

4DLT Administrator's Guide

Document Number 007-1396-020

CONTRIBUTORS

Written by Pam Sogard
Edited by Nan Schweiger
Production and illustrations by Gloria Ackley
Engineering contributions by John Ng

© Copyright 1993, Silicon Graphics, Inc.—All rights reserved
© Copyright 1991, Meridian Technology Corporation—All rights reserved
© Copyright 1991, Bell Atlantic Integrated Systems, Inc.—All rights reserved

This document contains proprietary and confidential information of Silicon Graphics, Inc. The contents of this document may not be disclosed to third parties, copied, or duplicated in any form, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

This manual has been adapted from the *SessionNet Administrator's Guide* by Bell Atlantic Integrated Systems, Inc. This material may be changed without notice by Bell Atlantic Integrated Systems, Inc. or Silicon Graphics, Inc. Bell Atlantic Integrated Systems, Inc., is not responsible for any errors that may appear herein.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure of the technical data contained in this document by the Government is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013, and/or similar or successor clauses in the FAR, or the DOD or NASA FAR Supplement. Unpublished rights reserved under the Copyright Laws of the United States. Contractor / manufacturer is Silicon Graphics, Inc., 2011 N. Shoreline Blvd., Mountain View, CA 94039-7311.

Silicon Graphics, IRIS, and the Silicon Graphics logo are registered trademarks, and IRIX and 4DLT are trademarks of Silicon Graphics, Inc. DEC, DECnet, DECserver and LAT are trademarks of Digital Equipment Corporation. UNIX is a registered trademark of UNIX System Laboratories.

4DLT Administrator's Guide
Document Number 007-1396-020

Contents

Introduction	xi
Purpose of This Guide	xi
Organization of This Guide	xii
Typographical Conventions	xiii
Related Documentation	xiii
Product Support	xiv

1. Notes to New 4DLT Administrators	1
LAT Networks and 4DLT	1
LAT Network Fundamentals	2
LAT Terminal Functions	4
LAT Terminal Server Functions	4
LAT Service Node Functions	5
LAT Service Management	7
LAT Service Ratings	7
LAT Group Codes	8
LAT Network Architecture	8
The LAT Protocol Stack	9
LAT's Service Class Layer	10
LAT's Slot Layer	11
LAT's Virtual Circuit Layer	13
LAT's Data Link Layer	15
LAT's Physical Link Layer	15

- 2. **4DLT Software Fundamentals** 17
 - The LAT Driver—*/dev/lats* 18
 - The LAT Daemon—*latd* 18
 - Configuring the LAT Protocol Stack 19
 - Providing Initial Database Information 20
 - Setting Up Incoming Service Connections 20
 - Multicasting Service Announcements 20
 - The User Interface Utility—*slat* 21
 - The 4DLT Configuration Utility—*lati* 21
 - The 4DLT Maintenance Utility—*slcp* 21
 - How the LAT Database Works 22
 - The Permanent LAT Database 22
 - The Volatile LAT Database 23
 - Summary of LAT Database Events 25
 - The Services File and Setup Utility 26
 - How Printing Works 26

- 3. **4DLT Setup Procedures** 27
 - Preparing for 4DLT Setup 27
 - The Default 4DLT Configuration 28
 - The Default LAT Node Name 29
 - The Default Network Interface for LAT 29
 - The Default LAT Configuration File 30
 - The Default LAT Services File 30
 - Operating as a Dual Purpose LAT Node 32
 - Preparing User Environments for 4DLT 32
 - Standard 4DLT Startup Procedures 33
 - Starting 4DLT from the Command Line 33
 - Starting 4DLT from a File 34

Starting 4DLT with an Alternate LAN Interface	34
Displaying the LAT Network Interface	34
Using the Alternative Startup Command	35
Automatic Startup With an Alternative LAN Interface	35
Stopping and Restarting 4DLT	36
4. Performing Routine LAT Maintenance	37
About the <i>slcp</i> Utility	37
Displaying <i>slcp</i> Options	38
Guidelines for Entering <i>slcp</i> Commands	38
Monitoring LAT	39
Managing Overall LAT Node Functions	40
Changing the LAT Node Name or Description	40
Setting the CPU Rating and LAT User Limits	41
Setting a Maximum on LAT Login Sessions	42
Setting the LAT Service Announcement Timer	42
Setting a LAT Service Rating	43
Manual Binding to 4DLT Devices	43
Maintaining the LAT Services File	45
A Sample <i>latservice</i> File	46
Adding Logins to the <i>latservice</i> File	46
Removing Logins from the <i>latservice</i> File	47
Adding Printers to the <i>latservice</i> File	47
Removing Printers from the <i>latservice</i> File	48
Managing Group Codes	49
Assigning Group Codes with <i>slcp</i>	49
Automatic Group Code Assignments	50
Maintaining the LAT Database	51
Overwriting the Permanent Database	51
Purging Unavailable Nodes and Services	51
Deleting LAT Service Queue Entries	52

- 5. Reconfiguring LAT Operating Parameters 53**
 - The LAT Configuration Utility Menu 54
 - Selecting Menu Options 54
 - Putting Your Changes into Effect 54
 - Setting Ethernet Parameters 55
 - Setting LAT Node Identification Parameters 56
 - Setting LAT Configuration Limits 57
 - Setting Circuit Parameters 59
 - Setting Session Parameters 61
 - Setting Host-initiated Parameters 62
 - Setting Advertised Service Parameters 63
 - Setting Learned Service Parameters 63
 - Setting STREAMS Configuration Parameters 65
 - Summary of 4DLT Configuration Parameters 67

- Glossary 71**

- Index 77**

Figures

Figure 1-1	LAT Functions on a 4DLT Node	2
Figure 1-2	A Simple LAT Network	3
Figure 1-3	Host-initiated Request	7
Figure 1-4	LAT Protocols Compared to the DNA and OSI Models	9
Figure 1-5	LAT Layers and Layer Data	10
Figure 1-6	Multiplexing Slot Data into a Virtual Circuit Message	12
Figure 1-7	Virtual Circuit Message Components	14
Figure 2-1	4DLT Software Components	18
Figure 2-2	The STREAMS Stack for 4DLT	19
Figure 2-3	LAT Volatile Database Components	24

Tables

Table 2-1	LAT Database Events	25
Table 5-1	Summary of 4DLT Configuration Parameters	67

Introduction

4DLT™ is a Silicon Graphics implementation of the Local Area Transport (LAT™) protocol, a network protocol originally developed by Digital Equipment Corporation (DEC™) to optimize communications between terminal users and network hosts. 4DLT enables Silicon Graphics computers to participate in a LAT network.

4DLT software performs LAT terminal server functions in a Silicon Graphics computer, allowing a user to connect to the LAT network from his or her workstation. 4DLT software also enables a Silicon Graphics system to supply its resources to other LAT network users.

Purpose of This Guide

The *4DLT Administrator's Guide* explains how to configure and maintain 4DLT software on a Silicon Graphics workstation. The audience for the *4DLT Administrator's Guide* is system and network administrators who are responsible for implementing and maintaining LAT services on a Silicon Graphics node. The procedures in this guide require the following proficiencies:

- Familiarity with LAT network features and user functions
- Experience in managing LAN resources
- Familiarity with the IRIX™ operating system
- Facility with a UNIX editor
- Knowledge of network protocols (for STREAMS configuration only)

Organization of This Guide

The *4DLT Administrator's Guide* is organized into these chapters:

Chapter 1, "Notes to New 4DLT Administrators"

Introduces concepts and terms that apply to LAT networks. Read this chapter for an overview of LAT network theory and how it applies to the 4DLT application.

Chapter 2, "4DLT Software Fundamentals"

Explains the components of 4DLT that are of primary interest to the administrator. Read this chapter to learn about 4DLT processes, files, utilities, and the organization and function of the LAT database. This chapter also explains how 4DLT implements LAT services.

Chapter 3, "4DLT Setup Procedures"

Describes the as-shipped 4DLT configuration and the parameters you might need to change before starting the 4DLT application. The chapter also contains procedures for starting and stopping 4DLT.

Chapter 4, "Performing Routine LAT Maintenance"

Describes the commands available to monitor and modify 4DLT and provides procedures and examples for using 4DLT commands and utilities. It describes the organization of the LAT services file and explains how to maintain it.

Chapter 5, "Reconfiguring LAT Operating Parameters"

Explains 4DLT internal parameters and how to adjust them to suit the needs of a particular LAT network or individual 4DLT node.

Glossary

Index

Typographical Conventions

This document uses the standard UNIX[®] convention when referring to entries in IRIX manual pages. The entry name is followed by a section number in parentheses. For example, *cc(1)* refers to the *cc* manual entry in Section 1 of the IRIX manual pages

In text descriptions, file and directory names, as well as commands, are shown in *italics*. Command syntax descriptions follow these conventions:

`typewriter font`

Indicates portions of the command that are required in a entry.

italic font

Indicates a variable argument. Substitute a valid string or value for these variables.

[*brackets*]

Indicates that the argument shown in brackets is optional. You can omit these arguments in a command entry.

`typewriter bold`

Indicates a user entry. Type these entries, or portions of entries, exactly as shown.

<Enter>

Indicates a key legend on the workstation keyboard.

Related Documentation

In addition to the *Administrator's Guide*, 4DLT documentation consists of the *4DLT Release Notes*, which contain technical notes and installation instructions for 4DLT, and the *4DLT User's Guide*, which contains instructions for accessing LAT resources from a Silicon Graphics workstation.

For additional introductory information on LAT, see *LAT Network Concepts*, published by Digital Equipment Corporation.

Product Support

Silicon Graphics provides a comprehensive product support and maintenance program for its products. For further information, contact your Silicon Graphics service provider.

Notes to New 4DLT Administrators

This chapter explains the function of 4DLT in a LAT network and acquaints readers with some of the concepts and terms that apply to LAT networks. The chapter contains these sections:

- “LAT Networks and 4DLT”
- “LAT Network Fundamentals”
- “LAT Service Management”
- “LAT Network Architecture”
- “The LAT Protocol Stack”

LAT Networks and 4DLT

Local Area Transport (LAT) is a communications protocol, originally developed by Digital Equipment Corporation, that specializes in communications between a terminal server and multiple hosts on a local area network (LAN). The purpose of LAT is to increase the resources available to terminal users while reducing the burden that terminal communications normally impose on the reciprocating host and LAN.

4DLT is Silicon Graphics' implementation of the LAT protocol for IRIS computers. An IRIS system running 4DLT can operate as either a general purpose or a dual purpose LAT node. As a *general purpose node*, the IRIS system contributes its resources to the LAT network; as a *dual purpose node*, the IRIS system contributes local resources and also handles requests from its own users to access LAT services.

Figure 1-1 illustrates the LAT functions that 4DLT performs.

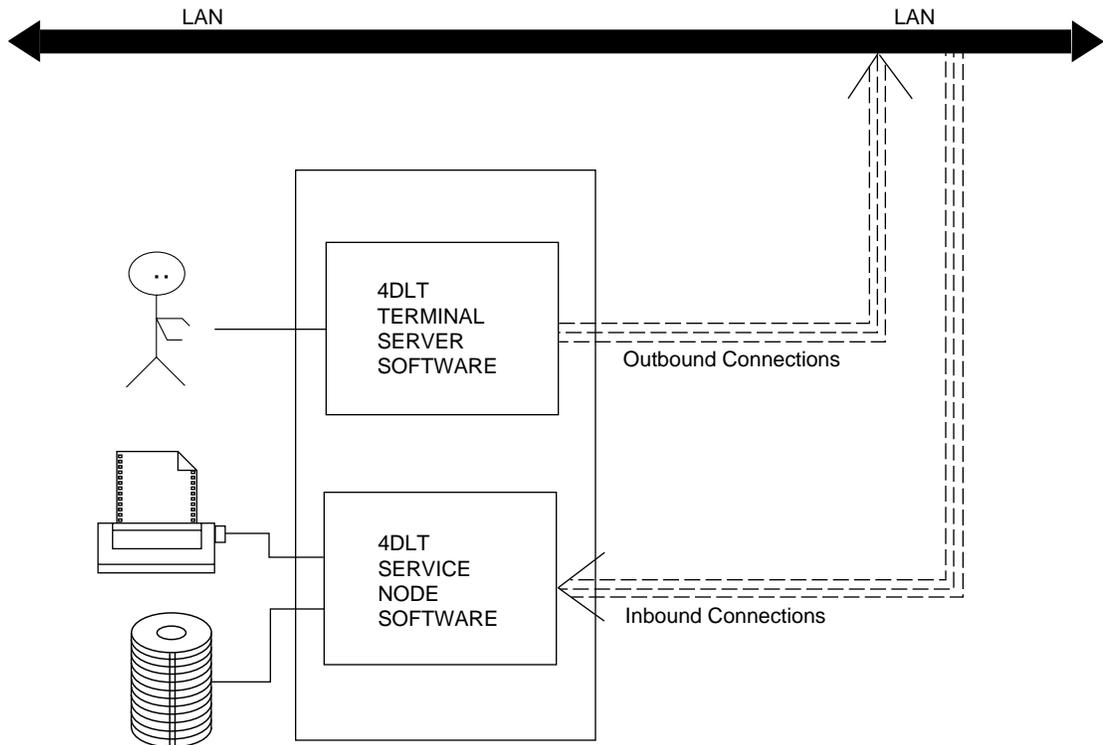


Figure 1-1 LAT Functions on a 4DLT Node

LAT Network Fundamentals

A LAT network consists of user terminals, terminal servers, and service nodes, attached to a LAN, that communicate by means of LAT software. The purpose of the LAT network is to offer services to terminal users. *Services* are resources, such as a login facility or remote printer, that are known to users and network software by a unique name. A name translation process converts service names into an address that is usable by LAN software.

A LAT network also includes printers, modems, and other devices made available to users by means of LAT software. However, only those systems running LAT software are considered *LAT nodes*. LAT nodes can also

function concurrently as DECnet™ and Transport Control Protocol, Internet Protocol (TCP/IP) nodes.

Figure 1-2 illustrates a simple LAT network configuration.

