\$sel</code> .
<code>myaddress</code>	The concatenation of <code>\$prefix:\$mac:\$sel</code> , resulting in the default address for the local interface.
<code>anymac</code>	A wild card representing any 6-byte ESI. Should only be used in <code>a</code> entries.
<code>anymacsel</code>	A wild card representing any 7-byte ESI and Selector combination. Should only be used in <code>a</code> entries.
<code>?</code>	A wild card matching one or two hexadecimal digits within any colon-separated field. For example, <code>\$prefix:\$anymac:?</code> is equivalent to both <code>\$prefix:\$anymac:??</code> and <code>\$prefix:\$anymacsel</code> . However, it is <i>not</i> the same as <code>\$prefix:\$anymacsel:0?</code> , which requires that the first digit of the selector byte is a 0. This wild card should only be used in <code>a</code> entries.

6.3.3 Sample LAN Emulation Configurations

The following examples demonstrate entries in the `/etc/opt/SUNWconn/atm/lanecfg` file for several common configurations.

Although some of the examples show only one sample `lanecfg` file, similarly configured files must appear on each LAN Emulation client.

1. Basic LAN Emulation client. The ATM and MAC address of a frequently used server is provided. The LECS provides the name of the Emulated LAN.

```
set srvr_mac = 08:00:20:01:02:03
```

Interface	MAC_Address/ ELAN Name	ATM_Address	VCI	Flag
lane0	-	\$myaddress	-	l
lane0	\$srvr_mac	\$prefix:\$srvr_mac:00	-	t

2. LAN Emulation client. The LECS requires that the client send the Emulated LAN name in its messages.

Interface	MAC_Address/ ELAN Name	ATM_Address	VCI	Flag
lane0	-	\$myaddress	-	l
lane0	elan1	-	-	n

6.4 Supporting Logical Interfaces

The SunATM software supports logical interfaces in the LAN Emulation environment. Logical interfaces allow you to assign multiple IP addresses to a single Emulated LAN interface. A logical interface name consists of three parts: the device name (in the case of SunATM LAN Emulation, `lane`); the major number, which corresponds to the lane instance number; and the minor number, which distinguishes the logical interfaces on a single physical interface. The format of a LAN Emulation logical interface name is `laneN:X`, where `N` is the major number and `X` is the minor number.

Each logical interface will be associated with a unique IP hostname and address. All logical interfaces on a given physical interface will be associated with the same ATM and MAC addresses. Configure logical interfaces by placing multiple entries for a given interface in the `/etc/opt/SUNWconn/atm/atmconfig` file.

Consider the following rules when you use logical interfaces with the SunATM software:

- Only one signalling protocol (UNI 3.0, 3.1, or 4.0) is supported per interface, and must appear in the first entry for that interface.
- Only one Classical IP hostname may be assigned to an interface; it can appear in any entry in any order in `/etc/opt/SUNWconn/atm/atmconfig`.
- The first `laneN` entry on an interface must be for `laneN:0`, or simply `laneN`. `laneN` and `laneN:0` are identical and interchangeable.
- IP limits the number of logical interfaces on a physical interface to 256 (the minor number *X* must be in the range 0 - 255) in Solaris 2.5.1, and to 8194 (the minor number *X* must be in the range 0 - 8193) in Solaris 2.6 and Solaris 7.

The following examples show the `atmconfig` and `laneconfig` files and the `ifconfig -a` output for a system with one physical interface, `ba0`. That interface runs both Classical IP and LAN Emulation under UNI 3.1, and has 4 different IP addresses. Configure the hostnames, `cip0`, `atm0`, `atm1`, and `atm2`, appropriately in `/etc/hosts`.

The example `/etc/opt/SUNWconn/atm/atmconfig` file:

Interface	UNI	CIP Hostname	LANE Instance	LANE Hostname
ba0	3.1	cip0	0	atm0
ba0	-	-	0:1	atm1
ba0	-	-	0:2	atm2

The corresponding example `/etc/opt/SUNWconn/atm/laneconfig` file:

Interface	MAC Address/ ELAN Name	ATM Address	VCI	Flag
lane0	-	\$myaddress	-	1

The resulting `ifconfig -a` output:

```

atmsw-12# ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000
ba0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu
9180
    inet 129.144.234.12 netmask fffffff0 broadcast
129.144.234.255
    ether 8:0:20:84:e5:31
lane0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu
1500
    inet 129.144.161.12 netmask fffffff0 broadcast
129.144.161.255
    ether 8:0:20:84:e5:31

atmsw-12# atmifconfig ba0 unplumb

atmsw-12# ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000
lane0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu
1500
    inet 129.144.161.12 netmask fffffff0 broadcast
129.144.161.255
    ether 8:0:20:84:e5:31

atmsw-12# atmifconfig lane0 unplumb
NOTICE: lane0: leaving ELAN

atmsw-12# ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000

atmsw-12# atmifconfig ba0 plumb-all
q93b on ba0: Data Link Up
Configuring ATM interfaces: ba0
Configuring ATM LAN Emulation interfaces: lane0

```

6.5 Supporting Multiple Emulated LANs on a Single Interface

The SunATM software allows a single ATM interface to join up to sixteen emulated local area networks (ELANs), provided this is allowed by the switch and LAN Emulation (LANE) services. Each ELAN joined will be represented by a unique lane instance (e.g. lane0 or lane1).

Note – A requirement for supporting this feature is that the adapter card be assigned multiple MAC addresses, which is supported in the SunATM/S 2.1 and SunATM/P 3.0 adapters. This feature *does not* work with the older SunATM/S 2.0 adapters. You can find the number of MAC addresses assigned to your SunATM adapter by using the `atmgetmac(1M)` command with the `count` option.

Configure multiple ELANs by placing multiple entries in the `/etc/opt/SUNWconn/atm/atmconfig` and `/etc/opt/SUNWconn/atm/laneconfig` files. Each lane instance will have a unique hostname and IP address, ATM address, and MAC address associated with it. In addition, assign an ELAN name to the instance if any ELAN other than the default is to be joined. Provide this information, with the exception of the MAC address, which is retrieved from the board itself, in the `/etc/opt/SUNWconn/atm/atmconfig` and `/etc/opt/SUNWconn/atm/laneconfig` configuration files.

Note – Only one signalling protocol (e.g. UNI 3.0 or 3.1) and one Classical IP instance are supported per physical interface. Specify the UNI version in the first `/etc/opt/SUNWconn/atm/atmconfig` entry for a given interface; specify the Classical IP instance may be specified in any entry.

The following example shows the `/etc/opt/SUNWconn/atm/atmconfig` and `/etc/opt/SUNWconn/atm/laneconfig` files and the `ifconfig -a` output for a system with one SunATM interface, `ba0`. The interface uses UNI 3.0 for signalling, and does not run Classical IP. It joins 4 emulated LANs: the default, `elan1`, `elan2`, and `elan3`.

The example `/etc/opt/SUNWconn/atm/atmconfig` file:

Interface	UNI	CIP	Hostname	LANE	Instance	LANE	Hostname
ba0	3.0	-		0			atm0
ba0	-	-		1			atm1
ba0	-	-		2			atm2
ba0	-	-		3			atm3

The corresponding example `/etc/opt/SUNWconn/atm/laneconfig` file:

Interface	MAC Address/ ELAN Name	ATM Address	VCI	Flag
lane0	-	\$myaddress	-	l
lane1	-	\$myaddress	-	l
lane1	elan1	-	-	n
lane2	-	\$myaddress	-	l
lane2	elan2	-	-	n
lane3	-	\$myaddress	-	l
lane3	elan3	-	-	n

The resulting `ifconfig -a` output:

```
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000
lane0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 192.29.240.36 netmask ffffffff broadcast 192.29.240.255
    ether 8:0:20:7a:37:af
lane1: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 192.29.241.36 netmask ffffffff broadcast 192.29.241.255
    ether 8:0:20:7a:37:b0
lane2: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 192.29.242.36 netmask ffffffff broadcast 192.29.242.255
    ether 8:0:20:7a:37:b1
lane3: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 192.29.243.36 netmask ffffffff broadcast 192.29.243.255
    ether 8:0:20:7a:37:b2
```

6.6 Plumbing and Unplumbing Individual ATM Interfaces

The `atmifconfig` utility allows interfaces to be added, modified, or removed without rebooting the system by setting up and tearing down the streams necessary to use an atm device. This setting up and tearing down of streams is referred to as “plumbing” and “unplumbing”.

The two required parameters for the utility are an interface name and plumbing instructions. If an interface is being set up for native ATM applications (no TCP/IP) or for Classical IP, the interface name should have the format `baN`, where `N` is the instance number. For LAN Emulation instances, the interface should have the format `laneN`, where `N` is the LAN Emulation instance number.

The second required parameter is the plumbing instruction. There are four possible values: `plumb`, `unplumb`, `plumb-all`, and `unplumb-all`. The `plumb` and `unplumb` options will set up or tear down the specified interface. The `plumb-all` and `unplumb-all` options will set up or tear down all ATM instances on the specified physical interface. In this case, the interface parameter must be a `baN` value, since this specifies the physical interface name. This will set up or tear down all the ATM instances on this physical interface, including up to 16 LAN Emulation instances and the Classical IP instance.

A summary of the parameter options for the `atmifconfig` utility is provided in TABLE 6-9.

TABLE 6-9 Parameter Options for `atmifconfig`

Utility	Interface Name	Plumb Instruction
<code>atmifconfig</code>	<code>baN</code>	<code>plumb</code>
<code>atmifconfig</code>	<code>baN</code>	<code>unplumb</code>
<code>atmifconfig</code>	<code>laneN</code>	<code>plumb</code>
<code>atmifconfig</code>	<code>laneN</code>	<code>unplumb</code>
<code>atmifconfig</code>	<code>baN</code>	<code>plumb-all</code>
<code>atmifconfig</code>	<code>baN</code>	<code>unplumb-all</code>

The following examples show the use of `atmifconfig` and the output of `ifconfig -a` before and after the utility is run.

```
atmsw-12# ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000
ba0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu
9180
    inet 129.144.234.12 netmask ffffffff broadcast
129.144.234.255
    ether 8:0:20:84:e5:31
lane0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu
1500
    inet 129.144.161.12 netmask ffffffff broadcast
129.144.161.255
    ether 8:0:20:84:e5:31

atmsw-12# atmifconfig ba0 unplumb

atmsw-12# ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000
lane0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu
1500
    inet 129.144.161.12 netmask ffffffff broadcast
129.144.161.255
    ether 8:0:20:84:e5:31

atmsw-12# atmifconfig lane0 unplumb
NOTICE: lane0: leaving ELAN

atmsw-12# ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232
    inet 127.0.0.1 netmask ff000000

atmsw-12# atmifconfig ba0 plumb-all
q93b on ba0: Data Link Up
Configuring ATM interfaces: ba0
Configuring ATM LAN Emulation interfaces: lane0
```


Classical IP and LAN Emulation Protocols

ATM is a connection-oriented network protocol, which means that a connection must be established between two communicating entities before data transfer can begin. IP is inherently connectionless. The implementation on the host must therefore reconcile the differences in these two paradigms.

There are two standard ways of doing this: Classical IP, standardized in RFC 1577, and LAN Emulation, standardized in the LAN Emulation 1.0 specification from the ATM Forum. The SunATM architecture supports both of these methods. This chapter discusses some of the key ideas of these two methods.

Both methods allow IP to run transparently over the ATM interface. Thus IP itself sees the ATM interface just as it sees any traditional network interface. Every SunATM interface has a subnet IP address. As an ATM interface starts up, appropriate modules and drivers are plumbed. All the TCP/IP and UDP/IP applications run without modifications over these modules, and all the utilities associated with the network interfaces also run without modification and display similar results (for example, `netstat` and `ifconfig` utilities), with one exception. Because of the different plumbing of the ATM modules, the `plumb` and `unplumb` options of `ifconfig` will not work on ATM interfaces. The `atmifconfig(1M)` command may be used to `plumb` and `unplumb` ATM interfaces. IP treats the ATM interface as a subnet, choosing the interface used to send a packet out based on the IP address of the destination and on the IP address and netmask of the interface itself.

The transparency to IP is enabled in different ways in Classical IP and LAN Emulation. Those differences will be discussed in later sections of this chapter.

SunATM signalling conforms to the UNI specification of the ATM Forum. Versions 3.0, 3.1, and 4.0 of that specification are supported. This signalling, called Q.2931, runs on top of QSAAL and uses VC 5 for signalling as specified in the Forum specification.

7.1 ATM Addresses and Address Registration

UNI signalling uses ATM addresses for signalling. Every ATM interface has an ATM address in addition to its IP address.

ATM addresses, like Network Service Access Point (NSAP) addresses, are 20 octets long. The End System Identifier (ESI) field within the ATM address is a unique 6 octet value; this can be the IEEE hardware MAC address conventionally associated with every network interface. The Selector field is one octet long. The 13 octets that make up the rest of the ATM address are called the Network Prefix, and are derived from the ATM switch fabric to which the interface is connected. Every ATM switch fabric is configured with a 13 octet prefix.

On a SunATM host, the prefix associated with the local switch fabric is represented by the `$prefix` variable. Its value is obtained by the system at configuration time.

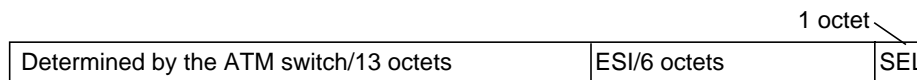


FIGURE 7-1 ATM Address Fields

The UNI specification specifies the Interim Local Management Interface (ILMI) service interface for a client to learn and register its ATM address. The ILMI service interface is based on the use of SNMP over AAL5. In the SunATM software package, ILMI service is provided by an address registration daemon, `ilmid`.

7.1.1 ATM Address Registration Daemon (`ilmid`)

Address registration with a switch is controlled by `ilmid`. When an ATM interface is brought up at boot time, `ilmid` is also started. `ilmid` then begins an exchange of messages with the switch: relaying local address information (the 7 octet ESI and selector) to the switch and receiving the 13 octet network prefix information from the switch.

The default local address that is registered with the switch at boot time consists of the network prefix provided by the switch, the MAC address assigned to the local interface, and the default selector for that interface (usually 0). Additional addresses are registered in two different ways. `aarsetup(1M)` and `lanesetup(1M)` register additional local addresses that may appear in `aarconfig(4)` and `laneconfig(4)`, respectively. You can also use `atmreg(1M)` to register addresses, de-register addresses, and check the status of any address.

7.2 Classical Internet Protocol

For ATM to work transparently under IP, an IP address must be resolved to an ATM address and a connection to that destination must be established. Classical IP does this via a database of IP/ATM address pairs that is either provided by an ATM ARP server that is accessible to all hosts on the subnet, or is maintained locally in each host.

7.2.1 ATM Address Resolution

Traditional TCP/IP and UDP/IP applications use IP addresses for communicating to a destination. For these applications to run like traditional applications, IP addresses need to be resolved into ATM addresses. The ATM address then signals to establish an ATM connection to the destination. An ATM connection in turn is represented by a VPI/VCI. The host must use this returned VPI/VCI to send packets to the destination that represents the ATM connection.

