Printing History

First Edition - January 1991

Contents

Figures

Tables

1

Introduction

SRM/UX is the Shared Resource Management product on the HP-UX operating system. SRM/UX is a computer networking product that allows HP BASIC/WS and Pascal Workstation System SRM/UX clients to boot from the SRM/UX server's mass storage, share files under the server's file system, and spool files to shared peripherals that are connected to the server. The SRM/UX server is a daemon process on an HP-UX host computer.

What You Should Know Before You Begin

There are many terms used in this manual that may be completely new to you. Every attempt has been made to define terms when they are first used. However, we recommend that you scan the Glossary at the back of this guide first to familiarize yourself with the terminology. Then in chapter 2 you will go through an SRM/UX system orientation that will clarify these terms further and put them into context for you.

This guide is written for two different audiences: SRM/UX System Administrators and client users.

The bulk of the material is aimed at SRM/UX System Administrators. The discussion throughout this guide assumes that the SRM/UX System Administrator has been trained in HP-UX system administration, either formally or through on-the-job training. Many of the procedures in SRM/UX system administration cannot be done without a basic understanding of the HP-UX operating system.

Note

SRM/UX System Administrators may find the following HP-UX manuals helpful for some of the SRM/UX administration tasks. Refer to:

- Section 1M and the Glossary of the *HP-UX Reference*
  - □ For descriptions of HP-UX system administration commands.
  - □ For definitions of HP-UX terms.
- *HP-UX System Administration Concepts* and *HP-UX System Administration Tasks*

  For HP-UX system administration tasks, such as:
  - □ Installing new software (SRM/UX) on your HP-UX system.
  - □ Configuring the HP-UX kernel.
  - □ Setting up HP-UX spooling.
  - □ Adding devices.
  - □ Administering an HP-UX cluster server.
- *HP-UX System Security*
  - □ For HP-UX security concepts. This may prove helpful in planning for your SRM/UX security needs.
- *ARPA Services/300 User's Guide*
- *Installing and Administering NFS Services* and/or *Installing and Administering LAN/9000 Series 300* and/or *Installing and Administering LAN/9000 Series 800*
  - □ For descriptions of Internet Addresses. The SRM/UX Administrator will need to know about Internet Addresses if the SRM/UX server will use LAN interfaces to communicate with SRM/UX clients.
  - □ For descriptions of NFS. The Administrator will need to know about NFS if the SRM/UX server will allow SRM/UX clients to access remote file systems.

Client users do not need to know HP-UX, but there are times that such knowledge would be helpful. The client users need only be familiar with the particular language system they will be using: either HP BASIC/WS or Pascal Workstation System.

A Map Through This Guide

The chapter descriptions below are preceded by either an S (server), a C (client), or both (SC) to indicate the audience to which the chapter is addressed. That is, entries headed with an S indicate that there is significant content on the server side, and entries headed with a C indicate that there is significant content on the client side.

Following the chapter list is the recommended path through the chapters to most efficiently bring up your SRM/UX system.

- (SC) Chapter 1: "Introduction"

  Read this chapter to understand the requirements for the SRM/UX system (both hardware and software), as well as the requirements for the system users (both server system administrator and client workstation users). This chapter also recommends a flow or path through the chapters to use this guide most efficiently.

- (SC) Chapter 2: "SRM/UX Orientation and System Planning Guide"

  Read this chapter to orient yourself with SRM/UX. This chapter will give you a broad overview of what SRM/UX is all about, and will familiarize you with all of the component parts of the system. It discusses how SRM/UX emulates the SRM product: what is the same and what is different. Then it guides you through the process of planning your SRM/UX system and provides tools to help with the planning.

- (SC) Chapter 3: "Installing the Hardware"

  Read this chapter to learn how to install and configure the SRM/UX system hardware. (This chapter will NOT tell you how to actually install the server or client workstations. See the appropriate manuals that come with your workstations for those instructions.) It will tell you how to configure and install SRM and LAN interfaces and how to install the cabling, adapters, terminators, etc. between the workstations. Since the administration of the server's discs, printers, and plotters has moved into the domain of HP-UX system administration, this manual does not tell you how to do that administration—you should refer to the appropriate HP-UX system administration manuals, instead.