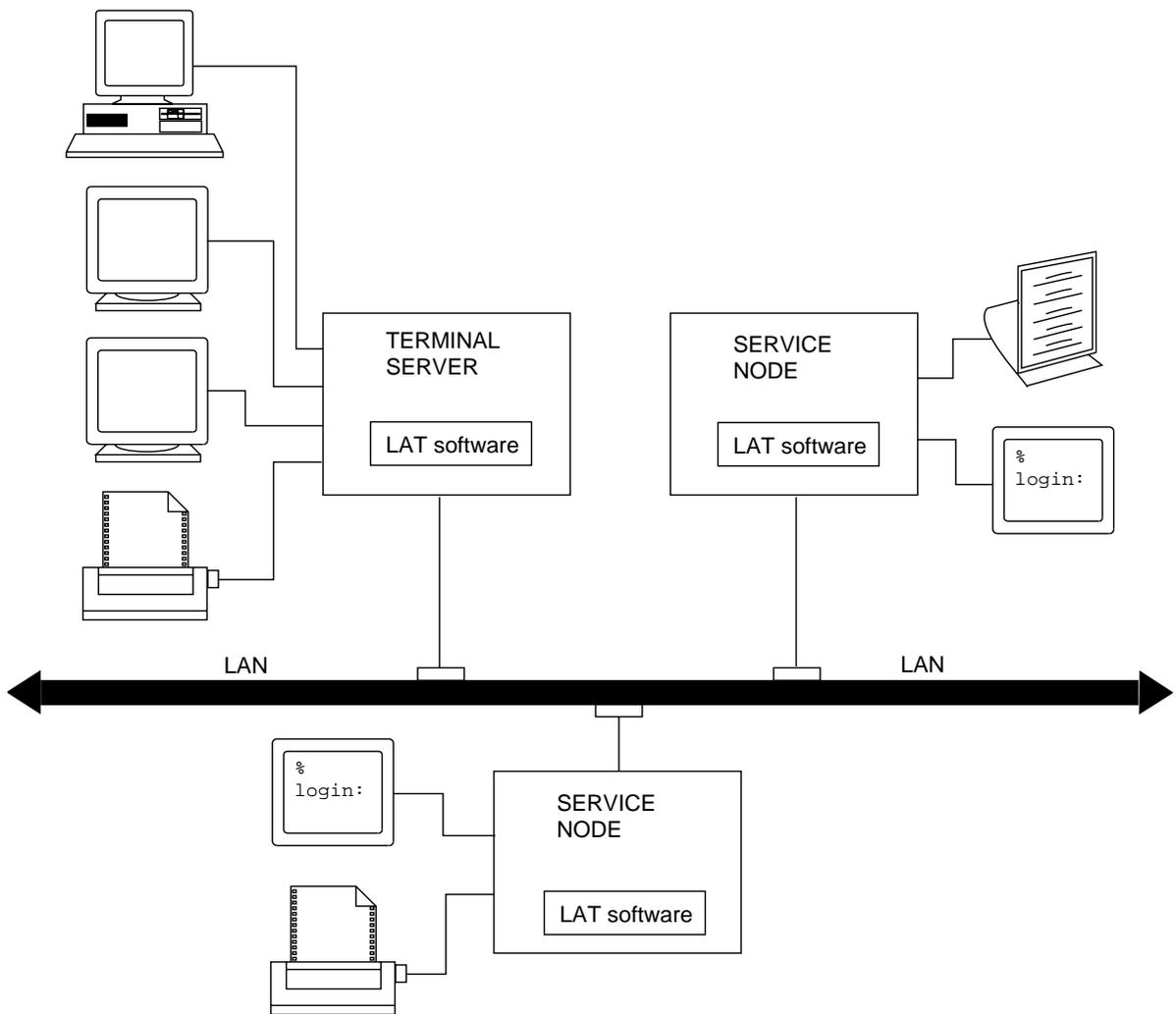


Figure 1-2 A Simple LAT Network

LAT Terminal Functions

LAT terminals are the interactive devices on the LAT network where users request services and conduct LAT sessions. The device can be a dumb terminal, such as a VT100, or it might be a personal computer running terminal emulation software. 4DLT allows an IRIS workstation user to establish a terminal-type connection to LAT hosts.

LAT User-initiated Requests

To request services, users enter a *connect* command at their terminals. The command is processed by the terminal server, which handles the request by establishing a connection to a LAT node offering the service. Requests that originate with a *connect* command on a user terminal are known as *user-initiated requests* (see "Host-initiated Requests" on page 6 for details).

LAT Sessions

The period of time during which a user remains connected to a service is known as a *session*. A user can establish several sessions from a single terminal and switch between sessions, but only one session may be active at a time. Connections to the inactive sessions are maintained so that users can resume a session at any time.

LAT Terminal Server Functions

A *terminal server* is a LAT node that establishes connections to LAT services and manages communication between the user and the service provider. A terminal server may be a specialized device, such as a DECserver™ 90L, or a host running terminal server software. 4DLT software provides terminal server functions to an IRIS workstation user, allowing the user to connect to LAT network resources.

The LAT Database

An important function of a terminal server is to maintain the *LAT database*, a body of information about LAT operating parameters and the nodes and services available to LAT users. A terminal server keeps its database current by collecting information from service announcements and connection events (see “LAT Service Node Functions” on page 5), and by recording information supplied by the LAT administrator.

LAT Terminal Connections

Notice from Figure 1-2 that user terminals are directly attached to the terminal server, and that the terminal server manages communications to all terminals over a single line. During LAT sessions, terminal server software performs the multiplexing required to support this line configuration. Terminal servers also perform the character interrupt processing normally done by the host in terminal-host transactions, which conserves host resources.

LAT Printer Connections

Figure 1-2 shows that printers, as well as terminals, can be directly attached to a terminal server. As this figure suggests, a terminal server not only requests LAT services but can also supply them. When a printer is attached to a terminal server, the terminal server establishes a connection to the printer only at the request of a LAT host (see “Host-initiated Requests” on page 6 for details). During the ensuing session, an application on the requesting host (not the terminal server) controls the printer.

LAT Service Node Functions

A *service node* is a host on a LAT network that offers one or more services to terminal users. 4DLT enables an IRIS computer to function as a service node on a LAT network. Connections between a service node and a user terminal are always through the LAT terminal server to which the user is attached, but to the LAT user, this connection appears direct.

LAT Service Announcements

To make its services known to the LAT network, a service node advertises its services in *service announcements*. A service announcement contains the name and address of the node sending the announcement, the name and description of each service the node provides, and a *service rating*—a measure of the node’s ability to provide the service efficiently. Service announcements can also contain group codes if the network uses them (see “LAT Group Codes” on page 8).

A service node broadcasts service announcements every 60 seconds by *multicasting*; that is, the service node sends announcements to a multicast address that designates LAT terminal servers. Terminal servers collect service announcements and enter announcement information in the LAT database.

Host-initiated Requests

In addition to responding to user-initiated requests, service nodes also generate requests for services. Requests generated by a LAT service node are known as *host-initiated requests*. In a host-initiated request, a service node asks a terminal server to establish a session between one of the server’s ports (which is configured for an application) and an application on the requesting node. Typically, LAT print services, including the 4DLT print service, are provided by means of a host-initiated request.

Figure 1-3 illustrates a host-initiated request.

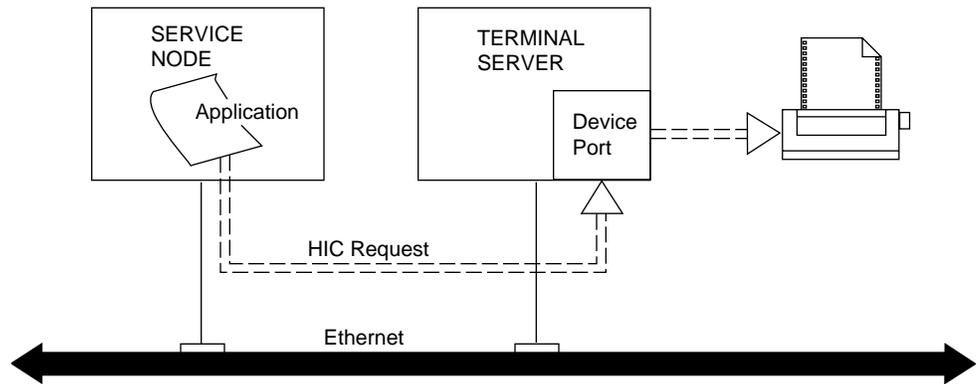


Figure 1-3 Host-initiated Request

LAT Service Management

In LAT networks, one or several nodes may be offering the same service. When more than one LAT node offers a particular service, terminal server software selects the node that will provide the service most efficiently. To evaluate efficiency, terminal server software uses rating information it maintains in the LAT database.

LAT Service Ratings

The LAT service directory maintained by terminal servers includes a *service rating*, a measure of the availability of a node to provide the service it is advertising. Service nodes include rating information in their service announcements. When a service is provided by multiple LAT service nodes, terminal servers select the provider node offering the highest service rating.

Dynamic service ratings are assigned by the service node providing a service, based on the node's activity level, its available memory, and its processor type. *Static service ratings* are assigned by the LAT administrator and remain constant in all service announcements. Static service ratings provide a means of weighting the chances that a particular host will be selected by a terminal server as the provider node.

LAT Group Codes

By default, LAT users have access to every service on a LAT network. However, a LAT administrator may choose to use *group codes*, numbers that may be assigned to service nodes, individual services, terminal servers, and terminal server ports to manage access to LAT resources. Group code values range from 0 to 255. They are included in service announcements.

When group codes appear in a service announcement, terminal servers check the codes for a match with their own group codes. If a group code in the announcement matches a group code assigned to the terminal server, the terminal server enters the service information in its database. If the group code does not match, the terminal server ignores the service announcement.

LAT Network Architecture

Most public and private networks in widespread use are implemented on layered architectures that parallel the OSI reference model. Common examples include the TCP/IP suite, widely used for UNIX networking; and Digital Network Architecture (DNA), Digital Equipment Corporation's proprietary networking architecture.

LAT architecture also conforms to a layered network model. Figure 1-4 shows the LAT protocol layers that are peer-to-peer with the OSI and DNA models.

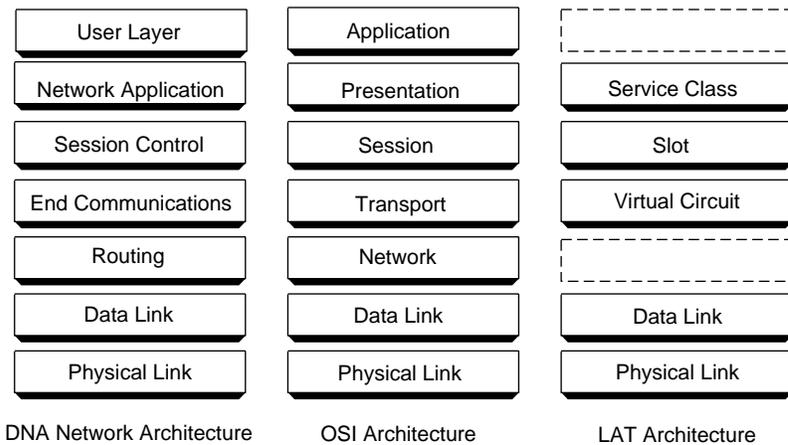


Figure 1-4 LAT Protocols Compared to the DNA and OSI Models

An important difference between LAT architecture and the OSI and DNA reference models is the lack of a Network layer. As a consequence, no cross-network routing is possible on a LAT network; a LAT network is constrained to a single LAN.

Note: The DNA network layer, IP, and the LAT network can coexist on a single Ethernet LAN.

The LAT Protocol Stack

The LAT protocol stack consists of three LAT layers—Virtual Circuit, Slot, and Service Class, and two LAN layers—Physical Link and Data Link. This multilayer architecture supports peer-to-peer communications between terminal servers and service nodes. The LAT protocol assumes that all of its terminal servers and service nodes are on the same LAN.

Figure 1-5 illustrates LAT architecture and the processing of outbound data that occurs in each layer.

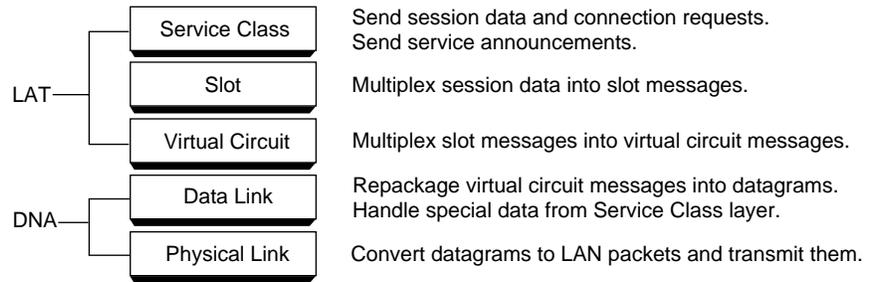


Figure 1-5 LAT Layers and Layer Data

LAT's Service Class Layer

The Service Class layer is the uppermost layer of the LAT protocol. On a service node, the Service Class layer offers services, multicasts service announcements to terminal servers (see "LAT Service Node Functions" on page 5), and assigns service ratings (see "LAT Service Ratings" on page 7).

On a terminal server, the Service Class layer makes service requests and maintains the database of LAT nodes and services (see "LAT Terminal Server Functions" on page 4). The Service Class layer on a terminal server also performs flow controls of session data moving between the terminal server and devices attached to it.

The LAT Multicast Timer

To control multicasting on the service node, the Service Class layer maintains a *multicast timer*. The timer regulates the intervals at which service announcements are sent. Multicast intervals are specified in seconds; the default interval is 60 seconds, but intervals can be adjusted by the LAT administrator.

Special Service Class Messages

The Service Class layer uses the services of the Slot and Virtual Circuit layers for transmitting session data, but it sends some special data directly to the Data Link layer. This special data includes service announcements, queueing requests to the server's connection queue, and status information about queued requests.

LAT's Slot Layer

The Slot layer establishes sessions, multiplexes session data, and manages internode flow control. The Slot layer also terminates sessions at the request of the Service Class layer.

Slot Layer Multiplexing

The Slot layer places information for a given session into *slots*, message segments containing several characters of data belonging to an individual user. The maximum size of a slot is 255 bytes, but slot sizes can vary for each side of a connection. For example, a typist at a terminal would not need as large a slot size as a host application that may send full screens of data.

In addition to containing user data slots (*Data-A* slots), slots can also contain session control data, such as breaks, parity, and framing errors (*Data-B* slots). Some slots contain out-of-band data (*Attention* slots), while others contain session start and stop requests (*Start* slots and *Stop* slots).

At regular intervals, the Slot layer places accumulated slots for all sessions with a particular service node into a single Virtual Circuit message, multiplexing the data sent from the terminal server to a particular host. This multiplexing makes efficient use of LAN bandwidth by increasing the ratio of data to packet information and reducing the number of required frames.

Figure 1-6 illustrates the multiplexing performed by the Slot layer.