ATM address resolution, also called ATM ARP, follows RFC 1577, the classic draft that describes the ATM ARP process.

RFC 1577 assumes the existence of an ATM ARP server on every subnet. Every client on the subnet communicates with the ATM ARP server to derive the destination's ATM address from its IP address. The ATM ARP server holds the IP-to-ATM address information for all hosts in the ATM subnet. It is likely that initial ATM configurations will not rely on dynamic ATM address resolution because it requires the presence of an ATM ARP server on every subnet. Also, there are no specified standards for providing redundant ATM ARP servers for a subnet. As specified, the ATM ARP server would constitute a single point of failure in the system. From a practical standpoint, however, early configurations can use an IP-to-ATM address database in every system, thus avoiding the IP-to-ATM address resolution step altogether.

The RFC requires a router for passing data between subnets. SunATM software provides ATM utilities that allow configurations to specify IP-to-ATM addresses in `/etc/opt/SUNWconn/atm/aarconfig` files. The `aarsetup` program uses the information in `/etc/opt/SUNWconn/atm/aarconfig` to create IP-to-ATM address resolution tables. Dynamic entries into a server's resolution table are also supported.

TABLE 6-1 on page 6-2 shows the format of the `/etc/opt/SUNWconn/atm/aarconfig` file for specifying the IP-to-ATM address. It is important for the file to be consistent on all systems in the subnet. See Section 6.2.1 "Editing the `/etc/opt/SUNWconn/atm/aarconfig` File" on page 6-4.

7.2.2 ATM ARP Address Resolution Tables

Depending on the `aarconfig` file, the Classical IP software runs as either a server or a client. As a server, the Classical IP software handles ATM ARP requests originating from its clients. An ATM server has to be configured for each subnet. The ATM ARP server code conforms to RFC 1577: clients send ATM ARP requests to the server to resolve a destination IP address to an ATM address. The server then replies to ATM ARP requests by sending an ATM ARP response. If the server does not have the IP-to-ATM address entry, then it replies with NAK.

All the IP-to-ATM address entries specified in the `/etc/opt/SUNWconn/atm/aarconfig` file are entered into a kernel resident table by the ATM ARP setup program, `aarsetup`. Additional entries in the kernel table are added dynamically using the inverse ARP process. When a client connects to the server, the server sends an inverse ARP request back to the client to obtain the client's IP address. When a response is received, an entry is created for that client. The Classical IP software also responds to client ARP requests. The software looks up a kernel IP-to-ATM address entry and responds to an ATM ARP request with either an ATM ARP reply or ATM ARP NAK (if there is no entry in the table). Note that an ATM ARP client uses the virtual channel (VC) specified in the `/etc/opt/SUNWconn/atm/aarconfig` file to communicate with the server; or, if an ATM address is specified, it establishes a switched virtual circuit (SVC) connection to communicate with the server.

While dynamic entries in the ARP server's table make network administration less complex, they also create a security problem. Any host can register with the ARP server and therefore gain access to the subnet. To resolve this issue, you can provide a list of hosts or networks with `a` entries in the server's `/etc/opt/SUNWconn/atm/aarconfig` file. If no `a` entries appear, any host can connect to the server. If any `a` entries exist, only those hosts whose addresses match those specified will be allowed to connect.

Although the `a` entry requires a complete ATM address, you can reference multiple addresses in a single entry using the provided wildcards. See Section 6.2.2 "Using Variables in the `/etc/opt/SUNWconn/atm/aarconfig` File" on page 6-7, for more information about this feature.

The advantage of having an ATM ARP server in the subnet is that it represents a known source for all address resolutions. It is the only host that a client must know about to have IP addresses resolved to ATM connections, and it allows for access control in the ATM network.

When the `/etc/opt/SUNWconn/atm/aarconfig` file has been modified on a system, it is necessary to rerun `aarsetup`.

Note – For better caching, all clients have the option of adding to their configuration file the IP-to-ATM address information for other clients. This can benefit clients that communicate frequently because it eliminates having to go through the ATM ARP server for IP-to-ATM address resolution.

If a host has multiple SunATM cards, the host can be a server for one IP subnet and a client for another. This is handled transparently by `aarsetup`.

7.3 LAN Emulation

As described in previous sections, Classical IP provides its own (IP-to-ATM) address resolution mechanism that corresponds to and replaces ARP, thus allowing IP-based applications to run transparently over ATM. A shortcoming of Classical IP, and a primary reason it must replace the traditional ARP, is that it does not support broadcast messages.

Because ATM is a connection-oriented protocol (unlike ethernet), implementing broadcast is much more difficult. The only host that receives a message is the host to which the message is addressed, and a call must be established to that host before the message can be sent.

Local Area Network (LAN) Emulation, as standardized by the ATM Forum, provides mechanisms to send broadcast messages in an ATM environment. Given this capability, LAN Emulation is also able to work transparently with ARP, as well as IP. IP and ARP send broadcast messages over the ATM interface, and thus resolve IP addresses to MAC addresses; messages are then sent to the LAN Emulation driver, which has its own address resolution protocol (similar to that of Classical IP) to resolve the medium access control (MAC) address to an ATM address and connection.

The SunATM software implements the client side of the LAN Emulation standard. To use LAN Emulation in an environment, several LAN Emulation services must also exist in the emulated LAN. These services, called the LAN Emulation Configuration Server (LECS), the LAN Emulation Server (LES), and the Broadcast and Unknown Address Server (BUS), are generally provided in an ATM switch. The following sections provide an overview of the functions of these servers.

7.3.1 LAN Emulation Servers

There are three types of LAN Emulation servers. Each type is briefly described in this section.

7.3.1.1 LAN Emulation Configuration Server

This server is first contacted first by a host interface when the host is brought up on the emulated LAN. Its address is generally a well-known address specified by the LAN Emulation standard that is coded into the host software; thus establishing this connection requires no input from you. When contacted by a host wishing to join its emulated LAN, the LECS replies with configuration parameters for the emulated LAN, as well as the address of the LES.

7.3.1.2 LAN Emulation Server

The second step in joining an emulated LAN is to make a connection to the LAN Emulation Server. After receiving the LES address from the LECS, a host will establish a connection to the LES. The LES may add the host to a point-to-multipoint call that is maintained by the LES with connections to every host in the emulated LAN. This point-to-multipoint connection, if created by the LES, is used to send control information to each host on the emulated LAN.

The LES acts as the ATM ARP server. Since IP and ARP work with MAC addresses, an additional address resolution step is required to convert a MAC address to the corresponding ATM address, which is used to make a connection to the target host; this resolution step is provided by the LES.

7.3.1.3 Broadcast and Unknown Address Server

The final step in joining an emulated LAN is to make a connection to the BUS. The ATM address of the BUS is obtained by sending a LAN Emulation ARP request to the LES for the broadcast address. Once established, this connection is used to send broadcast messages to the BUS, which will add the client to a point-to-multipoint call including all hosts on the emulated LAN. Thus when a broadcast message (such as an IP ARP request) is received by the LAN Emulation host from its upper layers, it sends that message to the BUS, which forwards it to all hosts in the emulated LAN. Just as in the case of ethernet, the correct host responds to the sender, and thus the IP address is resolved to a MAC address.

7.3.2 Resolving an IP Address to an ATM Connection

The entire process from the time IP sends a message addressed to an IP address to the arrival of that message at the appropriate destination was hinted at in the previous descriptions of the LAN Emulation servers. To demonstrate how those pieces work together during the actual transmission of a message, the process is described below. This description assumes that none of the needed addresses have

been previously resolved and cached. The two hosts involved are referred to as the source (the system that wishes to send a message) and the target (the system to which the message is addressed).

1. IP has a message to transmit and only knows the IP address of the target system. IP first sends a message to ARP, to resolve the IP address to a MAC address.
2. ARP creates a broadcast request for the MAC address corresponding to the given IP address, which it sends to the LAN Emulation driver.
3. The LAN Emulation driver recognizes that this message has a broadcast address, and sends it to the BUS, which forwards the message to every host on the emulated LAN.
4. The message is received on each host, and sent up to ARP by the LAN Emulation driver.
5. On the target, ARP recognizes the IP address as its own and sends a response with its MAC address (addressed to the source's MAC address) down to the LAN Emulation driver.
6. The LAN Emulation driver sends an LE ARP request to the LES to resolve the source's MAC address to its ATM address.
7. The LES responds with the requested ATM address, and the target host sets up an ATM connection to the source host, over which it sends the IP ARP response.
8. The LAN Emulation driver on the source receives the IP ARP response message and sends it up to ARP. ARP then inserts the MAC address into the original message and sends it back down to the LAN Emulation driver.
9. The LAN Emulation driver then must send an LE ARP request to the LES to resolve the MAC address in the message from ARP to an ATM address. When it receives an LE ARP response, it then sees that it has a connection to that address (established by the target to return the IP ARP response) and sends the original IP message to the target over that connection.

7.3.3 LAN Emulation Connections

There are several connections established at all times when a host is a member of an emulated LAN. The following table outlines the various LAN Emulation-related connections that should be on a LAN Emulation client (LEC).

Note – Use the command `qccstat(1M)` to view all existing connections for a given interface.

TABLE 7-1 LAN Emulation Connections

VCC	Endpoints	Comments
Configuration Direct	LEC → LECS	This connection is not required to remain open after the initial join of the emulated LAN, and thus may time out after a host has joined the LAN.
Control Direct	LEC → LES	Point-to-point connection over which the host may send LE ARP requests and receive responses from the LES.
Control Distribute	LES → LEC	Point-to-multipoint connection over which the LES may send administrative information to all hosts. Hosts may not send on this connection.
Multicast Send	LEC → BUS	Point-to-point connection over which the host may send broadcast messages to the BUS. A limited amount of data is also allowed on this connection.
Multicast Forward	BUS → LEC	Point-to-multipoint connection over which the BUS sends broadcast messages. Hosts may not send on this connection.

7.4 ATM and SNMP

Two of the ATM standards supported by the SunATM software (the User Network Interface (UNI) and LAN Emulation (LANE) specifications) include definitions of SNMP-style Management Information Bases (MIBs) relevant to those standards. These MIBs are referred to as the ATM Forum (ATMF) and LAN Emulation (LANE) MIBs, respectively.

The ATM SNMP daemon (`atmsnmpd`) handles requests for information in both MIBs, as well as the system MIBs, from SNMP-based network management systems (such as the SunNet Manager program), and from `ilmid`, when it is required, for SNMP requests coming from the switch.

`atmsnmpd` can be used as a forwarding agent. If you configure it as a forwarding agent, `atmsnmpd` will forward SNMP requests for unknown MIBs to the port specified with the `forward` option, `-f`. This allows a system to have two SNMP

agents respond to requests received over the SNMP port. FIGURE 7-2 illustrates the required configuration. To set up this example configuration, `atmsnmpd` must be started with the parameter `-f 1000` and `other_snmpd` must be started so that it listens on port 1000.

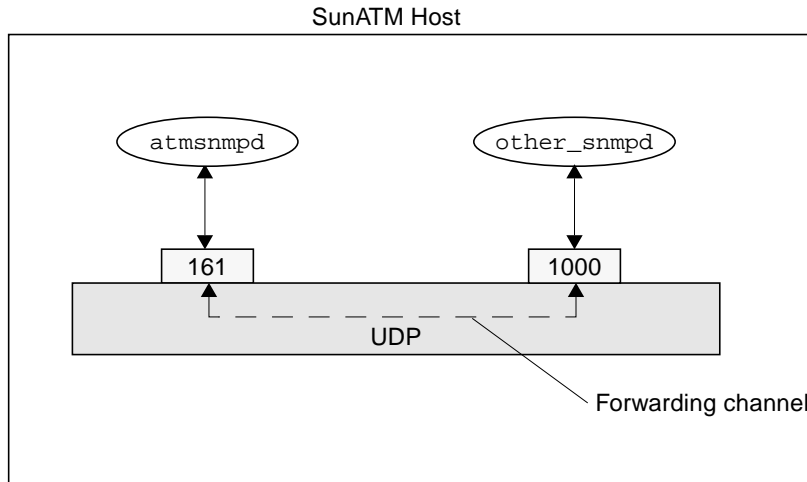


FIGURE 7-2 Using `atmsnmpd` as a Forwarding Agent

Note – If you do not specify a forwarding port for unknown requests, `atmsnmpd` will respond with a “No Such Name” error to requests for MIBs that it does not support. If you do specify a forwarding port, `atmsnmpd` instead forwards the request to that port. Responses received from the agent running on the forwarding port are sent to the requesting SNMP manager with no modification. If the agent does not respond, then `atmsnmpd` does not send a response back.

7.4.1 SNMP and Solaris

The ATM SNMP agent in SunATM 4.0 supports a new framework for SNMP agents; however, this means that its set up depends on the version of the Solaris operating environment in which it is running. This section discusses those differences, which for the most part are transparent to the user.

7.4.1.1 Releases of Solaris Prior to 2.6

SNMP agents by default bind to UDP port 161 to send and receive information. Since the SNMP protocol is set up with one common UDP port number, only one SNMP agent, such as `atmsnmpd`, can run on a system at a time. Many SNMP agent daemons, including `atmsnmpd`, allow an alternate port number to be specified, but this will limit the accessibility of that agent to network managers such as the SunNet Manager program. Depending on your requirements, you might want to run `atmsnmpd` on an alternate port. Change the UDP port from the default 161 if using this program. Use the `atmadmin` program to configure `atmsnmpd`.

Note – Even if the system is not configured to be an SNMP agent, `atmsnmpd` is started so that it can still be used by `ilmid`. It will be started with the `-n` option, which indicates that it should not bind to any UDP port.

7.4.1.2 Solaris 2.6 and Solaris 7 and Compatible Software

The Solaris 2.6 and 7 operating environments include a bundled SNMP agent that is designed to run as a *master* agent, binding by default to UDP port 161. Configure other agents to listen to other UDP ports and act as subagents, then configure the master agent to forward particular requests to those subagents. This framework provides a single agent at port 161 with the combined capabilities of the master agent and all the additional subagents.

The SunATM 4.0 software has been designed to take advantage of this framework if it is installed on a system running Solaris 2.6 or 7. The files necessary for the ATM SNMP agent to be recognized by the master agent (`atm.reg` and `atm.rsrc`) are copied under `/etc/snmp/conf` by the `S00sunatm` startup script if it detects that the system is running Solaris 2.6 or 7. SNMP requests pertaining to the ATM Forum subtree (`atmForum`) are forwarded to the `atmsnmpd` from the master agent. In addition, `atmsnmpd` binds, by default, to port 1000, rather than 161, under Solaris 2.6 or 7. The UDP port may still be changed using `atmadmin`, but the default will be 1000 in Solaris 2.6 or 7.

Appendix E “Managing SunATM Interfaces with SNMP,” provides more information about using `atmsnmpd` to manage and monitor the SunATM interfaces with a network manager such as the SunNet Manager program.

Declarations of Conformity

The following pages certify that the SunATM adapters conform to the stated requirements.