- (S) Chapter 4: "Installing, Configuring, and Running the SRM/UX Server"

  Read this chapter for the in-depth details of administering the SRM/UX server. This chapter will tell you the details about the SRM/UX file system and spooling environment from the server's perspective. It tells you how to configure SRM/UX through the server's /etc/srmdconf configuration file. It also tells you how to start the server process, *srmd*(1M), initially and how to stop and restart it. Finally, it tells you the details of dynamic reconfiguration, that is, how to change the server's configuration without shutting *srmd*(1M) down.

- (SC) Chapter 5: "Booting Clients From the SRM/UX Server"

  Read this chapter for the in-depth details of booting client language systems—specifically HP BASIC/WS and Pascal Workstation system—from the server's mass storage. This chapter covers client booting from both the server's and client workstation's perspectives and provides enough of a common language for client users and server administrators to communicate about the booting process. It provides a model of the booting process first and then provides the step-by-step procedures for both the server administrator and client user to perform. The procedures lead you through getting a client to boot from the server the first time and then adding the remaining clients to the booting environment.

- (C) Chapter 6: "Installing and Configuring HP BASIC/WS Clients"

  Read this chapter if you will have HP BASIC/WS clients on your SRM/UX system. It tells you how to use the server's SRM/UX file system and spooling environment from your HP BASIC/WS client.

- (C) Chapter 7: "Installing and Configuring Pascal Workstation Clients"

  Read this chapter if you will have Pascal Workstation System clients on your SRM/UX system. This chapter tells you how to establish access to the SRM/UX server and create the required Pascal directory structure on the SRM/UX file system. It discusses designating your client workstation's system volume and creating links to the system files. It also tells you how to add driver modules to the initialization library and how to compile, run, and verify modified CTABLE programs. Finally, it details system customization options.

- (SC) Chapter 8: "Migrating Applications and Data from SRM to SRM/UX Using *srmdrestore*(1)"

  Read this chapter if you will be migrating your SRM files over to SRM/UX. This chapter discusses the command *srmdrestore*(1) which you can use to restore SRM files from backup tapes onto your SRM/UX file system. It details the options to *srmdrestore*(1) and gives examples of how you would use it to catalog a backup tape and restore the files from it.

- (SC) Chapter 9: "SRM/UX System Examples"

  Read this chapter to follow the examples of setting up actual SRM/UX systems. You will see, step-by-step, all of the processes involved in establishing, configuring, and running three SRM/UX systems. The first example is basic and will get you started on all of the issues involved in establishing a "real" system. Then, the second example brings in more client workstations and adds some complexity. Finally, the last example is a complex, multiple server, multiple client system that exposes you to more advanced issues and guides you through setting up the complex system.

- (C)Appendix A: "Porting BASIC and Pascal Application Programs from SRM to SRM/UX"

  Read this appendix if you need to port either HP BASIC/WS or Pascal Workstation System application programs from your SRM environment to the SRM/UX environment.

- (SC) Appendix B: "Troubleshooting Your SRM/UX System"

  Read this appendix if you have problems when trying to set up the SRM/UX system. It provides troubleshooting guidance, mostly from the client user's perspective—but with plenty of useful information for the SRM/UX System Administrator as well. The possible problems are organized into four major categories: general information, client booting problems, file system problems, and file spooling problems.

- (S) Appendix C:"SRM/UX Reference"

Read this appendix for reference purposes. It contains the SRM/UX command "man pages".

- (SC) There is a Glossary and Index at the back of the manual to help you quickly locate specific information.

RecommendedP aths Through This Guide

For the SRM/UX System Administrator

Read the chapters in the following order:

- Chapter 1
- Chapter 2
- Chapter 3
- Chapter 4
- Chapter 5
- Chapter 9

These chapters will give you the core information you need to administer the SRM/UX system.

Then you can read chapter 8 if you need to port SRM files over to SRM/UX, Appendix B (Troubleshooting hints) if you have problems with the system, and Appendix C for reference (man pages) for the SRM/UX commands.

You can also read chapters 6 and 7 to understand more about the client language systems (for example, to understand the client terminology, what the client user's needs are, and how they will be using the system).

For the User of an HP BASIC/WS Client

Read the chapters in the following order:

- Chapter 1
- Chapter 2
- Chapter 5
- Chapter 6

These chapters will give you an understanding of the overall SRM/UX system and how your client workstation fits in to the environment. They will show you how to boot your client workstation from the server, use the SRM/UX file system, and spool files to the spooling devices on the system.

For a more in-depth understanding of the system and its capabilities, you could read about the server in chapter 4 and look at the complete SRM/UX system examples in chapter 9.