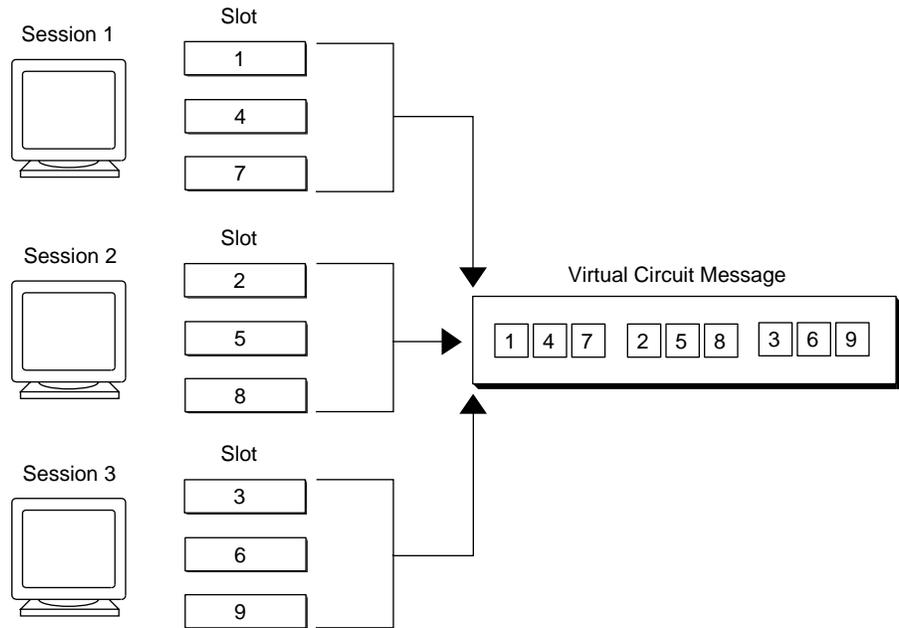


Figure 1-6 Multiplexing Slot Data into a Virtual Circuit Message

Slot Layer Load Balancing

The Slot layer also provides load balancing using service ratings supplied in service announcements. The value range for service ratings is 0 to 255. When hosts offer multiple services with the same name, the Slot layer establishes a session with the host that has the highest rating.

Slot Layer Flow Control

The Slot layer provides internode flow control by means of a credit system. A *credit* is a marker that the Slot layer allocates each receive buffer it creates for a session (typically, the Slot layer allocates two receive buffers, but this number is negotiated between the nodes at the start of the session).

During a session, credits are passed with each buffer exchanged between the session partners. When either partner's credit count is decremented to zero, partner with no credits stops sending data until it receives more credits. This prevents either node from transferring more buffers of data than the reciprocating node is able to accept.

LAT's Virtual Circuit Layer

The Virtual Circuit layer is the LAT transport mechanism. It establishes and maintains a logical communications path between host nodes and terminal servers. This path is called a *virtual circuit*, and it is bidirectional, sequential, and error-free.

A virtual circuit is established as soon as one message is exchanged between a terminal server and a particular service node. The virtual circuit is used for all subsequent sessions between the terminal server and that service node. During LAT sessions, the Virtual Circuit layer multiplexes and demultiplexes Slot layer data, performing error detection and correction on data transmissions.

Virtual Circuit Messages

The Virtual Circuit layer in terminal servers and service nodes send *virtual circuit messages* to transmit session data and virtual circuit information. There are three types of virtual circuit message: a *start message* is sent by terminal servers to open new virtual circuits; *run messages*, which contain slots and credits from the Slot layer, are exchanged by both partners during a session; and, when all sessions are ended, a *stop message* is sent by the terminal server to close the virtual circuit.

Figure 1-7 illustrates the components of a virtual circuit run message.

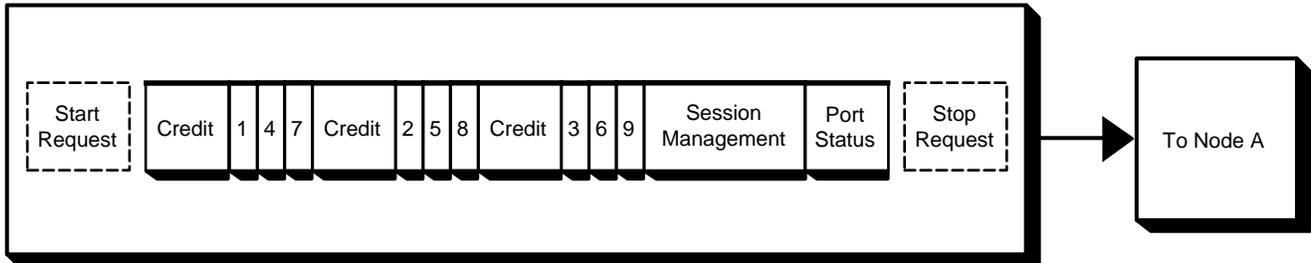


Figure 1-7 Virtual Circuit Message Components

Virtual Circuit Keep-alive Messages

When it has no data to send, a virtual circuit goes into *balanced mode*; that is, the virtual circuit is maintained by exchanging empty run messages, so either side can resume transmission when necessary. In the normal *unbalanced mode*, only the terminal server initiates the data exchange. In balanced mode, a *keep-alive* message is sent every 20 seconds, but no other data is sent. The advantage of the keep-alive message is that the virtual circuit can be maintained even after the circuit timer expires.

The Virtual Circuit Layer Circuit Timer

The Virtual Circuit layer on the terminal server maintains a *circuit timer* that transmits virtual circuit messages at specified intervals (typically every 80 milliseconds). When it receives a message, the service node returns an acknowledgment to the terminal server. Assuming the acknowledgment is received, the terminal server sends another virtual circuit message the next time the circuit timer expires.

In addition to timing the dispatch of virtual circuit messages, the circuit timer in the Virtual Circuit layer also measures response intervals. If a host does not acknowledge receipt of a virtual circuit packet within interval set on the circuit timer, the terminal server assumes that a problem exists. After a specified wait interval and a set number of retries, the terminal server assumes that the host is unreachable.

LAT's Data Link Layer

In a LAT network, connectivity is provided for Data Link peers. The Data Link layer is a general architectural layer that implements a communications facility on top of the Physical Link layer. The Data Link layer places outgoing virtual circuit messages into packets called *datagrams* and removes incoming messages from incoming datagrams.

LAT's Physical Link Layer

The Physical Link layer of LAT provides a specific medium, such as Ethernet, for transmitting the packets. The Physical Link layer also insulates the Data Link layer from any characteristics of the physical medium.

4DLT Software Fundamentals

4DLT software is composed of the LAT driver, */dev/lats*; the LAT daemon, *latd*; a user interface utility, *slat*; a configuration utility, *lati*; and two maintenance utilities, and *slcp* and *svc_setup*. These components rely on the LAT database and on the underlying LAN Data Link layer to provide LAT functions.

This chapter explains the functions of 4DLT components and describes the structure of the LAT database. The chapter contains these sections:

- “The LAT Driver—*/dev/lats*”
- “The LAT Daemon—*latd*”
- “The User Interface Utility—*slat*”
- “The 4DLT Configuration Utility—*lati*”
- “The 4DLT Maintenance Utility—*slcp*”
- “How the LAT Database Works”
- “Summary of LAT Database Events”
- “The Services File and Setup Utility”
- “How Printing Works”

Figure 2-1 illustrates the components of 4DLT software.

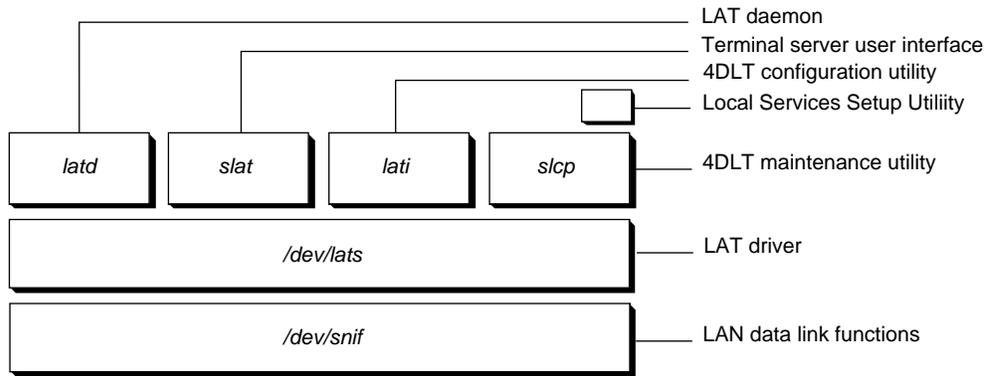


Figure 2-1 4DLT Software Components

The LAT Driver—*/dev/lats*

The LAT driver, */dev/lats*, is a cloneable STREAMS device that supports AT&T's Transport Provider Interface. This driver implementation allows application developers to write Transport layer interface (TLI) programs that use LAT as a transport agent.

The LAT Daemon—*latd*

The LAT daemon, *latd*, performs functions that are fundamental to IRIS system membership in a LAT network:

- Configuring the LAT protocol stack
- Providing LAT database information to the IRIX kernel
- Setting up incoming connections to LAT services
- Multicasting service announcements to other LAT nodes

The *latd* daemon is normally invoked during system startup, along with other network daemons, and runs as a background process. You can also invoke *latd* from the command line of an active IRIX window.

Configuring the LAT Protocol Stack

The *latd* daemon configures the LAT protocol stack using information that it reads in the */usr/etc/lat/latconfig* file (see “The Permanent LAT Database” on page 22 for information on the *latconfig* file). The stack is implemented as configurable STREAMS modules.

Figure 2-2 illustrates the default configuration of the 4DLT protocol stack.

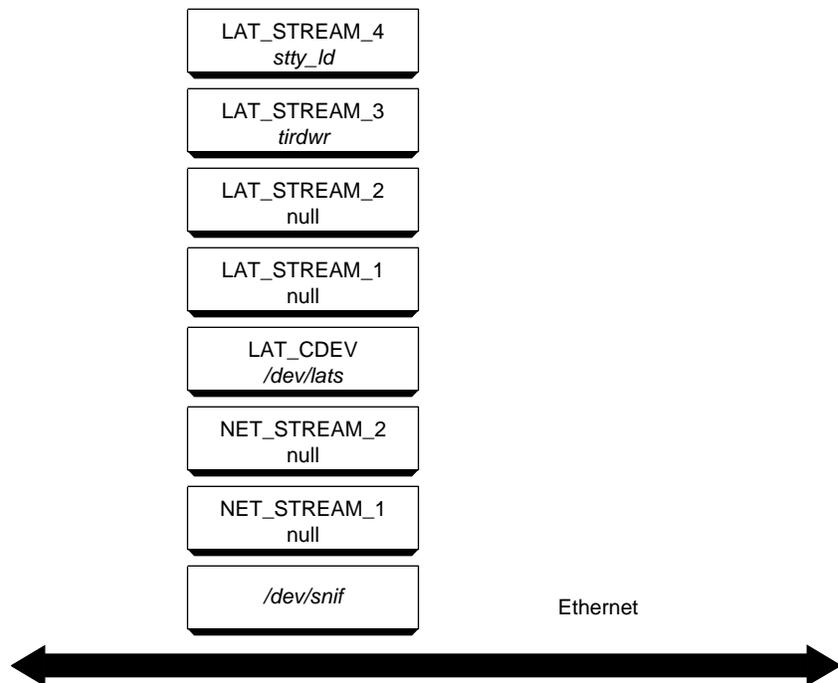


Figure 2-2 The STREAMS Stack for 4DLT

In the default configuration, */dev/snif* is the lowest STREAMS module in the stack, immediately above the Ethernet. With the exception of */dev/snif*, individual modules can be configured to customize data link and session level information processing. LAT_STREAM_1 through LAT_STREAM_4 modules pertain to login sessions only; these modules have no effect on printing services.

Providing Initial Database Information

When it first starts, the LAT daemon (*latd*) reads a copy of */usr/etc/lat/latconfig* and reports its contents to the IRIX kernel. The IRIX kernel uses *latconfig* as the nucleus of a working database. The kernel supplements this database with information from incoming service announcements as they arrive.

Setting Up Incoming Service Connections

When a LAT user or LAT node requests a service such as a login session or printer, the IRIX kernel hands the request to the *latd* daemon. Then *latd* allocates a pseudo-terminal or spooler process and executes the requested service. If the requested service is a login session, *latd* reads and writes characters to and from the pseudo-terminals during the session.

Multicasting Service Announcements

The *latd* daemon multicasts service announcements to other LAT nodes to advertise the LAT services that the local IRIS system offers. When *latd* is started, it reads the */usr/etc/lat/latservice* file for a list of the login and print services it should multicast. A default version of *latservice* is created when you install 4DLT software on an IRIS host (see “The Default LAT Services File” on page 30).

The User Interface Utility—*slat*

The *slat* utility is a terminal server emulation that allows an IRIS system user to connect to LAT network services and manage one or more sessions with LAT service nodes. The *4DLT User's Guide* describes *slat* features in detail and explains how they are used.

In addition to providing the user interface to LAT services, *slat* also reports outgoing connection requests from local users to the IRIX kernel. If an outgoing connection to a LAT node fails, the IRIX kernel marks the node and its services "unreachable." By default, unreachable nodes remain in the database for five hours, but the retention time is adjustable ("Setting Learned Service Parameters" on page 63 gives details).

The 4DLT Configuration Utility—*lati*

The LAT configuration utility, *lati*, is a database editor that is used to set values in the permanent 4DLT database. The permanent database is stored in a non-ASCII file called */usr/etc/lat/latconfig* (see "How the LAT Database Works" on page 22).

The values that are set in */usr/etc/lat/latconfig* are put into effect when *latd* is started. For this reason, you must stop and restart *latd* whenever you run *lati* to change *latconfig*.

The 4DLT Maintenance Utility—*slcp*

The *slcp* utility is used to display information about the LAT network, manage LAT functions on the IRIS system, and modify the LAT database for day-to-day operation. Typically, a LAT administrator uses *slcp* to perform these tasks:

- Adding and delete services
- Changing service and node identification information
- Setting service and CPU ratings

- Setting limits on LAT sessions
- Overwriting values in the */usr/etc/lat/latconfig* file

Any Silicon Graphics system user can display LAT information with *slcp*. But *slcp* commands that modify LAT functions, such as those required to perform the tasks listed above, require superuser privileges—you must perform these functions as the *root* user.

The changes you make with *slcp* become effective as soon as the *slcp* command completes. By default, new settings are recorded in the volatile database (see “The Volatile LAT Database” on page 23) and remain in effect until you either reset them or stop *latd*.

Note: Some parameters you set with *slcp* can also be written to the */usr/etc/lat/latconfig* file. See “Overwriting the Permanent Database” on page 51 for details.

How the LAT Database Works

4DLT maintains two databases: a *permanent database*, which is stored in the */usr/etc/lat/latconfig* file, and a *volatile database*, which includes a memory-resident copy of the permanent database and additional LAT information that the IRIX kernel records during 4DLT operation.

The Permanent LAT Database

The permanent database for 4DLT operation is contained in the */usr/etc/lat/latconfig* file. When you first install 4DLT, the permanent database contains a default version of *latconfig*. The information in *latconfig* is read by *latd*, *slcp*, and *slat*.