Declaration of Conformity

Compliance ID: SBUS-155MMF

Product Name: Sun ATM155MMF SBUS Adapter X1060A

This product has been tested and complies with the following EMC and Safety standards:

EMC

EC – Europe

This equipment complies with the following requirements of the EMC Directive 89/336/EEC:

EN55022 / CISPR22 (1985)	Class A
EN50082-1 IEC801-2 (1991)	4 kV (direct), 8 kV (air)
IEC801-3 (1984)	3 V/m
IEC801-4 (1988)	1.0 kV power lines, 0.5 kV signal lines
EN61000-3-2/IEC1000-3-2 (1994)	Pass

Safety

This equipment complies with the following requirements of the Low Voltage Directive 73/23/EEC:

- EN60950/IEC950

Supplementary Information

This product was tested and complies with all the requirements for the CE Mark when connected to a Sun workstation or server.

/ S /

Dennis P. Symanski
Manager, Product Compliance

Date

Sun Microsystems Computer Company
901 San Antonio Road
Palo Alto, CA 94303-4900, USA
Tel: 415-786-3255
Fax: 415-786-3723

/ S /

John Shades
Manager, Quality Assurance

Date

Sun Microsystems Limited
Springfield, Linlithgow
West Lothian, EH49 7LR
Scotland, United Kingdom
Tel: 0506 670000
Fax: 0506 760011

Declaration of Conformity

Compliance ID: SBUS-155UTP

Product Name: Sun ATM155UTP SBUS Adapter X1061A

This product has been tested and complies with the following EMC and Safety standards:

EMC

EC – Europe

This equipment complies with the following requirements of the EMC Directive 89/336/EEC:

EN55022 / CISPR22 (1985)		Class A
EN50082-1	IEC801-2 (1991)	4 kV (direct), 8 kV (air)
	IEC801-3 (1984)	3 V/m
	IEC801-4 (1988)	1.0 kV power lines, 0.5 kV signal lines
EN61000-3-2/IEC1000-3-2 (1994)		Pass

Safety

This equipment complies with the following requirements of the Low Voltage Directive 73/23/EEC:

- EN60950/IEC950

Supplementary Information

This product was tested and complies with all the requirements for the CE Mark when connected to a Sun workstation or server.

/ S /

Dennis P. Symanski Date
Manager, Product Compliance

Sun Microsystems Computer Company
901 San Antonio Road
Palo Alto, CA 94303-4900, USA
Tel: 415-786-3255
Fax: 415-786-3723

/ S /

John Shades Date
Manager, Quality Assurance

Sun Microsystems Limited
Springfield, Linlithgow
West Lothian, EH49 7LR
Scotland, United Kingdom
Tel: 0506 670000
Fax: 0506 760011

Declaration of Conformity

Compliance ID: SBUS-622MMF

Product Name: Sun ATM622MMF SBUS Adapter X1064A

This product has been tested and complies with the following EMC and Safety standards:

EMC

EC – Europe

This equipment complies with the following requirements of the EMC Directive 89/336/EEC:

EN55022 / CISPR22 (1985)	Class A
EN50082-1 IEC801-2 (1991)	4 kV (direct), 8 kV (air)
IEC801-3 (1984)	3 V/m
IEC801-4 (1988)	1.0 kV power lines, 0.5 kV signal lines
EN61000-3-2/IEC1000-3-2 (1994)	Pass

Safety

This equipment complies with the following requirements of the Low Voltage Directive 73/23/EEC:

- EN60950/IEC950

Supplementary Information

This product was tested and complies with all the requirements for the CE Mark when connected to a Sun workstation or server.

/ S /

Dennis P. Symanski
Manager, Product Compliance

Date

Sun Microsystems Computer Company
901 San Antonio Road
Palo Alto, CA 94303-4900, USA
Tel: 415-786-3255
Fax: 415-786-3723

/ S /

John Shades
Manager, Quality Assurance

Date

Sun Microsystems Limited
Springfield, Linlithgow
West Lothian, EH49 7LR
Scotland, United Kingdom
Tel: 0506 670000
Fax: 0506 760011

Declaration of Conformity

Compliance ID: 155MMF

Product Name: Sun ATM/P-155MMF

This product has been tested and complies with the following EMC and Safety standards:

EMC

EC – Europe

This equipment complies with the following requirements of the EMC Directive 89/336/EEC:

EN55022 / CISPR22 (1985)	Class A
EN50082-1 IEC801-2 (1991)	4 kV (direct), 8 kV (air)
IEC801-3 (1984)	3 V/m
IEC801-4 (1988)	1.0 kV power lines, 0.5 kV signal lines
EN61000-3-2/IEC1000-3-2 (1994)	Pass

Safety

This equipment complies with the following requirements of the Low Voltage Directive 73/23/EEC:

- EN60950/IEC950 (1993)

Supplementary Information

This product was tested and complies with all the requirements for the CE Mark when connected to a Sun workstation or server.

/ S /

Dennis P. Symanski
Manager, Product Compliance Date

Sun Microsystems Computer Company
2550 Garcia Avenue, M/S UMPK15-102
Mt. View, CA 94043, USA
Tel: 415-786-3255
Fax: 415-786-3723

/ S /

John Shades
Manager, Quality Assurance Date

Sun Microsystems Limited
Springfield, Linlithgow
West Lothian, EH49 7LR
Scotland, United Kingdom
Tel: 0506 670000
Fax: 0506 760011

Declaration of Conformity

Compliance ID: 155UTP

Product Name: Sun ATM/P-155UTP

This product has been tested and complies with the following EMC and Safety standards:

EMC

EC — Europe

This equipment complies with the following requirements of the EMC Directive 89/336/EEC:

EN55022 / CISPR22 (1985)	Class A
EN50082-1 IEC801-2 (1991)	4 kV (direct), 8 kV (air)
IEC801-3 (1984)	3 V/m
IEC801-4 (1988)	1.0 kV power lines, 0.5 kV signal lines
EN61000-3-2/IEC1000-3-2 (1994)	Pass

Safety

This equipment complies with the following requirements of the Low Voltage Directive 73/23/EEC:

- EN60950/IEC950 (1993)

Supplementary Information

This product was tested and complies with all the requirements for the CE Mark when connected to a Sun workstation or server.

/ S /

Dennis P. Symanski
Manager, Product Compliance

Date

Sun Microsystems Computer Company
2550 Garcia Avenue, M/S UMPK15-102
Mt. View, CA 94043, USA
Tel: 415-786-3255
Fax: 415-786-3723

/ S /

John Shades
Manager, Quality Assurance

Date

Sun Microsystems Limited
Springfield, Linlithgow
West Lothian, EH49 7LR
Scotland, United Kingdom
Tel: 0506 670000
Fax: 0506 760011

Wiring Scheme and Pin Descriptions

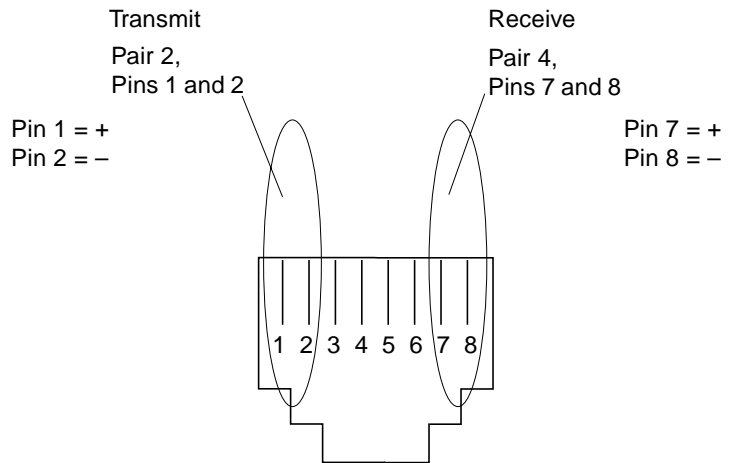


FIGURE B-1 Designation T568B

SunATM Adapters Specifications

C.1 SunATM/S Adapter Specifications

This section describes the SunATM/S adapter specifications.

C.1.1 SunATM/S 155 Adapter Specifications

C.1.1.1 Physical Dimensions

TABLE C-1 Physical Dimensions of SunATM/S 155 Adapters

Dimension	Measurement
Length	5.78 in. (146.70 mm)
Width	3.3 in. (83.82 mm)

C.1.1.2 Performance Specification

TABLE C-2 Performance Specifications of SunATM/S 155 Adapters

Feature	Specification
SBus Clock	25 MHz max., 12.5 MHz min.
Max SBus Burst Transfer Rate	34 Mbytes/sec (approximately)
Steady State SBus Transfer Rate	5 Mbytes/sec
SBus Data/Address Lines	D (31:0)/PA (27:0)
SBus Modes	Master/Slave
Capacitance per SBus Signal Line	< = 20 pF
SBus Parity	Yes
SBus Version	IEEE 1496
SBus Burst Sizes	16/32/64

C.1.1.3 Power Specifications

TABLE C-3 Power Specifications of SunATM/S 155 Adapters

Specification	Measurement
Power Dissipation	9.5 Watt max.
Voltage Tolerance	+/- 5%
Ripple	< = 100 mV
Operational Current	5V, 2.0 Amps

C.1.1.4 Environmental Specifications

TABLE C-4 Environmental Specifications of SunATM/S 155 Adapters

Condition	Operating Specification	Storage Specification
Temperature	0 to 70 C(+32 to +131 F)	-25 to 70 C(-25 to +131 F)
Relative Humidity	5 to 85% non-condensing (40 C, wet bulb temperature)	0 to 95% non-condensing 40 C/hour
Altitude	-1000 to +15,000 ft.	-1000 to +50,000 ft.

TABLE C-4 Environmental Specifications of SunATM/S 155 Adapters *(Continued)*

Condition	Operating Specification	Storage Specification
Shock	5g, 1/2 sine wave, 11 msec	30g, 1/2 sine wave, 11 msec
Vibration, pk to pk displacement	0.005 in. max. (5 to 32 Hz)	0.1 in. max (5 to 17 Hz)
Vibration, peak acceleration	0.25g (5 to 500 Hz) (Sweep Rate = 1 octave/min.)	1.0g (5 to 500 Hz) (Sweep Rate = 1 octave/min.)

C.1.2 SunATM/S 622 MMF Adapter Specifications

C.1.2.1 Physical Dimensions

TABLE C-5 Physical Dimensions of SunATM/S 622 Adapters

Dimension	Measurement
Length	5.78 in. (146.70 mm)
Width	6.6 in. (167.64 mm)

C.1.2.2 Performance Specifications

TABLE C-6 Performance Specifications of SunATM/S 622 Adapters

Feature	Specification
SBus Clock	25 MHz max., 12.5 MHz min.
Max SBus Burst Transfer Rate	100 Mbytes/sec (approximately)
Steady State SBus Transfer Rate	100 Mbytes/sec
SBus Data/Address Lines	D (31:0)/PA (27:0)
SBus Modes	Master/Slave
Capacitance per SBus Signal Line	< = 20 pF
SBus Parity	Yes
SBus Version	IEEE 1496
SBus Burst Sizes	16/32/64

C.1.2.3 Power Specifications

TABLE C-7 Power Specifications of SunATM/S 622 Adapters

Specification	Measurement
Power Dissipation	13 Watt max.

TABLE C-7 Power Specifications of SunATM/S 622 Adapters *(Continued)*

Specification	Measurement
Voltage Tolerance	+/- 5%
Ripple	< = 100 mV
Operational Current	5V, 2.6 Amps

C.1.2.4 Environmental Specifications

TABLE C-8 Environmental Specifications of SunATM/S 622 Adapters

Condition	Operating Specification	Storage Specification
Temperature	0 to 70 C(+32 to +131 F)	-25 to 70 C(-25 to +131 F)
Relative Humidity	5 to 85% non-condensing (40 C, wet bulb temperature)	0 to 95% non-condensing 40 C /hour
Altitude	-1000 to +15,000 ft.	-1000 to +50,000 ft.
Shock	5g, 1/2 sine wave, 11 msec	30g, 1/2 sine wave, 11 msec
Vibration, pk to pk displacement	0.005 in. max. (5 to 32 Hz)	0.1 in. max (5 to 17 Hz)
Vibration, peak acceleration	0.25g (5 to 500 Hz) (Sweep Rate = 1 octave/min.)	1.0g (5 to 500 Hz) (Sweep Rate = 1 octave/min.)

C.2 SunATM/P Adapter Specifications

This section describes the SunATM/P adapter specifications.

C.2.1 SunATM/P 155 Adapter Specifications

C.2.1.1 Physical Dimensions

Physical Dimensions of SunATM/P 155 Adapters

Dimension	Measurement
Length	6.875 in. (17.46 cm)
Width	4.125 in. (10.48 cm)

C.2.1.2 Performance Specifications

TABLE C-9 Performance Specifications of SunATM/P 155 Adapters

Feature	Specification
PCI Clock	66 MHz max., 33MHz min.
Maximum Burst Transfer Rate	34 Mbytes/sec (approximately)
Steady State Transfer Rate	5 Mbytes/sec
PCI Bus Modes	Master/Slave

C.2.1.3 Power Specifications

TABLE C-10 Power Specifications of SunATM/P 155 Adapters

Specification	Measurement
Power Dissipation	11 Watt max.

TABLE C-10 Power Specifications of SunATM/P 155 Adapters *(Continued)*

Specification	Measurement
Voltage Tolerance	+/- 5%
Ripple	< = 100 mV
Operational Current	2.2 Amps

C.2.1.4 Environmental Specifications

TABLE C-11 Environmental Specifications of SunATM/P 155 Adapters

Condition	Operating Specification	Storage Specification
Temperature	0 to 70 C(+32 to +131 F)	-25 to 70 C(-25 to +131 F)
Relative Humidity	5 to 85% non-condensing (40 C, wet bulb temperature)	0 to 95% non-condensing 40 C/hour
Altitude	-1000 to +15,000 ft.	-1000 to +50,000 ft.
Shock	5g, 1/2 sine wave, 11 msec	30g, 1/2 sine wave, 11 msec
Vibration, pk to pk displacement	0.005 in. max. (5 to 32 Hz)	0.1 in. max (5 to 17 Hz)
Vibration, peak acceleration	0.25g (5 to 500 Hz) (Sweep Rate = 1 octave/min.)	1.0g (5 to 500 Hz) (Sweep Rate = 1 octave/min.)