Finally, if you will be porting your HP BASIC/WS programs from the SRM environment to the SRM/UX environment, then you should read Appendix A.

For the User of a Pascal Workstation System Client

Read the chapters in the following order:

- Chapter 1
- Chapter 2
- Chapter 5
- Chapter 7

These chapters will give you an understanding of the overall SRM/UX system and how your client workstation fits in to the environment. They will show you how to use your client workstation to boot from the server, to use the SRM/UX file system, and to spool files to the spooling devices on the system.

For a more in-depth understanding of the system and its capabilities, you could read about the server in chapter 4, and look at the complete SRM/UX system examples in chapter 9.

Finally, if you will be porting your Pascal Workstation System programs from the SRM environment to the SRM/UX environment, then you should read Appendix A.

SRM/UX System Requirements

Hardware Requirements

The SRM/UX server will run on any Series 300, 400, or 800 HP 9000 which runs HP-UX and has its own local root file system.

An HP 9000 Series 300 server is capable of supporting both LAN and SRM interfaces simultaneously. Series 400 and 800 servers support only LAN interfaces.

HP 9000 Series 200 and 300 workstations will run as SRM/UX clients.

Software Requirements

The SRM/UX server must be running HP-UX version 7.0, or later.

The HP BASIC/WS clients must be running HP BASIC/WS version 6.1, or later. (The upgrade from version 6.0 to 6.1 is included with the SRM/UX product.)

The Pascal Workstation System clients must be running version 3.23, or later, together with the extra modules required to make the Pascal Workstation System run with SRM/UX.

2

SRM/UX Orientation and System Planning Guide

This chapter will give both SRM/UX System Administrators and client users an overview of the entire SRM/UX system. It describes each of the system's component parts and briefly lists and describes the SRM/UX System Administrator's tasks. It tells you what is involved in moving from an SRM system to and SRM/UX system. Finally, it provides a system planning guide with tools you can use to organize your system.

What Was SRM?

The SRM system was made up of three major component types. They were:

- clients
- servers
- SRM cabling configurations

The SRM server was a dedicated server hardware platform (HP 50960A or HP 9000 Model 226 9826A) with its own operating system, which ran software that managed the data communication links to the client workstations. The server handled all mass storage and I/O activities related to the operation and use of shared system resources.

The SRM server's three main functions were:

- client language-system booting
- SRM file system management
- client file-spooling management

SRM clients were applications, such as HP BASIC/WS, that made use of the server's shared resources.

The SRM System Administrator was responsible for doing backups on the server's system disc(s).

The SRM cabling configurations came in two types:

- coax network configuration
- multiplexer configuration

What Is SRM/UX?

SRM/UX is **Shared Resource Management** (SRM) on the HP-UX operating system. It emulates the SRM functionality, but provides expanded capabilities. The rest of this chapter and manual describes the SRM/UX system in detail.

The HP-UX Operating System

All of the discussion in this chapter and manual assumes that you are starting with a fully configured HP-UX system on your server workstation(s). That is, the server computer has been configured, at least, as an HP-UX version 7.0 or later system and all of the peripherals have been installed and are operating (such as discs, printers, and plotters).

This manual also assumes that the you, the SRM/UX System Administrator, have been trained in HP-UX system administration, either through formal training or through on-the-job training.

When HP-UX terminology appears in this manual, it is consistent with the HP-UX manuals. See the Glossary in the *HP-UX System Administration Concepts* manual for the definition of HP-UX terms that appear throughout this manual, such as cluster, file system, superuser, kernel, device file, and daemon.

The SRM/UX System

The **SRM/UX system** refers to the full combination of server and client computer, peripheral, and interconnection hardware, together with all of the software that performs remote client booting, SRM/UX file system management, and SRM/UX client spooling, as well as the client language systems software. (See the SRM/UX system schematic in Figure 2-3 later in this chapter for a graphical description of a small SRM/UX system.)

The SRM/UX system provides the same capabilities as the SRM, with some additional capabilities.

Like SRM:

- It allows HP BASIC/WS and Pascal Workstation System users to share data, test results, etc. through file sharing.
- It allows central system administration so that a smaller number of people will require training as system administrators.
- It reduces costs because system resources are shared. That is, only one set of printers, plotters, and disc drives is necessary for many users, rather than a set for each user.
- It allows flexible use of computer workstations. Because you can install both client language systems (i.e., HP BASIC/WS and Pascal Workstation System) on a server, you can boot either one on the same client workstation, depending on the users' needs.