The *latconfig* file stores four types of parameter that control LAT operations on the IRIS system:

- The name of the IRIS node and its LAN interface
- Limits on sessions and circuits supported by the IRIS system

- Timers for incoming and outgoing connection events
- LAT protocol stack specifications

The Default Permanent Database

Default operating parameters for 4DLT are already set in the as-shipped copy of */usr/etc/lat/latconfig*. In most LAT environments, the default settings are suitable for optimum 4DLT operation, but in some cases, the default values will need to be changed to suit your LAT network. Chapter 5, “Reconfiguring LAT Operating Parameters,” explains how to change the *latconfig* file.

Overriding Permanent Database Values

The *lati* utility is the primary means of changing the *latconfig* file. However, you can override some values in *latconfig* by issuing *slcp* commands. The new values take effect immediately, but they are not recorded in the permanent database unless you use a special *slcp* command to overwrite *latconfig* settings (see “Overwriting the Permanent Database” on page 51).

The Volatile LAT Database

The volatile database consists of a copy of the permanent database and information that 4DLT software accumulates as it operates. The volatile database is stored in main memory—when *latd* is stopped, the volatile database is erased.

Data in the volatile database comes from four sources:

- Information copied from the permanent database at *latd* startup
- Information arriving from service announcements
- Information that the LAT administrator enters using *slcp*
- Information from failed connection attempts

Accumulated data is stored in the *services directory*, which contains data pertaining to LAT services, and the *service node directory*, which contains data pertaining to LAT nodes. Although they are referred to as directories, the services directory and service node directory are not part of the IRIX filesystem; rather, they are collections of data stored in memory on the IRIS host.

Figure 2-3 shows the components of the volatile database.

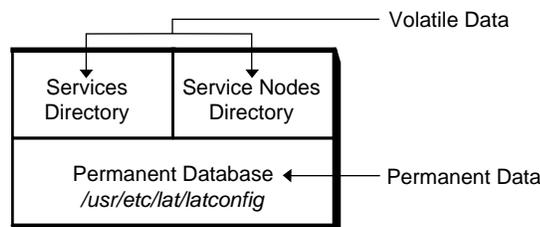


Figure 2-3 LAT Volatile Database Components

The Services Directory

The services directory on an IRIS system contains this information about each LAT service available to the local IRIS system user:

- The service name and description
- The name and rating of the node offering each service
- The current status of the service and its group codes

Incoming service information is referred to as *learned services*; learned services are dropped from the database when 4DLT is shut down. If a remote node does not refresh its service information within a specified time, the IRIX kernel marks the services from that node “unknown.” The unknown status is displayed in various listings to LAT users. By default, unknown nodes remain in the database for five hours, but the retention time is adjustable (“Setting Host-initiated Parameters” on page 62 gives details).

The Service Node Directory

The service node directory on an IRIS system contains this information about each LAT node whose service announcements the IRIS system is collecting:

- The service node name and description
- The address of the service node
- The availability of the node for connections

Summary of LAT Database Events

Table 2-1 summarizes the events in the lifecycle of the LAT database.

Table 2-1 LAT Database Events

Event	Action Taken
<i>latd</i> reads <i>/usr/etc/lat/latconfig</i> when it starts.	<i>latd</i> creates the volatile database by copying <i>latconfig</i> into IRIX kernel memory.
<i>latd</i> reads incoming service announcements and other network data.	<i>latd</i> supplements the volatile database with the new information.
Administrator changes database values with the <i>slcp</i> utility.	<i>slcp</i> updates the volatile database and puts new values into effect.
Administrator overwrites <i>/usr/etc/lat/latconfig</i> with <i>slcp</i> .	<i>slcp</i> changes the permanent database to reflect the current volatile database.
Connection requests to remote nodes fail.	IRIX kernel updates the volatile database.
<i>slat</i> user enters <i>show services</i> command.	<i>slat</i> posts a list of LAT services on the screen.
Administrator changes database values with the <i>lati</i> utility.	<i>lati</i> changes the permanent database. New values have no effect until <i>latd</i> is stopped and restarted.
<i>latd</i> is terminated.	Volatile database is erased from memory.

The Services File and Setup Utility

The LAT services file, */usr/etc/lat/latservice*, contains a record of each local LAT service that is to be established as part of the 4DLT startup process. A default copy of *latservice* is created during 4DLT installation. You add to and remove services from *latservice* using the */usr/etc/lat/svc_setup* utility.

The purpose of *latservice* is to provide a permanent record of local service information, since the services directory is erased from memory during a 4DLT shutdown. Any local service that is not recorded in *latservice* must be recreated when 4DLT resumes operation if it is to be available to LAT users. Remote services are added to *latservice* as service announcements arrive.

When *latd* starts, it reads *latservice* and establishes each service that the file contains. *latservice* can contain two types of service:

- Login services that the IRIS node offers LAT users
- LAT printers that the local IRIS system user can request

How Printing Works

LAT printer services are set up by the 4DLT administrator using the *svc_setup* utility (details are given in “Adding Printers to the latservice File” on page 47). These printers are known to the 4DLT user by a printer name. When the user specifies the printer name in a print command (usually *lp*), the printer application opens the device (an IRIX */dev* file) that is associated with the remote printer as if the printer were local. However, opening the device executes 4DLT functions that send the file over the LAT network to a remote printer.

The 4DLT installation process creates 256 devices for LAT, */dev/tty1* through */dev/tty255*. The *tty1* device is reserved for internal LAT administration. Incoming session connections are assigned tty numbers that start at *tty1* and increment with additional sessions; outgoing connection requests are assigned tty numbers that start at *tty255* and decrement with additional sessions. For this reason, when you assign devices to LAT printers during the printer setup process (described in “Adding Printers to the latservice File” on page 47), you should assign the printer a tty number in the middle of the 1 to 255 range.

4DLT Setup Procedures

This chapter explains what to do to configure the 4DLT application after you install it. It explains the default LAT configuration that results from 4DLT software installation and explains how to modify and add to the default configuration. This chapter also contains instructions for preparing the user environment for 4DLT, and instructions for starting and stopping 4DLT.

The chapter includes these sections:

- “Preparing for 4DLT Setup”
- “The Default 4DLT Configuration”
- “Preparing User Environments for 4DLT”
- “Standard 4DLT Startup Procedures”
- “Starting 4DLT with an Alternate LAN Interface”
- “Stopping and Restarting 4DLT”

Preparing for 4DLT Setup

Before you perform 4DLT setup procedures, 4DLT software must be installed on your IRIS system. If you have not already installed 4DLT software, see your *4DLT Release Notes* for specific product installation information. Your *4DLT Release Notes* also recommend additional documentation on product installation.

After 4DLT software is installed, you might want to modify the IRIX initialization file for the root user to include the pathnames of 4DLT software. During initialization, IRIX sets a path to directories that are listed in the initialization file; this enables you (as the root user) to execute 4DLT commands without specifying the entire pathname of each command in your entries.

4DLT commands are stored in two directories:

- */usr/bin/lat*
- */usr/etc/lat*

If you use the C shell (the IRIX system default), search paths are set in the *.cshrc* file in *root*'s home directory. Add the 4DLT directories as arguments to the *set path* command in the *.cshrc* file.

If you use the Bourne or Korn shell, search paths are set in the *.profile* file in *root*'s home directory. Include 4DLT directories as arguments to the *setenv PATH* command in the *.profile* file.

The Default 4DLT Configuration

The 4DLT installation process creates a default LAT configuration on the IRIS system. To define a default configuration, 4DLT installation performs these setup tasks:

- Assigning the LAT node name to the IRIS system
- Assigning the LAN interface to the LAT network
- Creating a default version of the 4DLT configuration file, */usr/etc/lat/latconfig*
- Creating a preliminary version of the LAT services file, */usr/etc/lat/latservice*

In many LAT networks, the default configuration that the 4DLT installation process creates is sufficient to begin running the IRIS system as a dual purpose LAT node. In other networks, however, the default configuration will require some modification.

To determine whether the default 4DLT configuration is suitable for your particular IRIS system and LAT network, review the information in the remainder of this section. If you determine that changes are needed, use the procedures in this chapter and in Chapter 4 and Chapter 5 to make the required modifications.

The Default LAT Node Name

The 4DLT installation process assigns the IRIS system a LAT node name by reading the system's */etc/sys_id* file and writing its contents to the */usr/etc/lat/latconfig* file. To determine the LAT node name assigned to your IRIS system, display the contents of */etc/sys_id* using the *hostname* (1) command:

```
# hostname <Enter>
```

If you prefer to use a LAT node name that is different from the name returned by *hostname*, you can change the LAT name with either the *slcp* utility (see “Changing the LAT Node Name or Description” on page 40) or the *lati* utility (see “Setting LAT Node Identification Parameters” on page 56). You must start 4DLT software before you use *slcp* (see “Standard 4DLT Startup Procedures” on page 33 for instructions).

Note: If your IRIS system is also a member of a TCP/IP or DECnet network, the name contained in */etc/sys_id* is usually the recognized name for your system on those networks as well. Although it is permissible to choose a different LAT node name, assigning the same node name for all networks makes system administration easier.

The Default Network Interface for LAT

The 4DLT installation process assigns the primary LAN interface on the IRIS system as the LAT network interface. By default, the primary LAN interface on your IRIS system is its built-in Ethernet interface, which is usually configured as either *ec0* or *et0* when the IRIS system is initialized. If you plan to use the primary LAN interface as the LAT network interface, you do not need to change the default network interface in order to run 4DLT.

If you are currently using your primary LAN interface for a network other than an Ethernet network (your primary interface is currently used for an FDDI or token ring network, for example), you must specify an alternate network interface during 4DLT startup. See “Starting 4DLT with an Alternate LAN Interface” on page 34 for details.

The Default LAT Configuration File

The LAT configuration file, */usr/etc/lat/latconfig* contains parameters that regulate LAT performance on the IRIS system. This file is read at 4DLT startup, and its contents remain in effect until 4DLT is stopped (or until you change *latconfig*). In most cases, the as-shipped version of *latconfig* contains the appropriate settings for an IRIS host to operate satisfactorily on a LAT network, but occasionally these parameters must be changed.

To verify that the default settings are satisfactory for your node and network, review the information in Table 5-1. If you determine that *latconfig* requires changing, start the 4DLT application (instructions are given in “Standard 4DLT Startup Procedures” on page 33) and follow the instructions in Chapter 5 to make your changes.

Note: If you are not the LAT network administrator, discuss any changes you plan with the administrator before you complete them.

The Default LAT Services File

The 4DLT installation process creates a default version of the */usr/etc/lat/latservice* file that contains one record. The record sets up a login service on the IRIS node. The name of the new login service is the IRIS host name as specified in the */etc/sys_id* file. The record also assigns a service rating and a description for the login service; it does not assign a password to the service.

For example, assume that you just installed 4DLT on an IRIS system that contains the host name *moonbeam* in its */etc/sys_id* file. After 4DLT installation, the default */usr/etc/lat/latservice* file would contain this record:

```
LOGN:moonbeam:-1:"Login service for host `moonbeam`":*:
```

Changing the Default LAT Services File

The */usr/etc/lat/latservice* file is an ASCII file that you can modify using your favorite editor. The list below explains each field shown in the sample default entry above and describes what you can do to change your default version of *latservice*.

Note: Do not change permissions on the *latservice* file. To ensure security, permissions on *latservice* are set to *read/write* for the root user only. If you change permissions on *latservice*, *latd* will not read this file and the services it contains will not be advertised.

<i>LOGN</i>	Is a keyword that specifies the type of service defined in this entry. Do not change <i>LOGN</i> ; it must appear as the first word in this entry.
<i>moonbeam</i>	Is the name of the login service to be advertised by this node. The service name was derived from the entry <i>moonbeam</i> in moonbeam's <i>/etc/sys_id</i> file. You can replace this service name with a different name if you wish.
-1	Is a service rating designation. This particular rating, negative 1, specifies that the service rating is to be dynamically derived by the <i>latd</i> daemon. See "Setting a LAT Service Rating" on page 43 for information about setting service ratings.
"LAT login service for host 'moonbeam'"	Is the description of the login service that is posted in various displays that list the service, including users' <i>CONNECT</i> and <i>SHOW SERVICES</i> commands. Notice that the description is enclosed in quotation marks. You can replace this description with a different one, but enclose the new description in quotation marks.
* (asterisk)	Is a placeholder to specify that there is no password on this service. You can replace the asterisk with a password if you wish.

Note: The asterisk must appear in any empty field of a *latservice* file record.

Adding to the Default LAT Services File

In addition to changing the default entry in */usr/etc/lat/latservice*, you can add more login services, password protect login services, and include LAT printer records in this file. Additional information about modifying */usr/etc/lat/latservice* is given in "Maintaining the LAT Services File" on page 45.

Operating as a Dual Purpose LAT Node

The 4DLT installation process assumes that you wish to operate the IRIS host as a dual purpose LAT node; therefore, it enables both service node and terminal server functions on the IRIS system. If you prefer, you can disable the terminal server functions on your system, which prevents the local IRIS system user from connecting to LAT services. To disable terminal server functions, you must modify `/usr/etc/lat/latconfig` using the `lati` utility (see "Setting Ethernet Parameters" on page 55 for instructions).

Preparing User Environments for 4DLT

When an IRIX user creates a new shell, the shell initialization procedure sets a path to the directories that will be searched to execute the user's commands. 4DLT commands must be included in this path to allow users to execute 4DLT commands without specifying the entire pathname for each command entered.

The procedure below explains how to modify the `.cshrc` file to include the 4DLT directory in the search path. You must do this procedure as either the root user or as the owner of this account.

Note: If you are preparing the environment for a Bourne or Korn shell user, modify the user's `.profile` file instead of the `.cshrc` file; add 4DLT directories as arguments to the `setenv PATH` command in the `.profile` file.

1. Change to the user's home directory:

```
# cd login_name <Enter>
```

2. Edit the `.cshrc` file in the home directory.

4DLT user commands are located in the `/usr/bin/lat` directory. To add this directory to the user's path, open the `.cshrc` file with an IRIX editor and add this line to the end of the path:

```
set path = ($path /usr/bin/lat)<Enter>
```

Note: If this `.cshrc` file already has a `set path` command in it, you can just add the `/usr/bin/lat` directory as an argument.

3. Close and save the *.cshrc* file.
4. Execute the *.cshrc* script:

```
# source .cshrc <Enter>
```

Once you complete this procedure, the user whose *.cshrc* file you edited will be able to execute 4DLT commands without specifying the 4DLT directory in each command entry. Repeat this procedure for other 4DLT users on this IRIS system.

Standard 4DLT Startup Procedures

Starting the LAT daemon starts 4DLT software on an IRIS system. You can start *latd* with a command line entry, or you can enable automatic startup by setting an initialization flag. The procedures below explain how to start 4DLT using each method.

Note: If your IRIS system is also running 4DDN, always start 4DDN before starting 4DLT software.