C.2.2 SunATM/P 622 MMF Adapter Specifications

C.2.2.1 Physical Dimensions

TABLE C-12 Physical Dimensions of SunATM/P 622 Adapters

Dimension	Measurement
Length	6.875 in. (17.46 cm)
Width	4.125 in. (10.48 cm)

C.2.2.2 Performance Specifications

TABLE C-13 Performance Specifications of SunATM/P 622 Adapters

Feature	Specification
PCI Clock	66 MHz max., 33MHz min.
Maximum Burst Transfer Rate	150 Mbytes/sec (approximately)
Steady State Transfer Rate	5 Mbytes/sec
PCI Bus Modes	Master/Slave

C.2.2.3 Power Specifications

TABLE C-14 Power Specifications of SunATM/P 622 Adapters

Specification	Measurement
Power Dissipation	14.7 Watt max.
Voltage Tolerance	+/- 5%
Ripple	< = 100 mV
Operational Current	2.94 Amps

C.2.2.4 Environmental Specifications

TABLE C-15 Environmental Specifications of SunATM/P 622 Adapters

Condition	Operating Specification	Storage Specification
Temperature	0 to 70 C(+32 to +131 F)	-25 to 70 C(-25 to +131 F)
Relative Humidity	5 to 85% non-condensing (40 C, wet bulb temperature)	0 to 95% non-condensing 40 C/hour
Altitude	-1000 to +15,000 ft.	-1000 to +50,000 ft.
Shock	5g, 1/2 sine wave, 11 msec	30g, 1/2 sine wave, 11 msec
Vibration, pk to pk displacement	0.005 in. max. (5 to 32 Hz)	0.1 in. max (5 to 17 Hz)
Vibration, peak acceleration	0.25g (5 to 500 Hz) (Sweep Rate = 1 octave/min.)	1.0g (5 to 500 Hz) (Sweep Rate = 1 octave/min.)

Troubleshooting and Error Messages

D.1 Troubleshooting While Starting a SunATM Interface

There are many steps involved in making an interface active on an ATM network. Problems in your configuration may cause a failure at any number of points along the way. The following sections contain steps you can take to determine where in the process your system failed, and what to do to remedy the situation. If you continue to experience problems, information gathered from these steps will help your service provider diagnose the problem.

D.1.1 Generic Configuration

▼ To Diagnose Problems

1. **Make sure that there is an entry for the interface in** `/etc/opt/SUNWconn/atm/atmconfig`.

Configuration of an interface begins during system boot. Configuration will be attempted for all interfaces listed in `/etc/opt/SUNWconn/atm/atmconfig`. For information about the format of this file, see Section 6.1 “Editing the `/etc/opt/SUNWconn/atm/atmconfig` File” on page 6-1, and the `atmconfig(4)` man page.

2. Check to see if any error messages were printed during the boot process.

If there were error messages, see Section D.2 “Error Messages.”

3. Verify linkstate in `qccstat(1M)`.

This command indicates the signalling status of your interface. If the linkstate is not `DL_ACTIVE`, your interface is not communicating properly with your switch.

- Make sure that your switch and interface are configured to run the same version of UNI signalling.

The SunATM software supports UNI versions 3.0, 3.1, and 4.0; set the version for each interface in the `/etc/opt/SUNWconn/atm/atmconfig` file.

- Verify that your interface is physically connected to the switch and that the switch sees the physical connection (most switches have a physical link LED for each port).

If your interface is a multimode fiber interface, one possible cause for a bad physical connection is that transmit and receive are swapped. “transmit” on your interface should be connected to “receive” on the switch, and “receive” on your interface to “transmit” on the switch. There is generally writing on one of the cables in a transmit-receive pair so that the two cables are distinct.

4. Verify that an address has been registered with the switch.

The `qccstat(1M)` command also lists all addresses registered to the interface with the switch. See Section 7.1 “ATM Addresses and Address Registration” on page 7-2, for more information about address registration. If there are no addresses registered, the `ilmid` daemon on your system is not communicating properly with the switch.

- Verify that there are incoming packets on VC 16 using `atmstat(1M)`.

If there aren’t any incoming packets, the switch is not responding to ILMI requests, and you should check its ILMI configuration.

- Verify that there are outgoing packets on VC 16 using `atmstat(1M)`.

If you do not see any outgoing packets on VC 16, your interface is not transmitting ILMI packets. Verify that `ilmid` is running on your system, and if necessary, start it in the background. Starting `ilmid` with the `-v` flag causes it to print a notice for every message received or transmitted, along with other diagnostic information.

5. Interfaces that are not running Classical IP or LAN Emulation will not appear in the output of the `ifconfig` command.

`ifconfig(1M)` displays interfaces that have been configured for IP. In order to support IP, ATM interfaces must run either Classical IP or LAN Emulation. Therefore, an ATM interface that is not configured to support IP by running one of these two protocols will not be displayed by `ifconfig`.

6. Verify the packets that are moving over the network with the `/etc/opt/SUNWconn/bin/atmsnoop` command.

D.1.2 Classical IP Configuration

1. Check all of the generic configuration points.

These are issues that apply to all SunATM interfaces, so they must all be working in order for Classical IP to work.

2. Verify the output of `ifconfig(1M)`.

Executing the command `ifconfig -a` should display the SunATM interface, `baN`, where `N` is the instance number.

- If your interface does not appear, an error probably occurred during the boot process.

Check for error messages during the boot process. The meanings and possible solutions for error messages can be found in Section D.2 “Error Messages.”

- If your interface appears, but has incorrect information, verify your configuration files.

The information given to `ifconfig` comes from the `/etc/opt/SUNWconn/atm/atmconfig` and `/etc/opt/SUNWconn/atm/aarconfig` files. Check the entries in those files that apply to this interface and verify their contents. For descriptions of the file formats, see Section 6.1 “Editing the `/etc/opt/SUNWconn/atm/atmconfig` File,” and Section 6.2.1 “Editing the `/etc/opt/SUNWconn/atm/aarconfig` File,” or the `atmconfig(4)` and `aarconfig(4)` man pages.

3. Check the `setup_state` with `aarstat(1M)`.

This command will provide information about the Classical IP status on your interface. The `setup_state` refers to the completion of the `aarsetup` program.

- If the `setup_state` is `setup-started`.

This indicates that the `aarsetup` program has not completed; it may be delayed by slow switch responses, or failed attempts to register ATM addresses in `/etc/opt/SUNWconn/atm/aarconfig`. Make sure that the local address given for your interface in `/etc/opt/SUNWconn/atm/aarconfig` is unique to this switch. Using `$myaddress` and the reserved server addresses is a good way to guarantee that all addresses are unique. After making any changes to `/etc/opt/SUNWconn/atm/aarconfig`, run `aarsetup` again.

- If the state is not `setup-started` or `setup-finished`.

Verify that the addresses and interfaces in `/etc/opt/SUNWconn/atm/aarconfig` are valid, and run `aarsetup` again. If you see any error messages, check their meanings in Section D.2 “Error Messages.”

4. Verify the `interface_state` in `aarstat(1M)`.

The `interface_state` is either up or down, and reflects the linkstate given in the output of `qccstat`. If the linkstate is `DL_ACTIVE`, the `interface_state` is up; otherwise, the `interface_state` is down. If `aarstat` indicates that the `interface_state` is down, try the suggestions for a linkstate that is not `DL_ACTIVE` given in Section D.1.1 “Generic Configuration.”

5. Make sure Classical IP is configured correctly.

The `aarstat(1M)` output lists several parameters for Classical IP. The field `arpcsmode` lists whether Classical IP is running as a client, a server, or stand-alone (a client with no server configured). Verify that this is correct; if it is not, check your `/etc/opt/SUNWconn/atm/aarconfig` file entries.

6. If the system is a Classical IP client, verify the server connection.

On systems running in client mode, `aarstat` also provides information about the server. Verify the server address, and that the `server_state` is connected.

7. If the `server_state` is no-connection or connecting.

The system is likely having a problem establishing a connection to the server. Verify that the server address is correct, and that there is a system on the network which has registered that address. The server and applicable switch ports must also be configured to support UNI signalling, also called Q.2931 or Q.93b.

8. Verify that addresses are resolved and connections are made with the `ping(1M)` command.

Once you have two systems configured and running to this point, they should be able to `ping` each other. On `client1`, type: `ping client2`. You should receive a response, after a small delay, of `client2 is alive`.

If the `ping` is not successful:

1. Check that ARP requests are being sent to the server.

Find the `server_vci` in the output of `aarstat`. Then run `atmstat`, and verify that there are outgoing packets on that VC. If not, make sure that your interface is up and configured properly.

2. Make sure that you are receiving ARP responses from the server.

In the `atmstat` output, check the output packets for the server VC (found in the `aarstat` information). If none are being received, your server is not responding to ARP requests from the client. If it is a SunATM server, verify its Classical IP status with the suggestions given here. If not, verify that it is up and running as a server.

3. Make sure the address is resolved correctly.

Run the `atmarp` command for the system you are trying to `ping`, and verify that its IP address has been resolved to the correct ATM address. If not, make sure that the remote system is registering the correct address with the ATM ARP server. If the address has not been resolved at all, make sure that the remote system has a connection to the server.

4. Verify that a connection has been established between the two systems.

The output of `qccstat` lists the source and destination addresses of all open connections. You should have at least one connection to the server, and you should also see a connection to the remote host you are trying to `ping`. If not, make sure both interfaces are up and registered with the switch, and that both interfaces and the switch are running UNI signalling (Q.2931 or Q.93b).

5. Check for IP problems.

If the address has been resolved correctly, and a connection has been established between the two systems, but they still cannot `ping`, the problem is likely outside the scope of ATM.

D.1.3 LAN Emulation Configuration

1. **Check all of the generic configuration points.**

These are issues that apply to all SunATM interfaces, so they must all be working in order for LAN Emulation to work.

2. **Verify the output of `ifconfig(1M)`.**

Executing the command `ifconfig -a` should display the ATM LAN Emulation interface, `laneN`, where `N` is the instance number.

- If your interface does not appear, an error probably occurred during the boot process.

Check for error messages during the boot process. The meanings and possible solutions for error messages can be found in Section D.2 “Error Messages.”

- If your interface appears, but has incorrect information, verify your configuration files.

The information given to `ifconfig` comes from the `/etc/opt/SUNWconn/atm/atmconfig` and `/etc/opt/SUNWconn/atm/laneconfig` files. Check the entries in those files that apply to this interface and verify their contents. For descriptions of the file formats, see Section 6.1 “Editing the `/etc/opt/SUNWconn/atm/atmconfig` File” and Section 6.3.1 “Editing the `/etc/opt/SUNWconn/atm/laneconfig` File,” or the `atmconfig(4)` and `laneconfig(4)` man pages.

3. Check the `setup_state` with `lanestat(1M)`.

This command provides information about the LAN Emulation status on your interface. The `setup_state` refers to the completion of the `lanesetup` program.

- If the `setup_state` is `setup-started`.

This indicates that the `lanesetup` program has not completed; it may be delayed by slow switch responses, or failed attempts to register ATM addresses in `/etc/opt/SUNWconn/atm/laneconfig`. Make sure that the local address given for your interface in `/etc/opt/SUNWconn/atm/laneconfig` is unique to this switch. Using the variable `$myaddress` for all systems is a good way to guarantee that all addresses are unique. After making any changes to `/etc/opt/SUNWconn/atm/laneconfig`, run `lanesetup` again.

- If the state is not `setup-started` or `setup-finished`.

Verify that the addresses and interfaces in `/etc/opt/SUNWconn/atm/laneconfig` are valid, and re-run `lanesetup`. If you see any error messages, check their meanings in Section D.2 “Error Messages.”

4. Verify that a connection has been made to the LAN Emulation server (LES).

A LAN Emulation client must establish and maintain a connection to the LES. In most cases, the LES also establishes and maintains a second connection to the client. Find the LES address in the output of `lanestat`, and then look for connections with that address as the destination or source in the output of `qccstat`.

If you do not see any connections with that address:

- If you have a LAN Emulation configuration server (LECS):

Make sure that the correct address is configured for the LECS. By default, the SunATM software uses the ATM Forum well-known address. If your LECS uses a different address, enter the alternate address in the `/etc/opt/SUNWconn/atm/laneconfig` file. See Section 6.3.1 “Editing the `/etc/opt/SUNWconn/atm/laneconfig` File,” for information on editing `/etc/opt/SUNWconn/atm/laneconfig`. You can check the address currently being used in the output of `lanestat`.

- If you do not have an LECS:

One of the LECS functions is to provide the LES address, so if you do not have an LECS, you must provide the address. Create an entry in `/etc/opt/SUNWconn/atm/laneconfig`. See Section 6.3.1 “Editing the `/etc/opt/SUNWconn/atm/laneconfig` File.” You can check the LES address currently being used in the output of `lanestat`.

- Verify that the LECS, if present, and LES are configured properly.

5. Verify that a connection has been made to the BUS.

In addition to the LES connection(s), a LAN Emulation client must also establish and maintain a connection to the BUS, and the BUS typically establishes and maintains a second connection to the client. You can find the BUS ATM address in the output of

lanestat, and then verify that there is a connection with that address as the destination, and probably a second connection with that address as source, in the output of `qccstat`. If there are not any connections, verify that the BUS is configured properly.

6. Verify that the host has joined the Emulated LAN

The `lanestate` field in the output of `lanestat` indicates that the client is in the active state.

If your system cannot join the emulated LAN, there may be a problem with the way in which your LAN Emulation Services are configured. If the Emulated LAN uses an MTU size larger than 9 Kbytes, the SunATM host will not join (9 Kbytes is the largest MTU size supported by the SunATM product). If the host is not able to join, an error message will be printed with an explanation.

7. Verify that addresses are resolved and connections are made with the `ping` command.

Once you have two systems configured and running to this point, they should be able to `ping` each other. On `client1`, type: `ping client2`. You should receive a response, after a small delay, of `client2 is alive`.

If the `ping` is not successful:

1. Check that the IP hostname or address is resolved to a MAC address.

LAN Emulation requires two address resolution steps to make a call. The first is to resolve an IP address to a MAC address. From the perspective of IP and ARP, this works exactly like it does on an ethernet interface; using the `arp` command, you can verify that this resolution has been made correctly. If it has not, verify the connections to the BUS, and make sure data is being transmitted and received on the connection(s) to the BUS by finding the VC in the output of `qccstat`, and looking at the statistics for that VC in `atmstat`.

2. Check that the MAC address has been resolved to an ATM address.

This is the second address resolution step, and is accomplished by the LAN Emulation software and communication with the LES. You can use the `lanearp` command to verify that MAC addresses have been properly resolved to ATM addresses. If they have not, verify the connections to the LES, and make sure data is being transmitted and received on the connection(s) to the LES by finding the VC in the output of `qccstat` and looking at the statistics for that VC in `atmstat`.

3. Verify that a connection has been established between the two systems.

The output of `qccstat` lists the source and destination addresses of all open connections. There you should see a connection to the remote host you are trying to `ping`. If not, make sure both interfaces are up and registered with the switch, and that both interfaces and the switch are running UNI signalling (Q.2931 or Q.93b).

4. Check for IP problems.

If the address has been resolved correctly, and a connection has been established between the two systems, but they still cannot ping, the problem is likely outside the scope of ATM.

D.1.4 Common Problems

This section describes some common problems that you may experience during or after the SunATM adapter installation. Please review this section before calling Sun Service for assistance.

D.1.4.1 Are you replacing an old SunATM SBus adapter?