Additionally:

- It provides LAN connectivity.
- It provides access to remote file systems through Network File System (NFS), an industry-standard networking tool available through HP-UX. This capability is provided entirely by the HP-UX operating system on the SRM/UX server and is completely transparent to the HP BASIC/WS and Pascal Workstation System clients and to the SRM/UX process itself. It provides access to both HP-UX and non-HP-UX computers, including computers which don't themselves support SRM/UX. It is independent of which interface card, LAN or SRM, the client uses to access SRM/UX.

- It allows the use of named pipes on HP BASIC/WS clients (but not Pascal Workstation Clients)

Thus, SRM/UX's added benefits are:

- It provides improved performance because LAN has a higher data rate than SRM.
- It opens up network connectivity to HP BASIC/WS and Pascal Workstation System clients through the HP-UX networks.

The SRM/UX system is made up of four component types. They are:

- Clients
- Servers
- SRM connections
- LAN connections

The following sections discuss each of these component types to provide you with a complete picture of the SRM/UX system.

Clients

The **clients** are why SRM/UX exists. The server exists only to service requests for the shared resources from the clients. The clients for SRM/UX are the HP BASIC/WS and the Pascal Workstation System.

An **SRM/UX client** workstation is an HP 9000 Series 200 or 300 computer, with either an SRM or a LAN interface installed. The client must:

- Have an SRM interface card if on the SRM connection, and/or a LAN interface card if on the LAN connection.
- Run as either an HP BASIC/WS or Pascal Workstation System.
- Have entries in the server's configuration file (/etc/srmdconf) which accurately represent its status in the SRM/UX system.

Note Throughout this manual, SRM/UX clients are identified in several different ways, such as "client", "client workstation", "LAN client", "SRM client", etc. Most of the time it is not necessary to differentiate between the terms. However, because it matters in a few places to be technically precise, a "client" is ultimately an interface in a workstation: either an SRM interface card or a LAN interface. That is, a workstation that contains two interfaces (one SRM interface card and one LAN interface) can technically be thought of as two clients, each uniquely identified by its `Server Dev file` and `Client Node Num` pair (on SRM), and by its `Client Link Level Address` (on LAN).

No SRM/UX client can communicate directly with another SRM/UX client. To pass information from one SRM/UX client to a second, the first client must place the information in a shared file on the SRM/UX server's file system. The second client can then access the file to get the information. With the exception of accessing the SRM/UX shared resources, the SRM/UX client workstations act as independent computers.

To establish a working client you need:

- an HP BASIC/WS language system (version 6.1 or later) up and running on your client computer(s) (e.g., an HP 9000 Series 200 or 300), and/or
- an SRM/UX-capable Pascal Workstation System running on your client computer(s) (e.g., an HP 9000 Series 200 or 300)
- an SRM interface card if on an SRM connection and/or a LAN interface if on a LAN connection
- the correct entries in the server's configuration file, srmdconf, which specifies the SRM/UX file system access and spooling services available to this client
- the correct entries in a set of support files on the server which help client booting work within the HP-UX environment (e.g., /etc/boottab and /etc/srmdconf) if you are using a LAN connection.

Servers

The **server** is the resource manager of the SRM/UX system. The server controls client access to the shared resources.

The server has three primary functions:

- Client language-system booting You can install all of the files needed to boot and run both HP BASIC/WS and Pascal Workstation System clients on the server. Once this is done, the client users don't need to boot up their systems from the local mass storage each time. Also, you have the flexibility to boot all clients into the same system configuration, or into unique system configurations.
- SRM/UX file system management The server manages the **SRM/UX file system** as described in the section below.
- Client file-spooling management The server manages the spooling of files to printers and plotters from HP BASIC/WS and Pascal Workstation System clients.

Note The SRM/UX server workstation need not be dedicated to SRM/UX; it can be used simultaneously as an SRM/UX server and for other HP-UX applications as well.

To establish a working server you need:

- an HP-UX version 7.0 (or later) operating system running on your server computer (e.g., an HP 9000 Series 300).
- an SRM interface card if serving an SRM connection and/or a LAN interface if serving a LAN connection (chapter 4 provides details on what this involves).
- the server program, *srmd*(1M), which runs as a background HP-UX daemon process.
- the configuration file, srmdconf, which specifies the SRM/UX file system access and spooling services available to the clients.
- the correct entries in a set of support files which help *srmd*(1M) work within the HP-UX environment (e.g., /etc/boottab and /etc/services) if using a LAN connection.