Starting 4DLT from the Command Line

The example below shows the command you use to start 4DLT from the command line. It assumes that you have included the */usr/bin/lat* directory in your search path. It also assumes that you plan to use the primary network interface for the LAT network (see "Using the Alternative Startup Command" on page 35 for details).

Notice that you must be logged in as root to execute this command:

```
# latd <Enter>
```

Once *latd* is started, 4DLT commands are ready to use and the IRIX system is ready to accept requests for outgoing and incoming LAT connections. Before turning the system over to LAT users, however, you should review the procedures in the remainder of this guide to determine whether additional setup procedures are necessary.

Starting 4DLT from a File

If you would like 4DLT to start automatically as part of the initialization process on the IRIS system, you can set a flag that will start *latd* each time the IRIS system is restarted. Enter this command to enable automatic startup:

```
# /etc/chkconfig 4DLT on <Enter>
```

If you decide to turn automatic startup off, enter this command:

```
# /etc/chkconfig 4DLT off <Enter>
```

Starting 4DLT with an Alternate LAN Interface

During the IRIX initialization process, the system's network interfaces are assigned numbers, based on the order in which they are enabled, that identify them to other software. The initialization process assigns the primary network interface a unit number of 0.

By default, 4DLT software uses unit 0 as its network interface. If you want 4DLT to use a network interface other than the primary interface, you can specify an alternate interface at 4DLT startup. During the 4DLT startup process, the */dev/snif* device is assigned to the designated network interface.

Displaying the LAT Network Interface

To display your IRIS system's network interfaces and their assigned unit numbers, use the *latmac* command:

```
# latmac <Enter>
```

Name	Unit
ec0	0
fv0	1

The example above shows that *ec0* is the primary network interface on this IRIS system, since it has a unit number of 0. Because unit 0 is assigned to the *ec0* LAN interface, 4DLT software will use *ec0* as its LAT network interface.

Using the Alternative Startup Command

If you do not plan to use the primary network interface on your IRIS system for the LAT network, you must specify a different unit number to 4DLT software in the startup command.

To specify a new unit number for 4DLT software, use the *latd -u* command:

```
# latd -u new_unit_number <Enter>
```

The argument to *latd -u* is the unit number of the network interface that you plan to use for the LAT network. For example, assume that a *latmac* command returns this information:

Name	Unit
ec0	0
ec1	1
enp0	2

To assign *enp0* as the LAT network interface, you would issue this startup command:

```
# latd -u 2 <Enter>
```

Automatic Startup With an Alternative LAN Interface

You can set up 4DLT to start automatically and specify an alternative LAN interface in the automatic startup process. To enable automatic startup, modify the */etc/rc2.d/S31lat* initialization file:

1. Open the *S31lat* file:

```
# vi /etc/rc2.d/S31lat
```

2. Modify the **OPTIONS** setting in *S31lat*, as shown below:

```
OPTIONS="-u unit_number"
```

3. Stop and restart 4DLT for your changes to take effect (see "Stopping and Restarting 4DLT" on page 36).

Stopping and Restarting 4DLT

Many 4DLT maintenance procedures require you to stop and restart the LAT daemon so that your changes will take effect. Use the commands below to stop and restart *latd*. Notice that you must complete this procedure as the *root* user of the IRIS system:

1. Stop *latd*:

```
# /etc/killall 15 latd <Enter>
```

2. Restart *latd*:

```
# latd <Enter>
```

Note: If you are using a non-primary network interface to the LAT network, specify the alternate network interface number when you restart 4DLT (see “Using the Alternative Startup Command” on page 35 for details).

Performing Routine LAT Maintenance

This chapter contains procedures that are needed to perform routine LAT maintenance on an IRIS node. It explains how to use *slcp*, a 4DLT maintenance utility, to add and remove LAT services and adjust LAT functions during day-to-day operation. The chapter contains these sections:

- “About the *slcp* Utility”
- “Monitoring LAT”
- “Managing Overall LAT Node Functions”
- “Maintaining the LAT Services File”
- “Managing Group Codes”
- “Maintaining the LAT Database”
- “Deleting LAT Service Queue Entries”

Note: Most *slcp* options are covered in this chapter, but a few *slcp* options are beyond the chapter’s scope. Please see the *slcp* man page for a complete explanation of *slcp* options.

About the *slcp* Utility

The *slcp* utility is a group of commands that allow LAT users and administrators to display 4DLT parameters and LAT administrators with superuser privileges to change 4DLT parameters. Any changes you make with *slcp* commands are recorded in the volatile database and take effect immediately. It is not necessary to stop and restart 4DLT to implement your changes.

Some parameters in the volatile database are also contained in the permanent database (see Table 5-1). When you change these parameters with *slcp*, you can incorporate your changes in the permanent database by using a special *slcp* command that overwrites */usr/etc/lat/latconfig* settings (see “Overwriting the Permanent Database” on page 51 for details).

Displaying *slcp* Options

To display a list of *slcp* command options, enter the *slcp* command with no arguments, as shown in this example:

```
# slcp <Enter>
slcp: usage
-d H|h|c|p|x|q|u[,#]|n[,node]|s[,service]
-b s,device,service[,“id”][,port][,password][,rating]
-b c,device,service[,node][,port][,password]
-b h,device[,service][,node][,port]
-B device|\* Delete touch device binding (\* delete all)
-c [cpu_rating][,user_limit] Set cpu rating, user limit
-f Save LAT configuration to /usr/etc/lat/latconfig
-m sessions Set maximum LAT sessions (0 no limit, 1-255)
-n [node][,“id”] Set local node name, description
-g [a|i,l]#[,#...][, #-#] Set ACL[a] or IDL[i] groups
(default both)
-G [a|i,l]#[,#...][, #-#] Clear ACL[a] or IDL[i] groups
(default both)
-Q sequence|\* Delete queue sequence # (\* delete all)
-r service,rating Set local service rating
-t seconds Set service announcement timer (10-180 sec)
-u service,unit Assign unit (minor #) to service
-U service,unit Remove unit (minor #) from service
-X Purge unknown, unreachable nodes, services
-Z [c][n[,node]] Zero host [c] and/or node [n] counters
```

Guidelines for Entering *slcp* Commands

An *slcp* command consists of the keyword *slcp* and a command switch. The switch is a letter preceded by a hyphen (-). For example, this is the *slcp* command to purge the LAT database of unavailable nodes and services:

```
slcp -X
```

Most *slcp* commands require at least one argument. For example, this *slcp* command to set the maximum number of sessions on a service node requires a maximum sessions argument:

```
slcp -m max_sessions
```

Some *slcp* commands take several arguments. The *slcp* commands with multiple arguments have this syntax:

```
slcp -switch argument[,argument][,argument][,argument]
```

When a command contains more than one argument, separate the arguments with a comma (,); include commas as space holders whenever you omit an argument from the arguments list. Do not include spaces between arguments. For example, this command assigns a service rating to a LAT service:

```
slcp -r IRIS2,250
```

Monitoring LAT

The *slcp -d* command posts information about the local LAT node and information about remote nodes and services in the LAT network. Any LAT user can use *slcp -d* to display information; superuser privileges are not required.

The syntax of *slcp -d* is shown below:

```
slcp -d h c p x q u[,port_number] n[,node_name] s[,service_name]
```

The arguments to *slcp -d* have these meanings:

<i>h</i>	Displays information about the local host.
<i>c</i>	Displays various counters for events on the local host.
<i>p</i>	Displays the contents of bad packets received.
<i>x</i>	Displays diagnostic information about the local host.
<i>q</i>	Displays information about queued requests.
<i>u, port_number</i>	Displays information about login processes. If you specify a port number, displays information about the process logged in at that port.

- n,node_name* Displays a list of known nodes and their availability. If you specify a node name, displays information about the node you specified.
- s,service_name* Displays a list of known services. If you specify a service name, displays the provider node and the status and ratings of its services.

Managing Overall LAT Node Functions

The *slcp* utility offers commands to set parameters that affect the overall operation of the IRIS system as a LAT network node. When you use the *slcp* commands described in this section, the settings you specify are recorded in the volatile database and remain in effect until *latd* is stopped. You can set equivalent parameters in the permanent database using *lati* (see Chapter 5 for instructions).

Changing the LAT Node Name or Description

By default, the 4DLT installation procedure assigns the IRIS system the host name that appears in the IRIS file */etc/sys_id*. You can change the default node name for the IRIS system if you wish. However, if your system is a member of multiple networks, it will be easier to maintain the system if it uses the same name on all networks.

Note: Changing the LAT node name on the IRIS host does not affect its name on other networks.

To change your IRIS system's LAT node name or description, use the *slcp -n* command:

```
# slcp -n new_name[,description] <Enter>
```

The arguments to *slcp -n* have these meanings:

<i>new_name</i>	<p>Is the LAT node name that you wish to assign the IRIS host (this name will not be used for the IRIS system on the TCP/IP or DECnet networks if it is also a member of those networks). Use these guidelines when assigning a LAT node name:</p> <ol style="list-style-type: none">1. A node name must be unique on the LAN.2. Node names can contain a maximum of 16 characters.3. Valid characters include all alphanumeric characters and the symbols dollar sign (\$), period (.), hyphen (-), and underscore (_).4. Uppercase and lowercase letters are valid, but lower case is converted to upper case for internal operations.
<i>description</i>	<p>Is an optional description of the LAT node that helps to identify it. If you include a description in your entry, the description appears in various displays of information about the node, including users' <i>SHOW HOST</i> and <i>SHOW NODES</i> commands. If you enter a multiple word description, enclose your entry with quotes, as shown in this example:</p> <pre>slcp -n lily, "engineering node for testing"</pre>

Setting the CPU Rating and LAT User Limits

LAT service nodes use their CPU rating as one factor in calculating dynamic service ratings. The *slcp -c* command lets you assign a fixed CPU rating to an IRIS node to weight its chances of being selected for a service. If you do not assign a CPU rating, a rating is assigned dynamically by 4DLT software.

The *slcp -c* command also lets you set a limit on the total number of LAT users concurrently using the IRIS system. This option allows you to manage the LAT load on the node by restricting user access (also see "Setting a Maximum on LAT Login Sessions" on page 42).

To set the CPU rating and user limit on an IRIS system, use the *slcp -c* command:

```
# slcp -c cpu_rating[,user_limit] <Enter>
```

Arguments to *slcp -c* have these meanings:

- cpu_rating* Is the static rating you wish to assign to the CPU. Valid CPU ratings range from 1 to 100: a high rating indicates a high availability to provide services, while a low rating indicates low availability. When assigning a CPU rating, consider the performance and loading of this node relative to other LAT nodes and the level of LAT services you want it to sustain; then, select its CPU rating accordingly.
- user_limit* Is an optional argument that assigns a limit to the combined number of LAT users who can use the IRIS system simultaneously. The limit includes the local terminal server user and LAT users who are remotely logged into the IRIS system.

Setting a Maximum on LAT Login Sessions

The 4DLT installation process sets the maximum number of LAT login sessions to zero (0). The zero value means there is no limit to the number of LAT sessions the IRIS system will accept. Use the *slcp -m* command to reset the limit to a value in the range of 1 to 255:

```
# slcp -m max_sessions <Enter>
```

Setting the LAT Service Announcement Timer

By default, service nodes multicast service announcements every 60 seconds. To change the multicast interval, use the *slcp -t* command:

```
# slcp -t number_of_seconds <Enter>
```

The argument to the *slcp -t* command has this meaning:

- number_of_seconds* Is the number of seconds between multicasts. Use an interval in the range of 10 to 180 seconds in your entry.

Setting a LAT Service Rating

Service ratings provide terminal servers a gauge for selecting a particular service when multiple hosts are offering the service (see “LAT Service Ratings” on page 7). You can assign either a dynamic or static service rating to a login service.

Dynamic ratings are calculated by *latd* on the provider node each time it broadcasts the service. Static ratings in the range of 0 to 255 are assigned by the LAT administrator. Assigning a higher rating makes the service more likely to be selected by a terminal server; assigning a lower rating makes the service less likely to be selected.

Use the *slcp -r* command to set a service rating:

```
# slcp -r service_name ,rating <Enter>
```

The arguments to the *slcp -r* command have these meanings:

<i>service_name</i>	Is the name of the service to which the rating is to be assigned.
<i>rating</i>	If you want <i>latd</i> to assign a rating for this service dynamically when it makes service announcements, set the rating to -1 (negative 1). If you prefer to assign a static service rating, specify a rating in the range of 0 to 255. If you do not assign a service rating, 4DLT uses a default rating of 135.

Manual Binding to 4DLT Devices

The 4DLT installation process creates 256 IRIX devices specifically for LAT services, *ttyl* and *ttyl1* through *ttyl255*. When a local LAT service is created, a specified LAT device is designated for the service (see “Adding Logins to the latservice File” on page 46 and “Adding Printers to the latservice File” on page 47). When this device is opened, the LAT service is automatically bound to the device for the duration of the session.

You can manually bind a LAT device to a LAT service (for diagnostic purposes, for example) using the *slcp -b* command. The *slcp -b* command takes separate arguments for binding to printer services (*-h*) and non-printer services (*-c*).

Note: When you bind to a device, choose one in the middle of the *ttyl1* to *ttyl255* range (see “How Printing Works” on page 26 for information on reserved device numbers).

Binding to Printer Devices

The *svc_setup* utility automates the LAT printer setup by performing three steps automatically. First, it adds a logical printer to the IRIX *lp* spooling system (local print files are routed to the remote printer by means of this logical printer mechanism). Second, *svc_setup* binds the LAT device to the printer service; and third, *svc_setup* adds the printer service to the *latSERVICE* file. If you choose, you can perform these steps manually to add a LAT printer:

1. Add the logical printer using the */usr/lib/lputil* utility.

The *lputil* utility requires that you specify the LAT device that will handle output to the remote printer, the printer model (*lat_dumb* and *lat_gpsinterface* are valid LAT model names), and the name of the printer as it is known to local users. The following sample entry illustrates the *lputil* command to add a logical printer to the *lp* spooling system. In this example, the LAT device *ttyl190* will be used for a PostScript model printer that is known as *laser2*:

```
# lputil add /dev/ttyl190 lat_gpsinterface laser2
```

2. Bind the LAT device to the LAT service using the *slcp -b* command with the *-h* argument.

The *slcp -b* command for binding a device to a printer service has this syntax:

```
slcp -b h,device[,service][[,node,port][,password]
```

When you use this command, you must specify either the name of an advertised service or a node name and port number in your entry. For example, the following command binds device *ttyl190* to an advertised printer service named *laser2* (in this case, *laser2* is also the name by which the printer is known on the local system, but the LAT service

name and local printer name can be different); *laser2* was previously added to the *lp* spooling system. Notice that no node name, port number, or password is specified in this example:

```
slcp -b h, tty1190, laser2
```

The following command binds device *tty1190* to port *ttyd_2* on a LAT node named *lily*. Notice that no service name is specified in this entry:

```
slcp -b h, tty1190, , lily, svr_port_2
```

3. Add the service to the *latservice* file (see “Adding Printers to the *latservice* File” on page 47).

Binding to Non-Printer Devices

The *slcp -b* command for binding a device to a non-printer service has this syntax:

```
slcp -b c, device, service[ , node][ , port][ , password]
```

For example, the following command binds device *tty1162* to a service named *date*. No service node or port number is specified in the binding, and the LAT service requires no password:

```
# slcp -b c, tty1162, date
```

Maintaining the LAT Services File

The LAT services file, */usr/etc/lat/latservice*, contains a record of all services that are to be made available on the IRIS system as a part of the 4DLT startup sequence. The *latservice* file can contain two types of records, login records and printer records. Records can be entered in any order in the file. You can add and remove records from *latservice* using the */usr/etc/lat/svc_setup* utility.