- **If you are replacing an old SunATM/S 155 adapter with a new adapter, you must edit the `/etc/path_to_inst` file to remove the old device instance.**

The SunATM/S 155 adapter originally shipped with an FCode name of “ba” (part numbers 501-2794-07, 501-2795-05, and prior versions). Since then, we have changed the naming convention to include SUNW at the beginning of every device name. When a third-party adapter was found, which also used the name property “ba”, the SunATM/S 155 adapter was updated to use the “SUNW,ba” name property instead (the change was made to part numbers 501-2794-08, 501-2795-06, and compatible versions).

As a result, when an older SunATM/S 155 adapter (with the “ba” name property) is replaced by a newer SunATM/S 155 adapter (with the “SUNW,ba” name property), the system does not recognize the new adapter as a replacement. Instead, the system sees it as a new interface and assigns a new instance number to the adapter. The `/etc/path_to_inst` file is created by the Solaris operating system to identify installed devices and their instance numbers. When a SunATM/S 155 adapter (with the “ba” name) is installed in a system, `/etc/path_to_inst` has an entry, similar to the following, to identify it as ba0:

```
"/sbus@1f,0/ba@0,0" 0 "ba"
```

When a replacement adapter (with the “SUNW,ba” name) is installed into the same location and the system is rebooted, it treated as a new device and a new entry in `/etc/path_to_inst` is created for ba1:

```
"/sbus@1f,0/SUNW,ba@0,0" 1 "ba"
```

To correct this, delete the original entry that contains the “ba” name. Then, modify the second field of the new entry, which contains the name “SUNW,ba”, to reflect the proper instance number. In this example, the new entry is designated as instance 0:

```
"/sbus@1f,0/SUNW,ba@0,0" 0 "ba"
```

After you have modified and saved `/etc/path_to_inst`, reboot the system for the changes to take effect.

Note – The physical name listed in `/etc/path_to_inst` varies from one architecture to another and might not match the previous examples exactly. However, modify only the instance number field. Be sure to leave all other fields as they are.

D.1.4.2 Are you trying to use the `/usr/sbin/arp` command?

Since the Classical Internet Protocol (IP) network model resolves IP-to-ATM address pairs rather than IP to MAC address pairs, the `/usr/sbin/arp` command does not support Classical IP interfaces at this time. A version of the `arp` command, `/etc/opt/SUNWconn/atm/bin/atmarp`, provides similar functionality for Classical IP interfaces. Refer to the `atmarp (1M)` man page for more information.

D.1.4.3 Are you using a Router with Classical IP and LAN Emulation (LANE)?

Performance problems occur if a router uses ATM Classical IP (default 9180 byte MTU) and LAN Emulation (default 1500 byte MTU) links simultaneously when a TCP connection is set up using one interface in one direction and the other interface in the opposite direction, TCP is confused about the maximum packet size.

For example, suppose a TCP connection is set up between Host A and Host B, where packets from Host A travel to Host B over the LANE interface and packets from Host B travel to Host A travel over the Classical IP interface. Host A attempts to send a 9180 byte packet that cannot traverse the LANE network to Host B. TCP recovers from this error and retransmits the packet, but a significant performance loss will be noted.

Possible workarounds to improve performance are:

- Adjust the MTU size, if possible, of the Classical IP link to 1500 bytes.
- Depending upon the network topology, adjust the routing table on Host B to ensure that the route back to Host A points to the LANE interface.

This problem is not unique to ATM networks. It may affect any network configuration that has multiple routes with differing MTUs (such as FDDI and Ethernet or Token Ring). The problem is more pronounced with ATM subnets because of the different default MTUs of Classical IP and LANE.

D.1.4.4 Are you trying to use the `/usr/sbin/snoop` command?

The `/usr/bin/snoop` command, which can be used to detect network problems, does not support SunATM interfaces at this time. A version of the `snoop` command, `/etc/opt/SUNWconn/atm/bin/atmsnoop`, provides this support. Refer to the `atmsnoop(1M)` man page for more information.

D.1.4.5 Do you want to increase system performance by adjusting TCP/IP parameters?

TCP/IP performance over an ATM network can be poor unless you carefully configure your network. Poor performance usually occurs because the TCP/IP packets are segmented into cells for transmission by the ATM software. Therefore, a loss of a single cell can cause the loss of an entire TCP/IP packet which can lead to retransmissions that congest the network. When it detects congestion, the destination system reduces the transmission rate, which significantly reduces the network performance.

You can achieve better network performance from the SunATM adapter and software by adjusting your application's socket buffer size to 48 Kbytes. Refer to the application's documentation for instructions on how to set the socket buffer size.

D.1.4.6 Do you want to improve the performance of NFS version 3?

If you are using the Solaris 2.6 operating environment, you can improve your NFS performance with a workaround also documented in the *Solaris 2.6 NFS Administration Guide*. Refer to the bug report (bugid 4123298) for more information about the NFS throughput over large MTU links in Solaris 2.6.

You can override defaults for the NFSv2 and NFSv3 readaheads on the NFS client. The kernel variable names are `nfs_nra` and `nfs3_nra`, respectively. The default values are 1 (1 block). Setting these variables to a value between 2-4 (2 to 4 blocks) improves performance (transfer rate) but increases memory used by the NFS client. This is not a problem if you mostly read files sequentially, but can be if you read files randomly.

The parameters affect all the NFS mounts on the client in the same way. You can change these tunable kernel parameters in the `/etc/system` file on the client machine.

To set the NFS Parameters:

1. **To change the `nfs3_nra` or `nfs_nra` values, you must edit the `/etc/system` file.**
For example, to set `nfs3_nra` to 4, add the following line to the `/etc/system` file.

```
set nfs:nfs3_nra=4
```

2. **Reboot the NFS client machine.**

D.1.4.7 Do you want to decrease the NFS Read and Write Buffer Sizes in a Solaris 2.5.1 or earlier operating environment?

Note – These workarounds are for systems running the Solaris 2.5.1 operating environment only.

When running Network File System (NFS) Version 3 on a SunATM network, you may notice poor network performance. By default, the version of the NFS protocol used between the client and the server is the highest one available on both systems. If the client is using version 2.5.1 of the Solaris software environment, it will use the NFS Version 3 software to share resources. Refer to the bug report (buidid 1260524) for more information about the NFS Version 3 performance issues on Solaris 2.5.1. This section describes two workarounds which will improve the NFS Version 3 performance on a SunATM network.

Note – These workarounds only apply when using Classical IP over a SunATM network, or when LAN Emulation interfaces are configured to use a 9218 byte maximum transmission unit (MTU). These workarounds will not affect LAN Emulation interfaces using 1500 or 4528 MTU sizes. Use the `ifconfig -a` command to find out the MTU size of your LAN Emulation interfaces.

The best way to increase NFS performance on a SunATM network is to decrease the read and write buffer sizes to 4096 bytes on NFS filesystems mounted over a SunATM network. To set the read and write buffer sizes to 4096 bytes, use the `rsize` and `wsiz` parameters when mounting the NFS filesystem.

For example, to mount a NFS Version 3 filesystem on a SunATM network, type:

```
# mount -o rsize=4096,wsiz=4096 server:/usr/src /usr/src
```

If the previous workaround does not solve your NFS performance problems, set the TCP receive highwater mark to 65535 bytes in order to increase NFS performance. Change this setting only if you cannot use the previous workaround. To set the TCP receive highwater mark to 65535 bytes, use the `ndd` command:

```
# ndd -set /dev/tcp tcp_recv_hiwat 65535
```

Perform this command every time you reboot; add the entire `ndd` command shown above to the `/etc/rc2.d/S00sunatm` run control script. Refer to your Solaris documentation for more information about editing run control scripts.

D.1.4.8 Are you trying to mount a diskless, dataless, or autoclient system?

The SunATM 4.0 adapters do not currently support diskless, dataless, or autoclient systems. The `root` filesystem must be local for the SunATM adapter to operate.

D.1.4.9 Did the `atmtest` diagnostic fail?

If the bandwidth or outstanding packets value is set too high on your system, the SunVTS `atmtest` diagnostic can fail, giving a error similar to the following:

```
SUNWvts.atmtest.4000 09/17/98 17:33:10 atmtest ba0
WARNING: "VC30 dropped pkt, seq: exp=41, obs=43; len: exp=1747,
obs=6022"
```

To correct this error, reduce the bandwidth or the number of outstanding packets in the SunVTS `atmtest`.

D.2 Error Messages

This section includes a list of some of the most common error messages you might see while configuring and bringing up your SunATM interface. For each message, there is a brief explanation of the problem and a possible solution.

D.2.1 Error Messages from S00sunatm

Cannot find ATM utilities in /etc/opt/SUNWconn/atm/bin;
exiting S00sunatm.

The SunATM utility directory /etc/opt/SUNWconn/atm/bin does not exist. Make sure that the SUNWatm package installation completed successfully (see Section 4.2.3 “Checking the Package Installation Using pkginfo” on page 4-5, for more information). You might need to reinstall the package.

Cannot find atmconfig file in /etc; exiting S00sunatm.

The /etc/opt/SUNWconn/atm/atmconfig file provides configuration information to the S00sunatm script so that it can bring up the SunATM interfaces during system boot. If the /etc/opt/SUNWconn/atm/atmconfig file is not present, S00sunatm prints this warning message and exit. The /etc/opt/SUNWconn/atm/atmconfig file is installed with the SUNWatm package as /etc/opt/SUNWconn/atm/atmconfig.template; if you choose autoconfiguration or if no previous /etc/opt/SUNWconn/atm/atmconfig file exists, pkgadd copies this template file to /etc/opt/SUNWconn/atm/atmconfig. If a previous /etc/opt/SUNWconn/atm/atmconfig file exists, it is not overwritten. See Section 6.1 “Editing the /etc/opt/SUNWconn/atm/atmconfig File.”

warning: can't plumb <device>; no UNI version provided

The first entry in /etc/opt/SUNWconn/atm/atmconfig for a physical interface must include a UNI value in the second field.

warning: can't plumb <uni version> on <device>; <uni version>
already plumbed

The system encountered an entry which attempted to plumb a signalling version on an interface that has already been plumbed with a different signalling version. The script ignores the new UNI version and continues processing the entry and the remaining entries in the file.

warning: can't plumb <lane instance>; too many lane instances
on <device>

A physical interface will support up to n lane instances, where n is the number of MAC addresses on the board (or 1 if the board has no MAC address). You can check the number of MAC addresses on a board using the `count` option of the `atmgetmac(1m)` command. If an entry is encountered that attempts to plumb more LANE instances than allowed, this message occurs; processing will continue with the next entry in the file.

warning: can't plumb signalling on <device>
warning: can't plumb classical IP interface <device>
warning: can't plumb <lane instance> on <device>

An error occurred when the script attempted to run `atmplumb(1m)` (either to plumb signalling, classical IP, or LAN Emulation on an interface) with information specified in `/etc/opt/SUNWconn/atm/atmconfig`. The `atmplumb` program will generally display an error message indicating why it failed; use that information to check your values in the `/etc/opt/SUNWconn/atm/atmconfig` entry for device. The script proceeds to read and process the remaining entries in `/etc/opt/SUNWconn/atm/atmconfig`, although further entries for the failed interface are not processed correctly.

warning: invalid interface <lane instance>

The minor number provided in a logical interface name was not in the range 0 - 255. The script proceeds without attempting to configure the invalid lane device.

warning: only one classical ip hostname is allowed on <device>

An additional entry was found containing a Classical IP hostname after an initial Classical IP hostname was already plumbed for the given device. Multiple Classical IP instances are not supported on a single physical interface. The script ignores additional Classical IP information for a physical interface.

warning: <laneN> entry must appear before <laneN:X> entry

When you use logical interface names, the first entry in `/etc/opt/SUNWconn/atm/atmconfig` must always be either `laneN` or `laneN:0`, which are equivalent. All entries that appear before the `laneN` or `laneN:0` entry are ignored.

Please install <SUNWatm>

A required software package is not installed on the system. Install the package and reboot the system.

warning: extra fields for <device> will be ignored

There were additional fields in the `/etc/opt/SUNWconn/atm/atmconfig` entry for the given device name. The script proceeds, ignoring the additional fields.

warning: duplicate entry <lane device>

There were multiple entries in `/etc/opt/SUNWconn/atm/atmconfig` using the same LAN Emulation instance number. This is not a fatal error; the script continues to run. However, only the first entry for each LAN Emulation instance number is configured for LAN Emulation.

warning: not enough fields to configure <device>

The `/etc/opt/SUNWconn/atm/atmconfig` entry for the given device did not have all the required fields. You must edit the `/etc/opt/SUNWconn/atm/atmconfig` file (see Section 6.1 “Editing the `/etc/opt/SUNWconn/atm/atmconfig` File”), filling in all the appropriate information, and reboot the system. Empty fields should be indicated with a hyphen (-).

warning: ifconfig failed for classical IP interface <device>

warning: ifconfig failed for <lane instance>

The script attempted to run `ifconfig` for the specified interface. Error messages indicate why `ifconfig` failed; use that information to check your values in `/etc/opt/SUNWconn/atm/atmconfig`. In particular, verify that the hostname you provide in `/etc/opt/SUNWconn/atm/atmconfig` appears in the `/etc/hosts` file on your system.

warning: invalid lane instance (<lane instance>) for <device>

The lane instance number provided in `/etc/opt/SUNWconn/atm/atmconfig` was not in the range 0 to 999. The script proceeds without attempting to configure the invalid lane instance.

warning: aarsetup failed; could not configure classical IP interfaces

warning: lanesetup failed; could not configure LAN Emulation interfaces

Either the LAN Emulation or the Classical IP startup script failed and exited with an error value. Check the error messages that were printed by `aarsetup` or `lanesetup`, and verify the values you have entered in `/etc/opt/SUNWconn/atm/aarconfig` and/or `/etc/opt/SUNWconn/atm/laneconfig`.

D.2.2 Error Messages from `aarsetup` and `lanesetup`

`aarsetup`: could not become control process

`lanesetup`: could not become control process

An instance of the setup program was running when another instance was started up. The second instance exits with this error message. Make sure that there is not a previous instance of the program still running. The setup program might take a while to complete if the switch is slow to respond.