Only one server process *srmd*(1M) can be running at one time on each server.

There can be multiple servers in the SRM/UX system. If there are multiple servers, then the overall SRM/UX system becomes an interconnection of SRM/UX subsystems. A subsystem is simply a single-server SRM/UX system—defined above—that is a small constituent of a larger SRM/UX system. You should configure each subsystem as you would a single-server system; that is, with clients on one or multiple SRM connection(s), with clients on one or multiple LAN connection(s), and with a set of shared resources.

Shared Resources

Each server in the SRM/UX system has a set of "resources" which it makes available to its clients. The resources are: the **SRM/UX file system**, and the **SRM/UX spooling environment**.

Both the SRM/UX file system and the SRM/UX spooling environment are defined in the sections below.

The SRM/UX File System. The SRM/UX file system is defined from two different perspectives. From the client's perspective the SRM/UX file system is a set of emulated SRM volumes. From the server's perspective the SRM/UX file system is one subtree or a set of subtrees of the HP-UX file system that is presented to the SRM/UX clients. The SRM/UX file system is established by the SRM/UX System Administrator under the HP-UX root directory and is configured into the mass storage of the HP-UX server (host) computer.

You can create the SRM/UX root directory anywhere under the HP-UX root directory that makes sense for your particular site. That is, you might wish to place the SRM/UX root directory at /srmroot or /srmuxroot, or it may fit your requirements better to place it at another location, such as /usr/local/srmuxroot.

The following figures show an example HP-UX file system and an example SRM/UX file system.