Note: Whenever you change a login record in */usr/etc/lat/latservice*, you must stop and restart *latd* (see “Stopping and Restarting 4DLT” on page 36 for instructions).

A Sample *lat*service File

The samples below illustrate records in a LAT services file:

```
LOGN:Accounting:Login for accounting staff:
PRNT:ttyl124:Colorcopy:*:*:
LOGN:Payroll:13:*:big_$$:
LOGN:Email:-1:Login for electronic mail:
```

The sample records would establish four services during a 4DLT startup:

- | | |
|------------|---|
| Accounting | A login service named "Accounting" that is used by the accounting staff; it has no service rating or password. |
| Colorcopy | A remote printer service named "Colorcopy" that is associated with the ttyl124 device on the IRIS system. |
| Payroll | A login service named payroll, which has a static service rating of 13, but no description (notice the asterisk); it is password protected with big_\$\$. |
| Email | A login service for electronic mail; the service rating (-1) will be dynamically derived by <i>latd</i> when the service is advertised. |

Adding Logins to the *lat*service File

To add a login record to the */usr/etc/lat/lat*service file, use the *svc_setup -l* command. The command has this syntax:

```
# svc_setup -lservice_name [-rrating] [-iidentification] [-apassword]
```

The arguments to *svc_setup -l* have these meanings:

- | | |
|---------------------|--|
| <i>service_name</i> | The name of a login service that is to be established on this node as part of the 4DLT startup process. |
| <i>rating</i> | (optional) A service rating designation (see "LAT Service Ratings" on page 7 and "Setting a LAT Service Rating" on page 43 for more information).

To specify a dynamic rating for this service, set the rating to -1 (negative 1). To assign a static service rating, specify a rating in the range of 0 to 255. When you omit a rating, 4DLT uses a default rating of 135. |

identification (optional) A description of the login service that identifies it in various displays, including user's *CONNECT* and *SHOW SERVICES* commands.

password (optional) A password that LAT users will be required to enter when they connect to the login service.

When the *svc_setup -l* command finishes executing, */usr/etc/lat/latservice* contains a new record for the login service you added (see "A Sample latservice File" on page 46 for sample login records). The service will be advertised after you stop and restart the LAT daemon (see "Stopping and Restarting 4DLT" on page 36 for instructions).

Removing Logins from the *latservice* File

To remove a login record from the */usr/etc/lat/latservice* file, use the *svc_setup -L* command. The command has this syntax:

```
# svc_setup -L service_name
```

When the *svc_setup -L* command finishes executing, the record for the login service that you specified is removed from the *latservice* file. However, you must stop and restart the LAT daemon to stop advertising this login service (see "Stopping and Restarting 4DLT" on page 36 for instructions).

Adding Printers to the *latservice* File

To add a printer service record to the */usr/etc/lat/latservice* file, use the *svc_setup -p* command. The command has this syntax:

```
# svc_setup -pprinter_name -dtty1num -mmodel -sservice [-nnode] [-tport]
```

The arguments to *svc_setup -p* have these meanings:

printer_name The name of a LAT printer as it is known to the IRIS system user.

<i>ttynum</i>	The name of the IRIX device (a <i>ttyl</i> device number, such as <i>ttyl132</i>) to be assigned to this printer. Choose a device number in the middle of the range from 0 to 255, such as <i>tty100</i> (see “How Printing Works” on page 26 for details).
<i>model</i>	A printer model name. The name that you enter must be recognized by the IRIX <i>lp</i> spooling program (see model interface files listed in <i>/usr/spool/lp/model</i> for valid model names).
<i>service</i>	The service name by which this printer is known to LAT users.
<i>node</i>	(optional) The name of the LAT node where the printer is attached.
<i>port</i>	(optional) The number of the port on the LAT node that you specified in the <i>node</i> argument.

When the *svc_setup -p* command finishes executing, */usr/etc/lat/latservice* contains a new record for the printer service you added (see “A Sample latservice File” on page 46 for sample printer records). The LAT printer service is available immediately to the IRIX system user; you do not need to stop and restart the LAT daemon.

Removing Printers from the *latservice* File

To remove a print service record from the */usr/etc/lat/latservice* file, start the *svc_setup* utility with the *-P* option. The command has this syntax:

```
# svc_setup -Pprinter_name -dttynum
```

When the *svc_setup -P* command finishes executing, the record for the print service that you specified is removed from the *latservice* file and the printer is no longer available to the IRIS user; you do not need to stop and restart the LAT daemon to remove the service.

Managing Group Codes

By default, 4DLT software sets all group codes to zero (0), meaning that all nodes and services are available to all LAT users. Using *slcp* commands, you can set up more restrictive group codes if your network requires them. When you implement group codes, access control lists (ACLs) are included in service announcements that the IRIS system multicasts to LAT terminal servers; and identifier lists (IDLs) are included in connection requests that the IRIS system sends to remote service nodes.

Assigning Group Codes with *slcp*

The *slcp -g* command assigns ACL and IDL group codes on an IRIS system. The codes are maintained in the volatile database (they lose their effect when you stop and restart 4DLT). If you assign ACL group codes, incoming service requests will be required to contain a matching group code in order to use the requested service. If you assign IDL group codes, the codes will be forwarded in the IRIS user's request for services and matched with the group codes for the provider node.

The *slcp -g* command has this syntax:

```
slcp -g a|i group_code[,group_code][start_range-end_range]
```

The arguments to the *slcp -g* command have these meanings:

<i>a</i> or <i>i</i>	An <i>a</i> signals that an ACL assignment will be specified; an <i>i</i> signals that an IDL assignment will be specified.
<i>group_code</i>	The number assigned as the code for this group. You can specify an entire list of group codes. Separate individual group codes in the list with a comma.
<i>start_range-</i>	The beginning group code in a group code range. Follow the beginning code with a hyphen (-).
<i>end_range</i>	The last group code in a group code range.

For example, this command assigns the group codes 7 and 15 to service announcements leaving an IRIS host:

```
slcp -g a 7,15
```

This example assigns a range of group codes, from 8 to 12, to all user requests for service leaving the IRIS system:

```
slcp -g i 8,12
```

Automatic Group Code Assignments

Although group codes are not part of the permanent database, you can automate the assignment of ACL and IDL information by specifying them in the IRIS system's */etc/rc2.d/S31lat* file (*slcp* commands are not required). These parameters are read and put into effect each time *latd* is started from a system reboot.

Use this procedure to reset the default group codes from zero (0) to group codes used on your network. The LAT daemon must be running when you do this procedure, and you must do it as the *root* user of the IRIS system:

1. Open the */etc/rc2.d/S31lat* file, as shown in this example:

```
# vi /etc/rc2.d/S31lat <Enter>
```

2. With *latd* running, add the ACL and IDL codes to */etc/rc2.d/S31lat*. Your entries should look like the samples below:

```
slcp -G a,0 <Enter>
slcp -G i,0 <Enter>
slcp -g a,new_ACL_code <Enter>
slcp -g i,new_IDL_code <Enter>
```

The first and second lines of this entry clear the old ACL and IDL codes (which are set to zero (0) by default); the third and fourth lines reset the ACL and IDL codes to a new number that you specify.

You can also specify a range of group codes as ACL or IDL settings, as shown in this sample entry line:

```
slcp -g i, 32-37 <Enter>
```

3. Stop and restart 4DLT for your changes to take effect (see "Stopping and Restarting 4DLT" on page 36).

After you complete this procedure, the IRIS system will send out the ACL codes you specified in its service announcements, and it will include the IDL group codes you specified in its connection requests to other LAT nodes.

Maintaining the LAT Database

Anytime you use the *slcp* utility, your changes are recorded in the volatile database and remain in effect until the LAT daemon is stopped. Some volatile database parameters have corresponding values in the */usr/etc/lat/latconfig* file. Using an *slcp* command, you can write these parameters to the *latconfig* file if you wish to do so. *slcp* also offers a command to purge the volatile database of information about unavailable nodes and services.

Note: See Table 5-1 for a list of */usr/etc/lat/latconfig* parameters that you can change with *slcp* commands.

Overwriting the Permanent Database

Some parameters you can change with *slcp* are also parameters in the permanent database, */usr/etc/lat/latconfig*. To record changes you make with *slcp* in the *latconfig* file, use this command:

```
# slcp -f <Enter>
```

When this command completes, any changes you made to settings in the permanent database using *slcp* commands are recorded in */usr/etc/lat/latconfig*.

Purging Unavailable Nodes and Services

When the terminal server is unable to establish a connection to a LAT node or service, it marks the node unreachable and the service unknown in the LAT database. Unavailable nodes and services and their status continue to be posted in displays, unless you have specified otherwise (see “Setting Learned Service Parameters” on page 63).

By default, 4DLT purges unreachable nodes and unknown services every five hours, removing them from all displays. *slcp* allows you to purge this information on an as-needed basis.

To purge the LAT database, use this command:

```
# slcp -X <Enter>
```

If any new information arrives after the database is purged, the new information is entered in the database in the usual manner.

Deleting LAT Service Queue Entries

4DLT terminal server software maintains a queue of connection requests that operates on a first-in, first-out basis. Incoming requests are assigned a sequence number in the order they are received. The *slcp -Q* command deletes an individual entry or the entire service queue.

Note: To display the service queue, use the *slcp -d q* command.

Use this command to delete an individual request:

```
# slcp -Q sequence_number <Enter>
```

To delete the entire service queue, use this command:

```
# slcp -Q * <Enter>
```

Reconfiguring LAT Operating Parameters

This chapter explains how to change the internal parameters of 4DLT using *lati*, an interactive configuration utility. The *lati* menu and each of its options are described in this chapter:

- “The LAT Configuration Utility Menu”
- “Setting Ethernet Parameters”
- “Setting LAT Node Identification Parameters”
- “Setting LAT Configuration Limits”
- “Setting Circuit Parameters”
- “Setting Session Parameters”
- “Setting Host-initiated Parameters”
- “Setting Advertised Service Parameters”
- “Setting Learned Service Parameters”
- “Setting STREAMS Configuration Parameters”
- “Summary of 4DLT Configuration Parameters”

Some of the parameters you set with *lati* can also be set using the *slcp* utility. Since changes you make with *lati* will only take effect after you stop and restart 4DLT, you might prefer to use *slcp* to change these parameters if a 4DLT shutdown will disrupt system users. See Table 5-1 at the end of this chapter for *slcp* command equivalents to using *lati*.

The LAT Configuration Utility Menu

To start the LAT configuration utility, you enter the *lati* command at the IRIX prompt as the superuser. After your entry, the *lati* menu is displayed:

```
# lati
1. Ethernet device
2. LAT host name, identification
3. LAT configuration parameters, limits
4. Circuit parameters
5. Session parameters
6. Host-Initiated parameters
7. Advertised Service parameters
8. Learned Service parameters
9. Streams configuration
```

Select a parameter category by typing a number from 1 to 9 or type 'q' to quit:

Selecting Menu Options

To select a *lati* menu item, type the number of the item and press **<Enter>**. After your selection, *lati* displays each parameter associated with your selection and the parameter's current value. You can take three actions on any parameter:

- To change a parameter value, enter a new value and press **<Enter>**.
- To retain the current parameter value, press **<Enter>**.
- To set a null string, press **<Esc><Enter>**.

Putting Your Changes into Effect

The changes you make with the *lati* utility are written to the */usr/etc/lat/latconfig* file. The changes take effect when the new copy of *latconfig* is copied into main memory. To write a new copy of *latconfig* to memory, you must stop and restart *latd* (see "Stopping and Restarting 4DLT" on page 36).

Setting Ethernet Parameters

To set LAT Ethernet interface parameters, use item 1 on the *lati* menu:

1. **Ethernet device**
2. LAT host name, identification
3. LAT configuration parameters, limits
4. Circuit parameters
5. Session parameters
6. Host-Initiated parameters
7. Advertised Service parameters
8. Learned Service parameters
9. Streams configuration

Select a parameter category by typing a number from 1 to 9 or type 'q' to quit: 1

Item 1 on the *lati* menu provides these parameters:

NET_DEVICE Specifies the pathname to the LAT network interface.

4DLT installation sets *NET_DEVICE* to */dev/snif*, which uses either the primary network interface (see "The Default Network Interface for LAT" on page 29) or a specified alternate interface (see "Starting 4DLT with an Alternate LAN Interface" on page 34).

Do not change the value of *NET_DEVICE*; it should remain set to */dev/snif*.

HS_MULTI_ON

Switches the IRIS system between dual purpose and general purpose operation.

Set to 1, the default setting, to operate the IRIS system as a service node and terminal server concurrently.

Set to 0 (zero) to operate the IRIS system as a service node only. This setting disables the IRIS system's ability to receive and process incoming service announcements, thereby disabling terminal server functions.

Setting LAT Node Identification Parameters

To set LAT node identification parameters, use item 2 on the *lati* menu:

1. Ethernet device
- 2. LAT host name, identification**
3. LAT configuration parameters, limits
4. Circuit parameters
5. Session parameters
6. Host-Initiated parameters
7. Advertised Service parameters
8. Learned Service parameters
9. Streams configuration

Select a parameter category by typing a number from 1 to 9 or type 'q' to quit: **2**

Item 2 on the *lati* menu provides these parameters:

HS_NODE Sets the LAT node name for the IRIS host.

4DLT installation sets this to the name contained in the IRIS system's */etc/sys_id* file. Keep these guidelines in mind if you change a LAT node name:

1. The node name must be unique on the LAN.
2. Node names can contain a maximum of 16 characters.
3. Valid characters include all alphanumeric characters and the symbols dollar sign (\$), period (.), hyphen (-), and underscore (_).
4. Uppercase and lowercase letters are valid, but lowercase letters are converted for internal operations.

Note: You can set this parameter with the *slcp -n* command.

HS_IDENT Is an optional character string that describes the IRIS node to LAT users.

Note: You can set this parameter with the *slcp -n* command.

HS_FACILITY Is an optional local facility number used at some LAT network sites.