`aarsetup`: could not open stream to Q93B

lanesetup: could not open stream to Q93B

The program was unable to communicate with the Q93B driver. Make sure that you run `aarsetup` or `lanesetup` as root, and that the SUNW`atm` package has been properly installed.

aarsetup: could not scan input file

lanesetup: could not scan input file

The program was unable to open the `/etc/opt/SUNWconn/atm/aarconfig` or `/etc/opt/SUNWconn/atm/laneconfig` file (or the file specified on the command line). Verify that the appropriate file exists, and has the proper permissions. Also make sure you run `aarsetup` or `lanesetup` as root.

aarsetup: exiting because of errors

lanesetup: exiting because of errors

Errors were encountered while parsing the `/etc/opt/SUNWconn/atm/aarconfig` or `/etc/opt/SUNWconn/atm/laneconfig` file, so the setup program cannot successfully complete. Correct the error condition and then execute either `aarsetup` or `lanesetup`.

aarsetup: <interface> running as a server, but PVC-only 't' entries exist

The `aarsetup` program has found an `L` entry in `/etc/opt/SUNWconn/atm/aarconfig`, meaning that this interface will be running as a server; however, there are table entries (`t` entries) containing only PVCs, which cannot be entered into the server's ATM ARP table. Verify your interface's status (server, client, or stand-alone), make sure all `t` entries include ATM addresses, and execute `aarsetup`. See Section 6.2.1 "Editing the `/etc/opt/SUNWconn/atm/aarconfig` File" on page 6-4, for more information.

aarsetup: waiting for ilmid to provide prefix

lanesetup: waiting for ilmid to provide prefix

In some cases, the address registration process may take several minutes. If so, `aarsetup` or `lanesetup` prints out this message saying that it cannot complete until address registration completes. If the messages continue for more than a minute or two, verify your connection to the switch, and that the switch and interface are both supporting ILMI.

undefined variable

You used a variable in a configuration file without using a `set` statement to assign the value. Add a `set` statement, or correct the variable name, and run `aarsetup` or `lanesetup` again. See Section 6.2.2 "Using Variables in the `/etc/opt/`

SUNWconn/atm/aarconfig File” on page 6-7, and Section 6.3.2 “Using Variables in the /etc/opt/SUNWconn/atm/lanconfig File” on page 6-16, for more information.

variable already defined

You tried to set a variable that had been previously set in the same configuration file. Remove the second assignment and run `aarsetup` or `lanesetup` again.

variable name ill-formed

You created a variable in `/etc/opt/SUNWconn/atm/aarconfig` or `/etc/opt/SUNWconn/atm/lanconfig` that was syntactically invalid. Variable names are a combination of letters, digits, and underscores (`_`). Choose a conforming variable name and run `aarsetup` or `lanesetup` again.

variable name too long

You created a variable in `/etc/opt/SUNWconn/atm/aarconfig` or `/etc/opt/SUNWconn/atm/lanconfig` with a name that is greater than the maximum length (32 characters). Choose a variable name shorter than 32 characters and run `aarsetup` or `lanesetup` again.

variable value too long

You assigned a value longer than the maximum value length of 128 characters to a variable in a configuration file. If you want a longer value, use a combination of variable names, with each value less than 128 characters. After correcting the variable value lengths, run `aarsetup` or `lanesetup` again.

`ifname:cannot join ELAN (frame size too large; please use a different ELAN and rerun lanesetup)`

The largest MTU size supported by the SunATM software is 9 kilobytes. If the LAN Emulation Services try to set a size larger than 9 Kbytes, the SunATM client cannot join the emulated LAN. Reset your LAN Emulation services to use an MTU size less than or equal to 9 Kbytes, and rerun `lanesetup` to join the emulated LAN.

`ifname: frame-size change (please rerun lanesetup)`

The MTU size was changed by the LAN Emulation Services; rerun `lanesetup` to notify IP of the change. There is a slight chance that TCP connections will remain open during this change, and if that is the case, performance on those connections is affected by the change. Either restart the affected applications or reboot the system if this becomes a problem.

`<ifname> could not download the MAC address`

This message indicates that an error occurred while `lanesetup` was attempting to retrieve a MAC address for the indicated interface. Most likely the kernel is out of memory or you have not run `atmplumb` for the specified interface.

`Could not find driver for <ifname>`

Each LAN Emulation interface is associated with an ATM driver when LAN Emulation is set up by `atmplumb`. This message indicates that this interface/driver association has not been made, most likely because you have not run `atmplumb` for the specified interface.

`Not enough MAC addresses on <ATM interface>`

The number of Emulated LANs that can be joined over a single physical interface is limited by the number of MAC addresses on the ATM interface board. This message indicates that you tried to join more Emulated LANs than allowed by the number of MAC addresses on the specified interface. You can find the number of MAC addresses on an interface by using the `count` option on the `atmgetmac(1M)` command; the number of Emulated LANs and lane instances indicated in `/etc/opt/SUNWconn/atm/atmconfig` and `/etc/opt/SUNWconn/atm/laneconfig` should not exceed this number. See Section 6.5 “Supporting Multiple Emulated LANs on a Single Interface.”

D.2.3 Error Messages from the Kernel Drivers

`q93b: warning: link coming back up on <interface>, but ilmid is not running`

The link has gone down and come back up on an interface, but `ilmid` is not running at this time. Register addresses with the switch again, because both the interface and switch must clear out their address tables when the link goes down. Start `ilmid`; if the interface does not seem to be running properly after doing this, you may need to reboot the system. It is likely that the interface was in an unusual or unknown state when the link came back up, and may need to be taken down completely by rebooting.

Managing SunATM Interfaces with SNMP

The SunATM software package provides an SNMP (Simple Network Management Protocol) agent that supports the ATM UNI and LAN Emulation Management Information Bases (MIBs) defined in the User Network Interface and LAN Emulation Specifications. This agent will provide information to a network management system, such as the SunNet Manager™ system.

E.1 Installing the SunATM SNMP Software

The SunATM SNMP software has three parts: the SunATM SNMP daemon, `atmsnmpd`; the SNMP management console configuration files; and the agent configuration files. The management configuration files, which are installed in the `/opt/SUNWconn/atm/snmp` directory, are part of the SUNWatmu package. The `atmsnmp` daemon and its configuration files, which are part of the SUNWatm package, are installed in the `/opt/SUNWconn/atm/bin` and `/opt/SUNWconn/atm/snmp` directories, respectively.

E.2 Setting Up the Management Console

The `schema` and `oid` files containing the required ATM MIB definitions for SunNet Manager are installed in `/opt/SUNWconn/atm/snmp`. In addition, the MIB files in abstract syntax notation (ASN.1) format are included if you are running a network manager that does not use schema files. Refer to the documentation for your network manager for information on how to generate the appropriate configuration files from the MIB files provided.

To configure your SunNet Manager console system to recognize SunATM agents:

1. Start the `snm` console program and save your management database.

Use File -> Save -> Management database from the `snm` console menus.

2. Copy the SunATM schema and oid files, which were installed on the SunATM hosts, to the schema directory on the manager.

The files (`atmf.mib.schema`, `lane.mib.schema`, `atmf.mib.oid`, and `lane.mib.oid`) are installed in the `/opt/SUNWconn/atm/snmp` directory on SunATM hosts. Copy these files to the `/opt/SUNWconn/snm/agents` directory on the management system.

3. Build the object identifier database to include the SunATM object identifiers.

Execute the following command on the management console system:

```
# /opt/SUNWconn/bin/build_oid /opt/SUNWconn/snm/agents
```

4. Start the `snm` console with the `-i` flag:

```
# snm -i
```

5. Load your management database using File -> Load -> Management database from the menus.

The SunATM MIBs, `atmf.mib` and `lane.mib`, should now be available when you create or update a component.

Note – For further information on using SunNet Manager to monitor `snmp` agents, refer to the SunNet Manager documentation.

E.3 Setting Up Agent Systems

To configure a SunATM host system to run as an SNMP agent, select the System Parameters option on the `atmadmin` main menu. From this option, you will be given the choice of setting your system to run as an SNMP agent or not (see Section 5.2 “Using the `atmadmin` Configuration Program,” for more information about the `atmadmin` configuration program).

Note – This option applies to the entire system, and not to individual SunATM interfaces.

Whether the system is running as an agent, the daemon must be running, because it communicates with other parts of the SunATM software. If the system is configured to run as an ATM SNMP agent, the daemon binds to the UDP port used by SNMP network managers (port 161). If the system is not configured as an agent, the agent does not bind to this port, and it does not respond to requests from network management software.

The default community values for the SunATM agent are public for read and private for write. Change these values in the `/etc/opt/SUNWconn/atm/atmsnmp/agent.cnf` file, if needed. This file contains SNMP agent configuration information, and you may customize these values as needed. Restart the `atmsnmpd` daemon after any changes to any of its configuration files, including the `agent.cnf` file.

Application Programmers' Interface

The Application Programmers' Interface (API) provided with this software release is an interim API from Sun that can be used on Sun Platforms.

In the ATM environment, data is sent between hosts over Virtual Circuits (VCs). VCs are point-to-point (or point-to-multipoint) connections between two or more ATM hosts.

VCs can be created in one of two ways:

- Manual configuration at each host and each intermediate network point, also known as Permanent Virtual Circuits (PVC)
- ATM signalling, also known as Switched Virtual Circuits (SVC)

After the VC has been created, the application notifies the SunATM `ba` driver that it is sending and receiving data on the new VC.

- If you are using a PVC, this is the only configuration required on the Sun host.
- If you are using an SVC, there are two required actions:
 1. Create the SVC with the Q.93B driver.
 2. Establish the data connection with the `ba` driver.

Note – For historical reasons, Q.93B and Q.2931 are used interchangeably.

F.1 Using the SunATM API with the Q.93B and the ATM Device Drivers

The architecture illustrated in FIGURE F-1 must be established on a SunATM system in order to perform Q.2931 signalling and send data over established connections. The ATM device driver, SSCOP modules, and Q.93B driver are “plumbed” at boot time. The task remaining for application developers is to create the connections between their application and the Q.93B and ATM device drivers.

Both the Q.93B and ATM device driver are STREAMS drivers; connecting to them is for the most part no different than connecting to other STREAMS drivers. The following sections describe the steps required to connect to each driver, use the drivers to establish ATM connections, and send data over those connections.

For examples of applications that use the SunATM API, see the sample programs installed in `/opt/SUNWconn/atm/examples`.

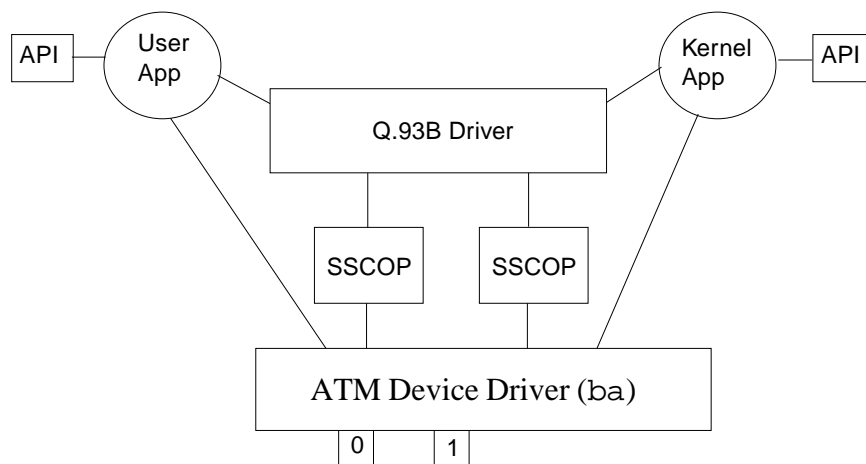


FIGURE F-1 ATM Signalling

F.1.1 Q.93b Driver Interface

The signalling API, called Q.2931 Call Control (qcc), consists of two sets of similar functions: one for applications running in the kernel and one for applications running in user space. Each set provides functions to build and parse Q.2931 signalling messages, which are required to set up and tear down connections.

One additional function assists applications in establishing appropriate connections to the Q.93B driver. `q_ioc_bind` associates a service access point (SAP) with the specified connection to the Q.93B driver. The driver uses the SAP to direct incoming messages to applications.

F.1.2 Establishing a Connection to the Q.93B Driver

Use the `open(2)` system call to obtain a file descriptor to the driver. After opening the driver, `q_ioc_bind` should be called, associating in the Q.93B driver a service access point (SAP) with this application. Finally, if the application is a kernel driver, it should be linked above the Q.93B driver, using the `I_LINK` or `I_PLINK` ioctl (refer to the `streamio(7)` man page for information about this ioctl).

F.1.3 Setting up an ATM Connection Over a Switched Virtual Circuit (SVC)

After connecting to the Q.93B driver, either by directly calling the functions as a user application or by having a setup program connect your application driver as described in the preceding section, the Q.93B driver is available to your application to establish switched virtual circuits (SVCs) using the Q.2931 signalling protocol. The Q.2931 message set is displayed in TABLE F-1.

TABLE F-1 Messages Between the User and the Q.93B Driver

Message Type	Direction*
SETUP	BOTH
SETUP_ACK	UP
CALL_PROCEEDING	BOTH
ALERTING	BOTH
CONNECT	BOTH
CONNECT_ACK	UP
RELEASE	DOWN
RELEASE_COMPLETE	BOTH
STATUS_ENQUIRY	DOWN
STATUS	UP

*UP is from Q.93B to user;
DOWN is from user to Q.93B

TABLE F-1 Messages Between the User and the Q.93B Driver (*Continued*)

Message Type	Direction*
NOTIFY	BOTH
RESTART	BOTH
RESTART_ACK	BOTH
ADD_PARTY	BOTH
ADD_PARTY_ACK	BOTH
ADD_PARTY_REJECT	BOTH
PARTY_ALERTING	BOTH
DROP_PARTY	BOTH
DROP_PARTY_ACK	BOTH
LEAF_SETUP_FAIL	BOTH
LEAF_SETUP_REQ	BOTH

*UP is from Q.93B to user;
DOWN is from user to Q.93B

The Q.93B driver is an M-to-N mux STREAMS driver. Multiple application programs can be plumbed above the driver, and multiple physical interfaces can be connected below Q.93B. Applications can access any or all of the physical interfaces, and messages received on the physical interfaces can be directed to any of the applications. To direct messages through the Q.93B driver, messages from applications must include a physical interface name to identify the outgoing interface and a SAP to identify the application to which the message should be directed on the receiving host.

Send messages to Q.93B by applications according to the format illustrated in FIGURE F-1; kernel applications use `putnext(9f)` to send the mblocks shown and user applications send two corresponding `strbufs` using `putmsg(2)`.

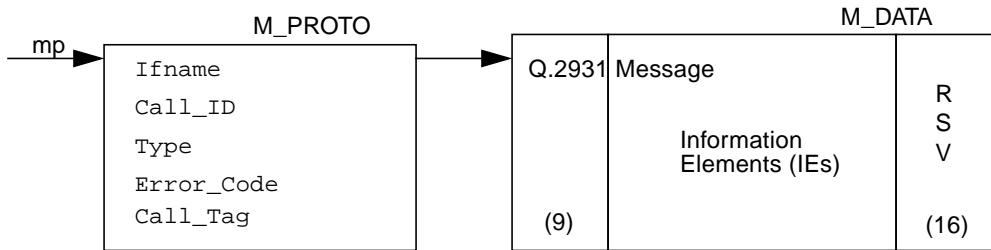


FIGURE F-2 Message Format

TABLE F-2 Fields in the M_PROTO mblock

Message	Explanation
Ifname	Null-terminated string containing the device name
Call_ID	Unique number from Q.93B for each interface.
Type	Same as the Q.2931 message type except there is a local Non-Q.2931 message type SETUP_ACK. The SETUP_ACK message is used to provide the Call_ID to the user.
Error_Code	Error returned from Q.93B when an erroneous message is received from the user. The same mblock chain is returned to the user with the Error_Code field set. The user must always clear this field
Call_Tag	Number assigned by the calling application layer to a SETUP message. When a SETUP_ACK is received from Q.93B, the Call_ID has been set; use the Call_Tag field to identify the acknowledgment (ack) with the original request. From that point on, use the Call_ID value to identify the call.