HS_CPU_RATING

Specifies the CPU rating, a factor that the service node uses in combination with HS_JOB_LIMIT to calculate service ratings.

Valid range is 1 through 100. Set higher to increase service ratings and host usage; set lower to reduce service ratings and host usage. Default setting is -1.

This parameter is not used if dynamic service ratings are enabled (*slcp -r* is set to -1).

Note: You can set this parameter with the *slcp -c* command.

HS_JOB_LIMIT

Specifies the maximum number of LAT users on the IRIX system, a factor that the service node uses in conjunction with HS_CPU_RATING to calculate service ratings.

Set to the expected user total—the local terminal server user and remote login service users combined.

Default setting is 50.

Note: You can set this parameter with the *slcp -c* command.

Setting LAT Configuration Limits

To set LAT configuration limits, use item 3 on the *lati* menu:

1. Ethernet device
2. LAT host name, identification
- 3. LAT configuration parameters, limits**
4. Circuit parameters
5. Session parameters
6. Host-Initiated parameters
7. Advertised Service parameters
8. Learned Service parameters
9. Streams configuration

Select a parameter category by typing a number from 1 to 9 or type 'q' to quit: **3**

Item 3 on the *lati* menu provides these parameters:

HS_SB_MAXIMUM

Specifies the maximum LAT sessions that can be created on the service node.

Set to 0 (zero), the default setting, for no limit.

Valid range is 1 to 255 sessions.

Note: You can set this parameter with the *slcp -m* command.

HS_CB_MAXIMUM

Specifies the maximum circuits that can be created to a service node.

Set to 0 (zero), the default setting, for no limit.

HS_ND_MAXIMUM

Specifies the maximum remote nodes learned from the network.

Both terminal server and service node software use this value in establishing virtual circuits.

The default value is 100.

HS_LS_MAXIMUM

Specifies the maximum number of LAT services the terminal server collects from multicast service announcements.

Default setting is 100.

Set to 0 (zero) for no limit. Set to -1 to prevent a terminal server from accepting any learned service information.

HS_QR_MAXIMUM

Specifies the maximum size of the terminal server's queue for host-initiated requests.

Default setting is 32.

Set to 0 (zero) for no limit. Set to -1 to prevent any requests from being queued.

Setting Circuit Parameters

To set circuit parameters, use item 4 on the *lati* menu:

1. Ethernet device
2. LAT host name, identification
3. LAT configuration parameters, limits
- 4. Circuit parameters**
5. Session parameters
6. Host-Initiated parameters
7. Advertised Service parameters
8. Learned Service parameters
9. Streams configuration

Select a parameter category by typing a number from 1 to 9 or type 'q' to quit: **4**

Item 4 on the *lati* menu provides these parameters:

HS_DLL_MAX

Specifies the maximum size for an outgoing Ethernet packet, not including header.

Default value is 1500 bytes.

Valid range is 576 to 1500 bytes.

To make efficient use of streams buffers, set at 1024.

HS_INP_MAX Specifies the maximum size for an incoming Ethernet packet, not including header.

Default value is 1500 bytes; this is the recommended setting for Silicon Graphics systems.

Valid range is 576 to 1500 bytes.

HS_CB_MAX_SES

Specifies the maximum number of sessions permitted on a single circuit.

Valid range is 1 to 255.

Default value is 64 sessions.

HS_CB_RETRIES

Specifies the maximum number of attempts the terminal server makes to establish a virtual circuit before it declares a circuit time-out error.

Valid range is 4 to 120 attempts.

Default setting is 8 attempts.

HS_CB_VC_TIMER

Specifies the setting in milliseconds for the virtual circuit timer on the terminal server; virtual circuit messages are sent when the timer reaches 0 (zero).

Valid range is 10 to 1000 milliseconds.

Recommended setting is 80 milliseconds.

Default setting is 80 milliseconds.

Note: You can set this parameter with the *slcp -t* command.

HS_CB_HD_TIMER

Specifies in milliseconds the time that a service node will wait before responding to a virtual circuit message from a terminal server.

Recommended setting is half the value of the setting on the circuit timer, HS_CB_VC_TIMER, or 40 milliseconds by default.

HS_CB_KA_TIMER

Specifies in milliseconds the intervals at which the terminal server sends keep-alive messages over an idle circuit.

Valid range is 10 to 255 milliseconds.

Recommended setting is 20 milliseconds.

Default setting is 20 milliseconds.

HS_CB_PROGRESS

Specifies in seconds the amount of time the service node will wait for a virtual circuit message before disconnecting the circuit with the terminal server.

Valid range is 0 (zero) to 255 seconds.

Default setting is 60 seconds.

If set to 0 (zero), the virtual circuit will never be disconnected as a result of a message processing time-out.

Setting Session Parameters

To set session parameters, use item 5 on the *lati* menu:

1. Ethernet device
2. LAT host name, identification
3. LAT configuration parameters, limits
4. Circuit parameters
- 5. Session parameters**
6. Host-Initiated parameters
7. Advertised Service parameters
8. Learned Service parameters
9. Streams configuration

Select a parameter category by typing a number from 1 to 9 or type 'q' to quit: 5

Item 5 on the *lati* menu provides these parameters:

HS_CREDITS Specifies the number of credits (that is, receive buffers) the terminal server allocates per session.

Recommended value is 2 (also the default value).

HS_SB_MAX_DATA

Specifies the maximum size in bytes of an individual virtual circuit message.

Valid range is 1 to 255 bytes.

Recommended setting is 255 bytes.

Reduce to 128 bytes to make efficient use of the 128-byte streams buffer pool.

Setting Host-initiated Parameters

To set parameters on host-initiated requests, use item 6 on the *lati* menu:

1. Ethernet device
2. LAT host name, identification
3. LAT configuration parameters, limits
4. Circuit parameters
5. Session parameters
- 6. Host-Initiated parameters**
7. Advertised Service parameters
8. Learned Service parameters
9. Streams configuration

Select a parameter category by typing a number from 1 to 9 or type 'q' to quit: **6**

Item 6 on the *lati* menu provides these parameters:

HS_HI_STS_TIMER

Specifies in milliseconds the time between queue status messages sent by a terminal server during a host-initiated request.

Valid range is 10 to 3600 milliseconds.

Default setting is 60 milliseconds.

HS_HI_CMD_TIMER

Specifies in seconds the time that a service node waits before resending a host-initiated requests to a terminal server.

Recommended value is 1 to 60 seconds.

Default setting is 15 seconds.

HS_HI_CMD_RETRIES

Specifies the number of times the service node will resend a host-initiated request to a terminal server before giving up.

Valid range is 2 to 120 attempts.

Default setting is 10 attempts.

Setting Advertised Service Parameters

To set parameters on advertised services, use item 7 on the *lati* menu:

1. Ethernet device
2. LAT host name, identification
3. LAT configuration parameters, limits
4. Circuit parameters
5. Session parameters
6. Host-Initiated parameters
- 7. Advertised Service parameters**
8. Learned Service parameters
9. Streams configuration

Select a parameter category by typing a number from 1 to 9 or type 'q' to quit: 7

Item 7 on the *lati* menu provides this parameter:

HS_AS_TIMER

Specifies in seconds the interval at which a service node sends out service messages.

Valid range is 10 to 180 seconds.

Default setting is 60 seconds.

Note: You can set this parameter with the *slcp -t* command.

Setting Learned Service Parameters

To set parameters for learned services, use item 8 on the *lati* menu:

1. Ethernet device
2. LAT host name, identification
3. LAT configuration parameters, limits
4. Circuit parameters
5. Session parameters
6. Host-Initiated parameters
7. Advertised Service parameters
- 8. Learned Service parameters**
9. Streams configuration

Select a parameter category by typing
a number from 1 to 9 or type 'q' to quit: **8**

Item 8 on the *lati* menu provides these parameters:

HS_LE_LS_DUPS

Specifies whether the terminal server should maintain duplicate service names.

Set to 0 (zero) to reject duplicate service names. This setting prevents the selection of services based on service ratings.

Set to 1 (the default) to maintain duplicate service names. This setting allows the terminal server to select services based on its service rating.

HS_LE_ND_RETAIN

Specifies whether the terminal server should display unknown and unreachable services in service listings.

Set to 0 (zero) to omit unknown and unreachable services when displaying the list of services.

Set to 1 (the default) to include unknown and unreachable services when displaying the list of services.

HS_ND_AGE_MAX

Specifies the number of seconds that a terminal server will wait before purging unknown and unreachable services from its database.

Valid range is 0 (zero) to 28800 seconds (8 hours).

The default setting is 18000 seconds (5 hours).

Set to -1 to disable purging of unknown and unreachable services.

Setting STREAMS Configuration Parameters

To set streams configuration parameters, use item 9 on the *lati* menu:

1. Ethernet device
2. LAT host name, identification
3. LAT configuration parameters, limits
4. Circuit parameters
5. Session parameters
6. Host-Initiated parameters
7. Advertised Service parameters
8. Learned Service parameters
- 9. Streams configuration**

Select a parameter category by typing a number from 1 to 9 or type 'q' to quit: **9**

Item 9 on the *lati* menu provides these parameters:

NET_STREAM_1

Specifies the first (lower) module on the Ethernet stream.

Default setting is null (' ').

NET_STREAM_2

Specifies the second (higher) module on the Ethernet stream.

Default setting is null (' ').

LAT_FDEV

Specifies the LAT device name for login terminals.

Default setting is */dev/ttyl*.

LAT_CDEV

Specifies the LAT device name for outgoing sessions.

Default setting is */dev/lats*.

LAT_STREAM_1

Specifies the first (lowest) module to push onto a session stream.

Default value is null (' ').

LAT_STREAM_2

Specifies the second module to push onto a session stream.

The default value is null ('').

LAT_STREAM_3

Specifies the third module to push onto a session stream.

Default value is *tirdwr* (handles STREAMS read/write calls).

The read/write module must be pushed onto the stack before the line discipline module.

LAT_STREAM_4

Specifies the fourth (higher) module to push onto a session stream. .

Default value is *stty_ld* (handles line discipline). This module must be the first to appear to the user.

Summary of 4DLT Configuration Parameters

Table 5-1 contains an alphabetical listing of the 4DLT configuration parameters that you can change with *lati*.

Table 5-1 Summary of 4DLT Configuration Parameters

Parameter Name	Parameter Function	Default Value	Item on <i>lati</i> Menu	<i>slcp</i> Equivalent
HS_AS_TIMER	Specifies the interval (in seconds) for multicasting service announcements	60	Item 7	slcp -t
HS_CB_HD_TIMER	Specifies time (in milliseconds) a service node waits before responding to terminal servers	40	Item 4	None
HS_CB_KA_TIMER	Specifies interval (in milliseconds) at which a terminal server sends keep-alive messages	20	Item 4	None
HS_CB_MAXIMUM	Specifies maximum number of circuits to a service node	0 (no limit)	Item 3	None
HS_CB_MAX_SES	Specifies the maximum number of sessions on a single circuit	64	Item 4	None
HS_CB_PROGRESS	Specifies time (in seconds) the service node waits for a virtual circuit message before disconnecting a circuit	60	Item 4	None
HS_CB_RETRIES	Specifies the maximum number of attempts a terminal server makes to establish a virtual circuit before declaring a time-out error	8	Item 4	None
HS_CB_VC_TIMER	Specifies the interval (in milliseconds) between virtual circuit messages from the terminal server	80	Item 4	slcp -t
HS_CPU_RATING	Sets the CPU rating for a service node	-1 (dynamic ratings)	Item 2	slcp -c
HS_CREDITS	Specifies the number of credits the terminal server allocates per session	10	Item 5	None
HS_DLL_MAX	Specifies (in bytes) the size of an outgoing Ethernet packet, excluding headers	1500	Item 4	None
HS_FACILITY	Specifies the local facility number	0	Item 2	None

Table 5-1 (continued) Summary of 4DLT Configuration Parameters

Parameter Name	Parameter Function	Default Value	Item on <i>lati</i> Menu	<i>slcp</i> Equivalent
HS_HI_CMD_RETRIES	Specifies the number of times that a service node resends a host-initiated request	10	Item 6	None
HS_HI_CMD_TIMER	Specifies the time (in seconds) that a service node waits between retries for a host-initiated request	15	Item 6	None
HS_HI_STS_TIMER	Specifies the interval (in milliseconds) between queue status messages sent by a terminal server during a host-initiated request	60	Item 6	None
HS_IDENT	Assigns a description to an IRIS node so LAT users can identify it	none	Item 2	<i>slcp -n</i>
HS_INP_MAX	Specifies the maximum size (in bytes) of Ethernet packets, excluding headers	1500	Item 4	None
HS_JOB_LIMIT	Specifies the maximum number of LAT users on an IRIS system (terminal server and service node combined)	50	Item 2	<i>slcp -c</i>
HS_LE_LS_DUPS	Specifies whether a terminal server should maintain duplicate service names	1	Item 8	None
HS_LE_ND_RETAIN	Specifies whether a terminal server should display unknown and unavailable services	1	Item 8	None
HS_LS_MAXIMUM	Specifies the maximum number of services maintained in a terminal server's database	100	Item 3	None
HS_MULTI_ON	Switches an IRIS system between a dual purpose and general purpose LAT node	1 (dual purpose)	Item 1	None
HS_ND_AGE_MAX	Specifies interval (in seconds) at which a terminal server purges unavailable nodes and services from its database	18000 (5 hours)	Item 8	None
HS_ND_MAXIMUM	Specifies the maximum number of nodes maintained in a terminal server's database	100	Item 3	None
HS_NODE	Sets the LAT node name on an IRIS host	name in <i>/etc/sys_id</i>	Item 2	<i>slcp -n</i>
HS_QR_MAXIMUM	Specifies the maximum size of a terminal server's queue for host-initiated requests	32	Item 3	None

Table 5-1 (continued) Summary of 4DLT Configuration Parameters

Parameter Name	Parameter Function	Default Value	Item on <i>lati</i> Menu	<i>slcp</i> Equivalent
HS_SB_MAX_DATA	Specifies the maximum size of an individual data slot message	255	Item 5	None
HS_SB_MAXIMUM	Specifies the maximum number of sessions that a terminal server or server node will allow	0 (no limit)	Item 3	<i>slcp -m</i>
LAT_FDEV	Specifies the device name for LAT login terminals	<i>/dev/ttyl</i>	Item 9	None
LAT_CDEV	Specifies the device name for outgoing LAT sessions	<i>/dev/lats</i>	Item 9	None
LAT_STREAM_1	Specifies the first (lowest) module to push onto a session stream	null	Item 9	None
LAT_STREAM_2	Specifies the second module to push onto a session stream	null	Item 9	None
LAT_STREAM_3	Specifies the third module to push onto a session stream	<i>tirdwr</i>	Item 9	None
LAT_STREAM_4	Specifies the fourth (highest) module to push onto a session stream	<i>stty_ld</i>	Item 9	None
NET_DEVICE	Specifies the IRIS node's LAT device	<i>/dev/snif</i>	Item 1	None
NET_STREAM_1	Specifies the first (lower) module on an Ethernet stream	null	Item 9	None
NET_STREAM_2	Specifies the higher module on an Ethernet stream	null	Item 9	None

Glossary

access control list (ACL)

A list of group codes that a LAT service node includes in a service announcement to designate the LAT users permitted to use the service.

balanced mode

The state entered by a LAT virtual circuit when it has no data to send. In balanced mode, empty data messages are exchanged between the LAT terminal server and service node to maintain an open virtual circuit.

bandwidth

The amount of data that can be transmitted in a specific period of time over a physical medium.

circuit timer

A software timer in a LAT terminal server that determines the interval at which data from user sessions is sent from the terminal server to a service node.

credit

A marker that a LAT terminal server includes in the slot data of a LAT session to permit the receiving node to transmit a slot of data in reply. The maximum size of slot data accompanying a credit is 255 bytes.

Data Link layer

In the ISO reference model, the network layer above the Physical Link layer that ensures the safe transport of datagrams between communication endpoints.

datagrams

Discrete data units, or packets, transferred over networks that have a distinct format dictated by the transport medium.

dual purpose node

A LAT node that performs both service node and terminal server functions.

dynamic service rating

A measure of the availability of a LAT service that is calculated by a service node at the time the service is requested.

general purpose node

A LAT node that performs service node functions.

group code

A number assigned to a group of LAT users (by means of their terminal ports) and LAT resources to designate access permissions; access is granted when a user's group code matches the group code of the node or service he or she is requesting.

host-initiated request

A request for service that a LAT service node makes to a terminal server.

identifier list (IDL)

A list of group codes that a LAT terminal server includes in a connection request to designate the services that the requester is permitted to access.

keep-alive message

In a LAT network, an empty virtual circuit message that a terminal server and service node exchange to keep an idle connection open.

LAN interface

An IRIX module that acts as the intermediary between a LAN hardware device (such as an Ethernet board) and the IRIX operating system.

LAT database

On a terminal server, the body of information that defines the LAT configuration and describes the LAT nodes and services that users can request.

LAT (Local Area Transport)

A protocol for local area networks that specializes in communications between the hosts and user terminals connected to the network.

LAT node

In a LAT network, any node running LAT protocol software.

LAT software

The software that implements the LAT protocol. 4DLT is LAT software that runs on an IRIS host.

latd

The daemon process on an IRIS node that implements the LAT protocol creates the remote terminal processes upon request by the terminal server.

lati

An interactive utility used to set the operating parameters for 4DLT software.

load balancing

Directing new connections to the node with the most available resources. In a LAT network, load balancing is performed by the terminal server, which routes connection requests to the service with the highest rating.

multicast timer

The software timer in a LAT service node that regulates the interval at which service announcements are dispatched.

multicasting

The process by which a message is sent to a set of logically related nodes by means of a single, designated address. In a LAT network, the group of related nodes are terminal servers, which receive service announcements by means of a multicast address.

multiplexing

The ability to combine accumulated user data from simultaneous sessions into a single packet or datagram. Multiplexing is performed by the terminal server and host LAT software.

OSI (Open Systems Interconnection)

A concept in communications technology, espoused by the International Organization for Standards (ISO), in which networks are designed to permit data exchange among all types of communications equipment. Also, the ISO standard for achieving this interconnectivity, by which networks are designed according to a hierarchy of layers, each of which performs a specified function.

permanent database (LAT)

Information about the LAT configuration, stored in a file on a LAT terminal server or service node, that is used as the initial database when the LAT protocol is started. (See volatile database.)

Physical Link layer

In a DECnet network, the lowest layer of network architecture. The Physical Link layer provides the hardware medium and defines the electrical and mechanical interfaces to the medium.

protocol stack

In networking, the combination of specialized functional layers that constitute a particular protocol, such as OSI or LAT.

service

A named resource that LAT hosts provide to network users.

service announcement

A multicast message from a LAT service node that announces the services offered by the node.

Service Class layer

The uppermost layer in the LAT network protocol. The Service Class layer multicasts service announcements to LAT nodes, maintains directories of LAT nodes and services, and makes services available to LAT users.

service directory

A collection of LAT service information that is maintained in main memory of a terminal server. For each service, the service directory contains the service name and description, the name of the provider node, the rating and availability of the service, and any group codes associated with it.

service name

A name, from 1 to 16 characters long, that a LAT administrator assigns to a LAT resource.

service node

A host or server on a LAT network that offers one or more services to LAT users on terminal servers. An IRIS system running 4DLT software functions as a service node.

service node directory

A collection of LAT service node information that is maintained in main memory of a terminal server. For each service node, the service node directory contains the name and description of the service node, the network address of the service node, and the node's availability for accepting connections.

service rating

A measure of the availability of a LAT service that can be dynamically calculated by a service node or assigned a static value by the LAT administrator. LAT service nodes advertise service ratings in service announcements.

session

An asynchronous, two-way logical connection between a LAT service node and terminal server that enables a user or an application to exchange data with a host providing a LAT service.

slcp

The 4DLT utility used to maintain LAT functions on an IRIS host.

slot

A segment within a virtual circuit message that contains user data and control information for an individual LAT session.

Slot layer

In the LAT network protocol, the layer between the Service Class and Virtual Circuit layers. The Slot layer opens and closes sessions, multiplexes slot data into a virtual circuit message, and manages the flow of data between the partner nodes in a LAT session.

static service rating

A measure of the availability of a service that is assigned by a LAT administrator and remains constant in all service announcements.

terminal server

A special-purpose communications device that provides LAT connections between its ports and other LAT nodes. Also, a host running software that performs terminal server functions, such as an IRIS system running 4DLT software.

user-initiated request

A request for a LAT service that a user issues by means of a *CONNECT* command.

virtual circuit

A logical communication path between a terminal server and a host. The LAT transport mechanism is a virtual circuit.

Virtual Circuit layer

In the LAT network protocol, the layer between the Slot layer and the Data Link layer. The Virtual Circuit layer establishes and maintains a logical connection (virtual circuit) between LAT partner nodes.

volatile database (LAT)

The version of the LAT database that is operating in main memory on a LAT terminal server. The volatile database contains the permanent database, information collected from the LAT network, and LAT administrator entries. (See permanent database.)

Index

Numbers

4DLT

- components, 18
- default configuration, 28
- directories, 28
- installation, 27, 28

A

- adding login services, 46
- adding printer services, 45, 47
- advertised services parameters, 63

B

- balanced mode, 14
- binding to devices, 43, 44

C

- circuit parameters, 59-61
- circuit timer, 14
- CPU rating, 41, 57
- credits, 12, 61

D

- Data Link layer, 15
- datagrams, 15
- device files, 26, 43
- /dev/lats* device, 65
 - purpose of, 18
- /dev/snif* device, 55
 - in the LAT protocol stack, 20
- /dev/ttyl* device, 65
- DNA reference model, 9
- dual purpose nodes
 - defined, 1
 - enabling, 32
 - setting for, 55
- dynamic service ratings. *See* service ratings .

E

- /etc/chkconfig* file, 34
- Ethernet interface parameters, 55
- Ethernet interface. *See also* network interface., 29
- Ethernet packets, 59

F

- flow control, 12

G

- general purpose nodes
 - defined, 1
 - setting for, 55
- group codes
 - assigning, 49
 - assigning automatically, 50
 - defined, 8
 - effects of, 49
 - purpose of, 8

H

- host-initiated request parameters, 62
- host-initiated requests
 - defined, 6
- HS_AS_TIMER* parameter, 63
- HS_CB_HD_TIMER* parameter, 60
- HS_CB_KA_TIMER* parameter, 60
- HS_CB_MAX_SES* parameter, 59
- HS_CB_MAXIMUM* parameter, 58
- HS_CB_PROGRESS* parameter, 60
- HS_CB_RETRIES*, 60
- HS_CB_VC_TIMER* parameter, 60
- HS_CPU_RATING* parameter, 57
- HS_CREDITS* parameter, 61
- HS_DLL_MAX* parameter, 59
- HS_FACILITY* parameter, 56
- HS_HI_CMD_RETRIES* parameter, 62
- HS_HI_CMD_TIMER* parameter, 62
- HS_HI_STS_TIMER* parameter, 62
- HS_IDENT* parameter, 56
- HS_INP_MAX* parameter, 59
- HS_JOB_LIMIT* parameter, 57
- HS_LE_LS_DUPS* parameter, 64

- HS_LE_ND_RETAIN* parameter, 64
- HS_LS_MAXIMUM* parameter, 58
- HS_MULTI_ON* parameter, 55
- HS_ND_AGE_MAX* parameter, 64
- HS_ND_MAXIMUM* parameter, 58
- HS_NODE* parameter, 56
- HS_QR_MAXIMUM* parameter, 58
- HS_SB_MAX_DATA* parameter, 61
- HS_SB_MAXIMUM* parameter, 58

I

- installing 4DLT, 27
- IRIX search path setup, 28, 32

K

- keep-alive messages, 14

L

- LAT configuration parameters, 57-58
- LAT database
 - defined, 5
 - initializing, 20
 - life-cycle events in, 25
 - maintaining, 51-52
 - organization of, 22-25
 - purging, 51, 64
- LAT devices, 26, 44
- LAT network
 - architecture of, 8
 - components of, 3
 - defined, 2
 - monitoring, 39

LAT protocol, 1
LAT protocol stack, 9, 19
LAT_CDEV parameter, 65
LAT_FDEV parameter, 65
LAT_STREAM_1 parameter, 65
LAT_STREAM_2 parameter, 66
LAT_STREAM_3 parameter, 66
latconfig file
 and the *latd* daemon, 19, 20
 changing, 21, 30
 changing with *lati*, 54
 default contents of, 23, 30
 overriding values in, 23
 overwriting with *slcp*, 51
 protocol stack in, 19
latd daemon
 functions of, 18-20
 starting, 19, 33, 35
 stopping, 36
lati utility
 main menu for, 54
 selecting options from, 54
 setting advertised services parameters with, 63
 setting circuit parameters with, 59
 setting Ethernet parameters with, 55
 setting host-initiated request parameters with, 62
 setting LAT limits with, 57
 setting learned services parameters with, 63
 setting node parameters with, 56
 setting session parameters with, 61
 setting streams parameters with, 65
 starting, 54
latservice file
 adding services to, 31
 and the *latd* daemon, 20
 changing, 30, 45
 contents of, 45
 default contents of, 30
 example of, 46

 maintaining, 45-48
 permissions on, 31
 purpose of, 26
 record syntax in, 30
learned services, 24, 58
learned services parameters, 63
load balancing, 12
Local Area Transport. *See* LAT.
login services
 adding, 46
 removing, 47

M

multicast timer, 10
multicasting
 service announcements, 20
 setting intervals for, 42, 63
multicasting (service announcements), 6
multiplexing
 by terminal servers, 5
 Slot layer, 11
 Virtual Circuit layer, 13

N

NET_DEVICE parameter, 55
NET_STREAM_1 parameter, 65
NET_STREAM_2 parameter, 65
network interface
 default, 29
 displaying, 34
 specifying, 55
 using non-default, 34-35
node description
 changing, 40
node identification parameters, 56-57

node name
 changing, 40
 default value of, 29, 40
 setting with *lati*, 56
nodes
 defined, 3
 naming, 29
 unknown, 24
 unreachable, 21, 51

O

OSI reference model, 9

P

packet size parameters, 59
path setup, 28, 32
permanent database
 Also see latconfig file and volatile database.
 changing, 21
 changing with *lati*, 54
 default contents of, 23
 defined, 22
 overriding values in, 23
 overwriting with *slcp*, 51
Physical Link layer, 15
printer services
 adding, 47
 adding manually, 44
 implementation of, 26
 removing, 48
purging services, 51

R

removing login services, 47
removing printer services, 48
removing service requests, 52
root user account
 for *slcp* commands, 22
 modifying, 28
run messages, 13

S

S31lat initialization file, 35, 50
service announcement timer, 42
service announcements
 contents of, 6
 defined, 6
Service Class layer, 10-11
Service Class messages, 11
service node directory, 24
service nodes
 defined, 5
service queue, 52, 58
service ratings
 defined, 6
 dynamic, 7, 43, 57
 purpose of, 7
 setting, 43, 64
 static, 7, 43
services
 adding, 31
 allocation of, 7
 and the *latd* daemon, 20
 defined, 2
 deleting requests for, 52

services directory, 24
session parameters, 61
sessions
 defined, 4
 setting maximum number of, 42, 58
slat utility
 functions of, 21
slcp utility
 and the *latconfig* file, 22
 and the volatile database, 22, 37
 changing node description with, 40
 changing node names with, 40
 changing the permanent database with, 51
 command syntax for, 39
 displaying information with, 39
 displaying options for, 38
 functions of, 21
 purging the database with, 51
 setting CPU ratings with, 41
 setting group codes with, 49
 setting maximum sessions with, 42
 setting multicast timer with, 42
 setting service ratings with, 43
Slot layer, 11-13
slots, 11
start messages, 13
startup command, 33
startup file, 34
startup procedures, 33-36
static service rating. *See also* service rating, 7
stop messages, 13
streams configuration parameters, 65-66
svc_setup utility
 adding services with, 46, 47
 purpose of, 26
 removing services with, 47, 48

T

terminal servers
 defined, 4
 printer connections to, 5
 terminal connections to, 5
terminals (LAT), 4
time-out errors, 60
ttyl devices, 26

U

unbalanced mode, 14
unit numbers, 34
unknown services, 51, 64
user limits, 41
user workstation preparation, 32
user-initiated requests. *See also* host-initiated requests., 4
/usr/etc/lat/latconfig file. *See* *latconfig* file.
/usr/etc/lat/latservice file. *See* *latservice* file.
/usr/etc/lat/svc_setup utility. *See* *svc_setup* utility

V

virtual circuit, 13
Virtual Circuit layer, 13-14
virtual circuit messages, 13
virtual circuits, 58
volatile database
 and *slcp*, 22
 contents of, 23-25

We'd Like to Hear From You

As a user of Silicon Graphics documentation, your comments are important to us. They help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics to comment on:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please include the title and part number of the document you are commenting on. The part number for this document is 007-1396-020.

Thank you!

Three Ways to Reach Us



The **postcard** opposite this page has space for your comments. Write your comments on the postage-paid card for your country, then detach and mail it. If your country is not listed, either use the international card and apply the necessary postage or use electronic mail or FAX for your reply.



If **electronic mail** is available to you, write your comments in an e-mail message and mail it to either of these addresses:

- If you are on the Internet, use this address: techpubs@sgi.com
- For UUCP mail, use this address through any backbone site:
[your_site]!sgi!techpubs



You can forward your comments (or annotated copies of manual pages) to Technical Publications at this **FAX** number:

415 965-0964