The structure included in the M_PROTO mblock is defined as the `qcc_hdr_t` structure in the `<atm/qccotypes.h>` header file. In the second mblock, the Q.2931 header portion (9 bytes) of the Q.2931 message is blank and later filled in by the Q.93B driver. The application should also reserve 16 bytes at the end of the second mblock for the layer 2 (Q.SAAL) protocol performance. The `qcc` functions can be used to create messages in this format.

The following sections give a brief overview of Q.2931 signalling procedures, from the perspective of an application using the SunATM API. For more details on the procedures, refer to the ATM Forum's User Network Interface Specification, version 3.0, 3.1, or 4.0. For further information on the `qcc` functions, which are outlined in TABLE F-3, see the appropriate man pages in Section 3 (for user applications) or Section 9F (for kernel applications). You can find the man pages under the function group name or any specific function name. For example, the man page that documents the `qcc_bld_*` function group may be accessed by one of the following

at a command prompt: `man qcc_bld`, `man qcc_bld_setup`, or `man qcc_bld_connect`. FIGURE F-3 illustrates the message flow during typical call setup and tear down.

TABLE F-3 qcc Functions

Name	Functionality	Input	Output
<code>qcc_bld_*</code>	Creates and encodes a message; enables customization of a limited set of values, depending on the message type. Configurable values are passed in as parameters.	Parameter values	Encoded Q.2931 message (in the format shown in FIGURE F-2)
<code>qcc_parse_*</code>	Extracts a defined set of values from an encoded message	Encoded Q.2931 message (in the format shown in FIGURE F-2)	Parameter values
<code>qcc_len_*</code>	Returns the maximum length of the buffer that should be allocated for the second strbuf in a Q.2931 message. Only applicable to user space applications; the kernel API allocates the buffers inside the <code>qcc_bld/qcc_pack</code> functions.	none	Maximum length of the message
<code>qcc_create_*</code>	Creates a message structure with the required values set. You can further customize the structure using <code>qcc_set_ie</code> .	Default parameter values	Message structure (defined in <code><atm/qcctypes.h></code>)
<code>qcc_set_ie</code>	Updates or inserts values for an information element into a message structure.	Message structure and IE structure (defined in <code><atm/qcctypes.h></code>)	Updated message structure
<code>qcc_pack_*</code>	Takes a message structure and encodes it into an actual Q.2931 message, consisting of the two mblks (or strbufs) illustrated in FIGURE F-2.	Message structure (defined in <code><atm/qcctypes.h></code>)	Encoded Q.2931 message (in the format shown in FIGURE F-2)
<code>qcc_unpack_*</code>	The reverse of <code>qcc_pack_*</code> : takes an encoded message and decodes the data into a message structure.	Encoded Q.2931 message (in the format shown in Message Format)	Message structure (defined in <code><atm/qcctypes.h></code>)

F.1.3.1 Call Setup

To make a call, the end user sends a SETUP message down to Q.93B and waits for a SETUP_ACK from Q.93B. The SETUP message should include a Broadband Higher Layer Information (BHLL) information element that contains a four-octet SAP identified as User Specific Information. The SAP is used to identify which

application on the receiving host the Q.93B should direct the message to. After receiving a SETUP_ACK with a 0 error field, the end user waits for either a CALL_PROCEEDING, ALERTING, CONNECT, or RELEASE_COMPLETE message from Q.93B (all other messages are ignored by Q.93B). After the CONNECT message is received, the end user can use the virtual channel.

The end user responds to a SETUP message from Q.93B, with either a CALL_PROCEEDING, ALERTING, CONNECT, or RELEASE_COMPLETE message to Q.93B. After the CONNECT_ACK message is received, the end user can use the virtual channel.

F.1.3.2 Release Procedure

To clear an active call or a call in progress, the end user should send a RELEASE message down to Q.93B and wait for a RELEASE_COMPLETE from Q.93B. Any time the user receives a RELEASE_COMPLETE message from Q.93B, the user releases the virtual channel if the call is active or in progress.

Q.93B never sends a RELEASE message to the end user; it will always send a RELEASE_COMPLETE. The end user only sends the RELEASE_COMPLETE message when rejecting a call in response to a SETUP message from Q.93B. At any other time, to reject or tear down a call, the end user sends a RELEASE message to Q.93B.

F.1.3.3 Exception Conditions

If for any reason Q.93B cannot process a SETUP message received from an end user, the SETUP_ACK is returned with an error value set, and call setup is not continued. The error value will be one of the cause codes specified in the ATM Forum UNI standard.

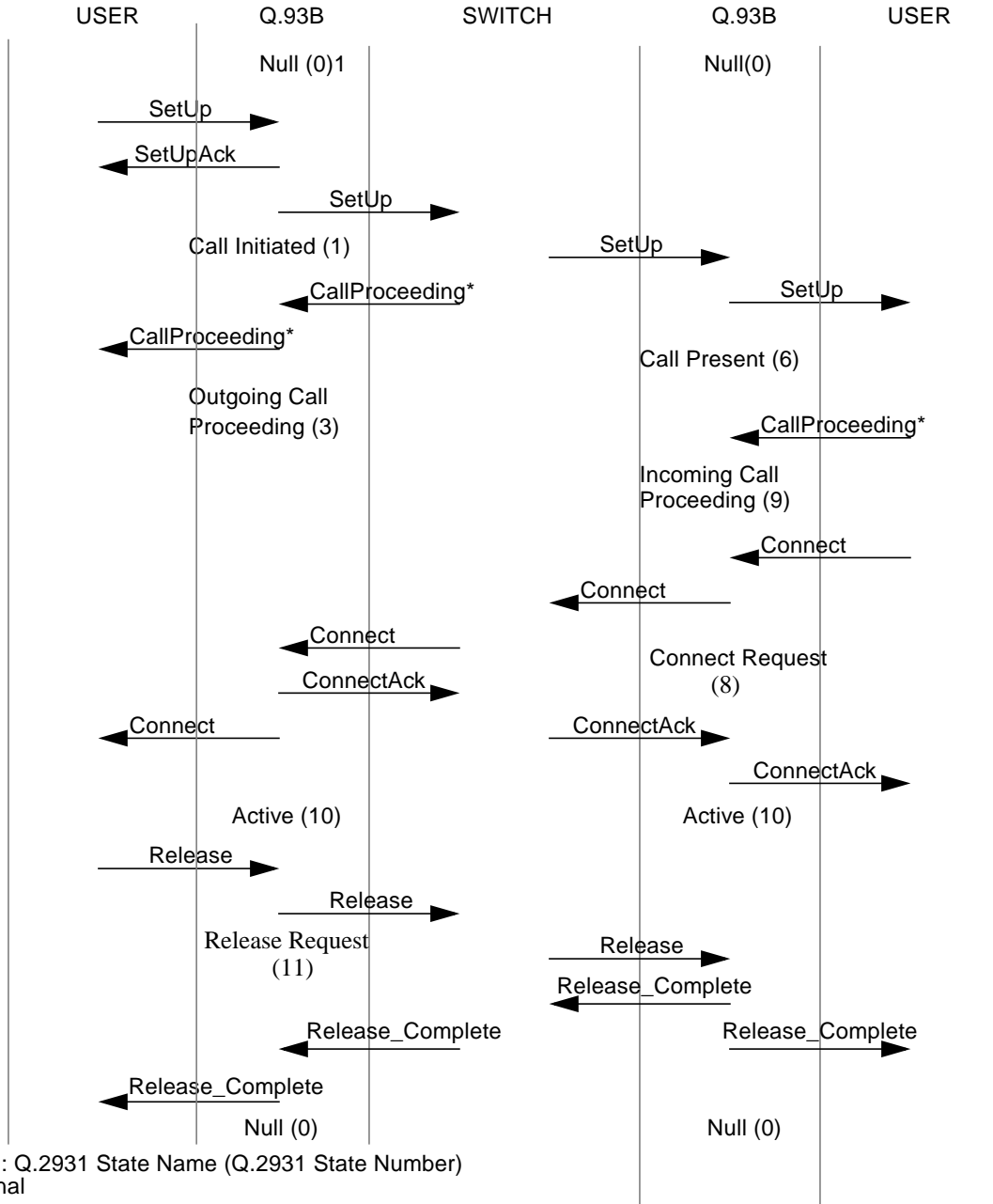


FIGURE F-3 Message Flow for Normal Call Setup and Tear Down

F.2 Connecting, Sending, and Receiving Data with the ATM Device Driver

Connecting to the ATM device driver involves several steps, some of which include several `ioctl` calls. To create a more standardized interface for user space applications, a set of `atm_util` functions is available to application writers. An overview of those functions is provided in TABLE F-4. For more detailed information, refer to the `atm_util(3)` man page. The `ba(7)` man page contains a more detailed discussion of the driver-supported `IOCTLS`.

TABLE F-4 `atm_util` Function Overview

Name	Functionality	Kernel Equivalent
<code>atm_open</code>	Opens a stream to the ATM device driver	Must be done by a user space setup program
<code>atm_close</code>	Closes a stream to the ATM device driver	Must be done by a user space setup program
<code>atm_attach</code>	Attaches to a physical interface	Must be done by a user space setup program
<code>atm_detach</code>	Detaches from a physical interface	Must be done by a user space setup program
<code>atm_bind</code>	Binds to a Service Access Point	<code>send DL_BIND_REQ</code>
<code>atm_unbind</code>	Unbinds from a Service Access Point	<code>send DL_UNBIND_REQ</code>
<code>atm_setraw</code>	Sets the encapsulation mode to raw	Send <code>DLIOCRAW</code>
<code>atm_add_vpci</code>	Associates a <code>vpci</code> with this connection	<code>A_ADDVC ioctl</code>
<code>atm_delete_vpci</code>	Dissociates a <code>vpci</code> from this connection	<code>A_DELVC ioctl</code>
<code>atm_allocate_bw</code>	Allocates constant bit rate bandwidth for this connection	<code>A_ALLOCBW ioctl</code>
<code>atm_allocate_cbr_bw</code>	Allocates constant bit rate bandwidth with more granularity than <code>atm_allocate_bw</code>	<code>A_ALLOCBW_CBR ioctl</code>
<code>atm_allocate_vbr_bw</code>	Allocates variable bit rate bandwidth	<code>A_ALLOCBW_VBR ioctl</code>
<code>atm_release_bw</code>	Releases previously allocated bandwidth	<code>A_RELSE_BW ioctl</code>

Note – The following discussion uses user space function names. Refer to TABLE F-4 for the corresponding kernel space function or `ioctl`.

To establish a data path, the application must first open the ATM driver and attach to a specific physical interface using `atm_open()` and `atm_attach()`. Next, the connection is associated with one or more VC(s), using `atm_add_vpci()`. If a call has been established using Q.2931 signalling, the `vpci` provided to `atm_add_vpci()` is the `vpci` that was included in the Q.2931 signalling messages received while establishing the call.

An encapsulation method must also be selected. The method of encapsulation is selected when the VC is associated with a stream (the `atm_add_vpci()` call). Currently, null and LLC encapsulation are supported. Null encapsulation implies that a message consists only of data preceded by a four-byte `vpci`. This type of encapsulation is most commonly used with raw mode. LLC encapsulation implies that an LLC header precedes the data. This header includes the SAP associated with the application's stream (using `atm_bind()`).

The end user can also select a mode of operation to determine the format of the message blocks passed to the ATM device driver. DLPI mode is set by default; however, the user can select raw mode with a call to `atm_setraw()`. DLPI mode implies that two or more mblocks will be sent to the driver. The first, which corresponds to the `ctl` buffer sent in the `putmsg()` system call, contains the `dlpi` message type, which is `dl_unitdata_req` for transmit and `dl_unitdata_ind` for receive. The `vpci` is included in this mblock as well. The `dl_unitdata_req` and `dl_unitdata_ind` header formats are defined in the header file `<sys/dlpi.h>`. The second and subsequent mblocks (corresponding to the data buffer in the `putmsg()` system call) contain the message.

F.2.1 Raw Mode Connections

Raw mode implies that the four-byte `vpci` is sent in the first mblock (corresponding to the `ctl` buffer in the `putmsg()` system call) followed by data. Any subsequent mblocks (the data buffer in the `putmsg()` system call) contain only data.

F.2.2 DLPI Mode Connections

Although the encapsulation and operational mode can be chosen independently, typically, DLPI mode is used for LLC-encapsulated traffic and raw mode is used for null encapsulation.

The driver's handling of packets depends on both the encapsulation method and the operational mode. For LLC-encapsulated traffic running in `dlpi` mode, the driver automatically adds the LLC header on transmit and strips the LLC header on receive before sending the message up the `dlpi` mode stream. In raw mode, however, the

driver does not modify the packets at all. This includes any header included with the packet. Thus, an application using raw mode and LLC encapsulation must include its own headers on transmit and receives data with the LLC header intact.

Received packets are directed to application streams based on the type of encapsulation in use. If a packet is null-encapsulated, it is sent up the stream associated with the vpci on which the packet was received. If a packet is LLC-encapsulated, it is sent to the stream that has bound (using `atm_bind()`) the SAP found in the LLC header.

SunVTS Validation and Test Suite

The `nettest` diagnostic test checks all the networking interfaces on a system, including SunATM adapters.

To use the `nettest` diagnostic, you must have the SunVTS™ Validation and Test Suite installed on your system. The SunVTS software runs diagnostic tests on Sun systems.

Note – You must have Classical IP up and running on an interface for the `nettest` to work.

Refer to the *SunVTS User's Guide* for more information on how to run the `nettest` diagnostic.

G.1 ATM Adapter Test (`atmtest`)

`atmtest` checks the functionality of the SunATM-155 and SunATM-622 SBus and PCI adapters.

It runs only in loopback (external or internal) mode. The Asynchronous Transfer Mode (ATM) adapter and ATM device driver must be present. To run the `atmtest` in external loopback mode, attach a loopback connector to the ATM adapter. The internal loopback mode does not require a loopback connector.

`atmtest` uses DLPI RAW mode to talk to the device driver. It establishes a virtual circuit (VC) to send, receive, and compare messages. If the message does not match, or the message is out of sequence, it displays an error message.

Using a random number generator, `atmtest` sends data into a data buffer and then sends each message from a different starting point. This assures that no two consecutive messages are the same.

`atmtest` can test more than one virtual circuit. Each additional virtual circuit increases the stress level of the test. `atmtest` automatically selects the virtual circuit number that is unique to the test. `atmtest` is nonscalable while it provides for more than one virtual circuit to be tested for each instance, which is adequate for the purpose of a stress test.

You can select `atmtest` only when the Intervention mode is enabled, because the test requires a loopback connector for external loopback testing. While Intervention mode is enabled, `atmtest` and `nettest` are both available as default selections. However, you must deselect `nettest` when running `atmtest`.



Caution – Do not run `nettest` while running `atmtest`.

Note – Bring the ATM interface down to make sure that the interface is in offline mode before running `atmtest`.

Note – Running `atmtest` for extended periods of time (i.e., over a day) will cause the system to panic. This is the result of a known OS problem (see BugID 4102546) against Solaris 2.6 and 7. This panic occurs when `ddi_dma_mem_free()` is called on a little-endian page whose virtual address is equal to or greater than `0x80000000`.

G.1.1 `atmtest` Test Modes

`atmtest` can only be run in offline mode. It is assumed that the host is not on a live network while it is testing in loopback mode.

G.1.2 `atmtest` Command Line Syntax

```
/opt/SUNWvts/bin/atmtest standard_arguments -o  
dev=device,tpkts=n,nv=n,m1=n,bw=n,opkts=n,ld,sd,sl,nc,ns,vcf
```

TABLE G-1 `atmtest` Command Line Syntax

Argument	Description
<code>dev=device</code>	Specifies the device name to be tested, such as <code>ba0</code> .
<code>tpkts=n</code>	Specifies the number of packets to loopback.
<code>nv=num_vc</code>	Specifies the number of simultaneous virtual circuits to be tested.

TABLE G-1 `atmtest` Command Line Syntax

Argument	Description
<i>ml=max_len</i>	Specifies the maximum length of the random packet.
<i>bw=bandwidth</i>	Specifies the bandwidth in MBits/s of each virtual circuit.
<i>opkts=n</i>	Specifies the number of packets for each virtual circuit that can be transmitted without receiving a corresponding packet.
<i>lb</i>	The internal loopback mode is selected.
<i>sd</i>	Changes the payload data to static from random.
<i>sl</i>	Changes all packets to their maximum length.
<i>nc</i>	Instructs the test not to check the receive payload (Improves throughput).
<i>ns</i>	Instructs the test not to exit on a packet reception failure.
<i>vcf=n</i>	Specifies the first virtual circuit number used.

G.1.3 `atmtest` Options

TABLE G-2 `atmtest` Options

Options	Description
Configuration	The Post Address, Host ID, and Domain Name of the system being tested.
Total Packets	The total number of packets sent. The default number of packets sent is 10000.
Number of VC	The number of virtual circuits to be set up by one instance. The default number of virtual circuits is 2 for each instance. The <code>atmtest</code> uses these two virtual circuits to send out messages simultaneously. The messages are received in the order they are sent.
Loopback	Enables the user to select either the external loopback field or internal loopback field. The default selection is the external loopback field. A loopback connector is only needed for external loopback testing.
MAX_PKG_LEN	The maximum packet length used by the test to send out the data. The default number is 9140.

TABLE G-2 atmttest Options

Options	Description
Outstanding_pkts	Describes the maximum number of outstanding packets. atmttest stops sending messages when the outstanding packet count is more than the number of packets this field specifies.
First_VC_no	Enables the user to set up the starting virtual circuit number to be used for each atmttest instance.
Bandwidth	Enables the user to select different bandwidths to test per VC. The default number is 14.

Index

SYMBOLS

? wildcard, 6-8, 6-16

A

a configuration flag, 6-6, 6-14, 6-15, 7-4

aarconfig file

 editing, 6-4

 file flags, 6-5

 flag options, 6-6

 sample configurations, 6-10

 using variables, 6-7

aarconfig file, 7-2, 7-3, 7-4

aarsetup program

 error messages, D-15

aarsetup program, 6-4, 7-2, 7-3, 7-4

allocating bandwidth, F-10

anymac variable, 6-8, 6-16

anymacsel variable, 6-8, 6-16

API, F-1

 allocating bandwidth, F-10

 atm_util functions, F-9

 CBR allocation, F-9

 device driver

 connecting, F-9

 receiving data, F-9

 sending data, F-9

 DLPI encapsulated connections, F-10

 message formats, F-5

 q93b and device drivers, F-2

 raw mode connections, F-10

 VBR allocation, F-10

Application Programmers Interface

See API

ARP address resolution tables, 7-4

ATM

 address, 5-13, 5-17, 6-4, 6-14, 7-2, 7-4

 aarconfig field, 6-4

 laneconfig field, 6-14

 registration, 7-2

 resolution, 7-3

 ARP address resolution tables, 7-4

 ARP server, 5-11, 5-12, 6-3, 6-5, 6-10, 6-12, 7-3, 7-4

 address, 5-14

 caching, 7-5

 M_PROTO mblock fields, F-5

 q93b driver, F-3

 router, 7-3

 switch, 1-4, 1-6, 1-9, 1-10, 2-4, 2-5, 2-9, 7-2

 switched virtual circuit, F-3

ATM Address field, 6-4, 6-7, 6-14

atm_util functions, F-9

atmadmin program

 Classical IP parameter group menu, 5-11

 common commands, 5-4

 ILMI parameter group menu, 5-11

 interface configuration menu, 5-6

 main menu, 5-4

 parameters, 5-7

 physical layer parameter group menu, 5-9

 signalling parameter group menu, 5-10

 starting, 5-3

 system parameter group menu, 5-5

 using, 5-3 to 5-18

atmconfig file

 editing, 6-1 to 6-3

example, 6-3

atmf.mib MIB file, E-2

atmf.mib.oid file, E-2

atmf.mib.schema file, E-2

atmreg program, 7-2

atmsnmpd SNMP daemon, E-1, E-3

atmstat command, D-2

atmtest

failure, D-12

atmtest, SunATM adapter test

command line syntax, G-2

B

ba device, 2-3, 2-9, 6-3, 6-7

boot PROM, 1-3

broadcast and unknown address server, 7-6

broadcast messages, 7-5

C

c configuration flag, 6-15

caching, 7-5

Call_ID message, F-5

Call_Tag message, F-5

CBR, F-9

checking

installation of a package, 4-5

the network, 4-7

CIP_Host field, 6-2

Classical IP, 5-11, 6-3, 7-1, 7-3, 7-4

configuring, 5-11, 6-3 to 6-12

no broadcast support, 5-11

sample configurations, 6-10

troubleshooting, D-3

configuration variables

in the aarconfig file, 6-7

rules, 6-9

setting, 6-7

constant bit rate, F-9

D

diagnostics

atmtest failure, D-12

DLPI encapsulated connections, F-10

E

ejecting the CD-ROM, 4-4

emulated LAN name, 5-18

end system identifier field, 7-2

Error Messages, D-12 to D-18

error messages

aarsetup, D-15

kernel drivers, D-18

lanesetup, D-15

Error_Code message, F-5

examining network interfaces, 4-7

extender plate, 2-1

F

Flag field, 6-14

framing interface

SDH, 5-9

setting, 5-9

SONET, 5-9

H

hardware

SunATM/P 155

installation, 3-1 to 3-2

requirements, 1-8

specifications, C-6 to C-8

supported platforms, 1-9

verifying the installation, 3-2

SunATM/P 622

installation, 3-1 to 3-2

requirements, 1-10

specifications, C-8 to C-9

supported systems, 1-11

verifying the installation, 3-2

SunATM/S 155

installation, 2-2

requirements, 1-4

specifications, C-1 to C-2

supported platforms, 1-4

verifying the installation, 2-3

SunATM/S 622

installation, 2-5 to 2-10

requirements, 1-6

specifications, C-4 to C-5

supported systems, 1-6

- verifying the installation, 2-8
- hostname, 5-16
- Hostname field, 6-4

I

- ifconfig command, 4-7, 7-1
- Ifname message, F-5
- ILMI service interface, 5-11, 7-2
- ilmid daemon, 7-2, D-2
- increasing performance, D-10
- installation patches
 - required, 4-2
- Interface field, 6-2, 6-4, 6-13
- IP hostname, 5-12
- IP to ATM resolution, 7-3

K

- kernel drivers
 - error messages, D-18

L

- L configuration flag, 6-5, 6-6
- l configuration flag, 6-5, 6-6, 6-14, 6-15
- LAN Emulation, 6-12, 7-1, 7-5
 - configuration server, 5-17, 7-6
 - configuring interface, 5-15 to 5-18, 6-12
 - connections, 7-7
 - driver, 7-5, 7-7
 - instance number, 6-2
 - IP address to an ATM connection, 5-18, 7-6
 - multiple Emulated LANs, 5-18
 - sample configurations, 6-17
 - server, 5-17, 7-6
 - troubleshooting, D-5
- lane# interface, 5-18
- lane# interface, 5-15, 6-3, 6-13
- lane.mib MIB file, E-2
- lane.mib.oid file, E-2
- lane.mib.schema file, E-2
- LANE_Host field, 6-2
- LANE_Instance field, 6-2
- laneconfig file
 - editing, 6-13

- entry descriptions, 6-13
- flag descriptions, 6-14
- using variables, 6-16
- laneconfig file, 7-2
- lanesetup program
 - error messages, D-15
- lanesetup program, 6-13, 7-2
- localswitch_server variable, 5-14, 6-8, 6-9

M

- M configuration flag, 6-6, 6-7, 6-14, 6-15
- m configuration flag, 6-6, 6-15
- MAC address, 5-13, 6-8, 6-9, 6-13, 6-16, 7-2, 7-5, 7-6, 7-7
- MAC Address/Emulated LAN field, 6-13
- mac variable, 5-13, 6-8, 6-16
- macsel variable, 5-13, 6-8, 6-16
- man pages, 4-4
- myaddress variable, 5-14, 6-8, 6-16

N

- n configuration flag, 6-14, 6-15
- netstat command, 4-7, 7-1
- nettest diagnostic, G-1
- network prefix, 7-2
- NFS version 3
 - decrease buffer sizes, D-11
 - overriding defaults, D-10
 - performance, D-10
 - Solaris 2.5.1, D-11
 - Solaris 2.6, D-10

O

- OBP Version, 1-3
- OpenBoot PROM, 1-3

P

- parameters
 - TCP/IP, D-10
- patches
 - required, 4-2
- PCI tests

- atmtest, G-1
- permanent virtual circuit, 5-14
- permanent virtual circuits, F-1
- ping command
 - troubleshooting, D-4, D-7
- ping command, 4-7, D-4
- pkgadd
 - adding software packages, 4-3
- pkgchk
 - checking package installation, 4-5
- pkginfo
 - checking package installation, 4-5
 - finding packages, 4-1
- pkgrm
 - removing packages, 4-1, 4-6
- prefix variable, 5-13, 6-8, 6-16
- PVC, F-1

Q

- Q.2931, 7-1, F-1
- Q.93B, F-1

R

- reconfiguration boot, 4-6
- removing a package, 4-6
- removing older software packages, 4-1
- requirements
 - SunATM/P 155, 1-8
 - software, 1-9
 - SunATM/P 622
 - hardware, 1-10
 - software, 1-11
 - SunATM/S 155, 1-4
 - software, 1-4
 - SunATM/S 622
 - software, 1-6
- RJ45 connector, 2-4

S

- s configuration flag, 6-5, 6-7, 6-15
- S00sunatm boot script
 - error messages, D-13
- SBus tests

- atmtest, G-1
- SC receptacle, 2-4, 2-9
- SDH, 5-9
- sel variable, 5-13, 6-8, 6-16
- selector field, 7-2
- show-devs command, 2-3, 2-9
- SNMP
 - installing the software, E-1
 - setting agent status, 5-5
 - setting up agent systems, E-2
 - setting up management console, E-1
- software
 - checking the network, 4-7
 - configuration, 5-3 to 5-18
 - troubleshooting, D-1
 - installation, 4-3 to 4-5
- SONET, 5-9
- specifications, C-1 to C-9
- SunATM adapter test, atmtest, G-1
- SunATM software
 - CD-ROM
 - ejecting, 4-4
 - mounting, 4-3
 - checking the network, 4-7
 - configuration, 5-3 to 5-18
 - troubleshooting, D-1
 - variables, 5-13, 6-8
 - installation, 4-3 to 4-5
 - predefined variables, 6-16
- SunATM/P 155
 - environmental specifications, C-8
 - features, 1-8
 - hardware requirements, 1-8
 - performance specifications, C-6
 - physical dimensions, C-6
 - power specifications, C-6
 - specifications, C-6 to C-8
 - supported platforms, 1-9
- SunATM/P 155 MMF
 - connecting to network, 3-4
 - illustrated, 1-7
 - introduction, 1-7
 - wiring configuration, 3-3
- SunATM/P 155 UTP5
 - connecting to network, 3-5
 - illustrated, 1-8
 - introduction, 1-7
 - wiring configuration, 3-4
- SunATM/P 622

- connecting to network, 3-5
 - environmental specifications, C-9
 - features, 1-10
 - hardware requirements, 1-10
 - illustrated, 1-10
 - introduction, 1-9
 - performance specifications, C-8
 - physical dimensions, C-8
 - power specifications, C-9
 - specifications, C-8 to C-9
 - wiring configuration, 3-5
 - SunATM/S 155
 - environmental specifications, C-2
 - extender plate, 2-1
 - features, 1-3
 - hardware requirements, 1-4
 - installation, 2-1 to 2-2
 - performance specifications, C-2
 - physical dimensions, C-1
 - power specifications, C-2
 - requirements, 1-4
 - supported platforms, 1-4
 - SunATM/S 155 MMF
 - connecting to network, 2-4
 - illustrated, 1-2
 - introduction, 1-2
 - SC receptacle, 2-4
 - wiring configuration, 2-4
 - SunATM/S 155 UTP5
 - connecting to network, 2-5
 - illustrated, 1-3
 - introduction, 1-2
 - RJ45 connector, 2-4
 - wiring configuration, 2-4
 - SunATM/S 622
 - connecting to network, 2-9
 - environmental specifications, C-5
 - features, 1-5
 - hardware requirements, 1-6
 - illustrated, 1-5
 - installation, 2-5 to 2-10
 - introduction, 1-5
 - performance specifications, C-4
 - physical dimensions, C-4
 - power specifications, C-4
 - SC receptacle, 2-9
 - specifications, C-4 to C-5
 - supported platforms, 1-6
 - supported systems, 1-6
 - wiring configuration, 2-9
 - sunmacselN* variable, 5-14, 6-8, 6-9
 - SunVTS Validation and Test Suite, G-1
 - SUNWatm
 - device drivers package, 4-3
 - SUNWatma
 - interim API support package, 4-3, 4-4
 - SUNWatmu
 - man pages, 4-4
 - runtime support package, 4-3, 4-4
 - SVC, F-1
 - switched virtual circuits, F-1
 - system performance, D-10
- T**
- t* configuration flag, 6-5, 6-7, 6-14, 6-15
 - T568B pin designation, B-1
 - TCP/IP
 - setting highwater mark, D-12
 - TCP/IP parameters
 - increasing performance, D-10
 - troubleshooting, D-1 to D-8
 - atmtest failure, D-12
 - autoclient, D-12
 - dataless client, D-12
 - diskless client, D-12
 - snoop command, D-10
 - Type message, F-5
- U**
- UNI field, 6-2
 - UNI specification, 6-2, 7-1
 - UNI version, 5-10
- V**
- variable bit rate bandwidth, F-9
 - VBR, F-10
 - VCI field, 6-4, 6-14
 - virtual circuit identifier, 5-14, 6-4
- W**
- wrist strap

attaching, 2-2, 2-6
attaching, 2-2, 2-6