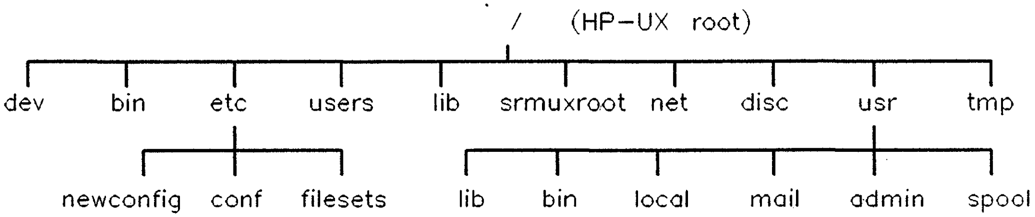


Figure 2-1. Example HP-UX File System

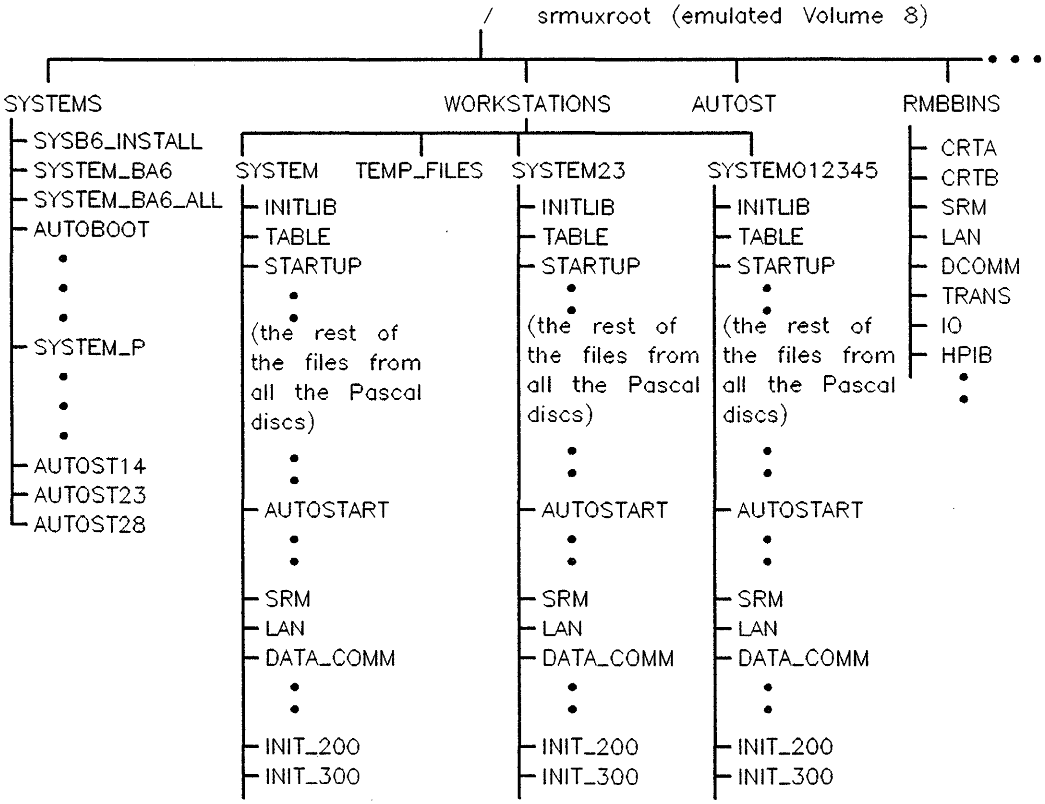


Figure 2-2. Example SRM/UX File System

The server process, *srmd*(1M), manages the SRM/UX file system based on the information you give it in the configuration file, /etc/srmdconf. That is, the SRM/UX System Administrator controls, through the entries in /etc/srmdconf, which HP BASIC/WS and/or Pascal Workstation System clients can access which directories (i.e., emulated SRM volumes) on the HP-UX file system.

The SRM/UX Spooling Environment. The SRM/UX spooling environment involves using the existing HP-UX spooling environment to emulate SRM spooling. The SRM/UX server supports printer and plotter spooling.

To properly configure your SRM/UX spooling environment, first you must configure your HP-UX spooling environment, and then configure the SRM/UX spooling environment to pass files to the HP-UX spooling environment. To learn how to configure your HP-UX spooling environment, refer to the *HP-UX System Administration Tasks* manual.

Both Pascal Workstation System and HP BASIC/WS client users do spooling to SRM/UX exactly as they do for SRM.

To use a spooler, a client user places a copy of the file to be printed or plotted into a spooler directory (for priority printing, the file is placed into a special "PRIORITY" subdirectory in the spooler directory). Any SRM/UX data file may be spooled. (Non-HP-UX files are handled by *srmdlpfilter*(1); see chapter 4 of this manual for details.)

A spool file is automatically sent to the printer or plotter associated with the spooler directory when the file meets the following criteria:

- The file is not empty.
- The file is closed.
- The file is either:
  - □ the head of the queue of files in the PRIORITY subdirectory of the spooler directory.
  - □ the head of the queue in the spooler directory and the PRIORITY queue is empty.

SRM Connections

One of the cabling schemes available in the SRM/UX system is the **SRM connection**. There are two supported varieties of SRM connections: the SRM coax connection and the SRM multiplexer connection. You can have multiple SRM connections on each server in your SRM/UX system. Each SRM coax connection consists of the server's SRM interface card, the cabling, connectors, and terminators, and the client workstations' SRM interface cards and adapters (if applicable). Each SRM multiplexer connection consists of the server's SRM interface card, the multiplexers, cabling, connectors, and terminators, and the client workstations' SRM interface cards and adapters (if applicable).

Chapter 3 of this manual provides details on installing the SRM connection.

LAN Connections

Although the SRM connection was the only available connection with the SRM product, SRM/UX provides the added capability of the LAN connection. You can have multiple LAN connections on each server in your SRM/UX system. Each LAN connection consists of the server's LAN interface, the coax cabling (ThinLAN or ThickLAN), connectors, and terminators, and the client workstations' LAN interfaces.

See chapter 3 of this manual for SRM/UX-specific details on the LAN connection and the "Inter-System Communication" chapter of the *HP 9000 Workstations Configuration Guide* for details on how to install your LAN interfaces and LAN cabling.

SRM/UX System Administration Tasks

The following is a brief overview of the SRM/UX System Administrator's responsibilities. Each of these SRM/UX system administration tasks is described in detail in the appropriate places throughout this manual. Whenever you need to perform the actual procedures, detailed instructions are provided at that time. Also, you can find the location of the details on each of these tasks by looking up keywords in the Index and Table of Contents.

Backups

The SRM "front panel" (i.e., the SRM server's command set as presented on its computer monitor) provides a backup facility. SRM/UX does not provide a specific backup capability; instead the underlying HP-UX backup facility should be used to provide whatever level of backup is required. If the SRM/UX System Administrator and the HP-UX System Administrator are two different people at your site, then they should discuss the SRM/UX system backup needs and plan accordingly.

See the *HP-UX System Administration Tasks* manual for instructions on how to perform HP-UX backups.

SRM/UX System Organization

The SRM/UX System Administrator configures the SRM/UX system based on the needs of the client workstation users at his or her site. The major variables involved are: the number of client and server workstations and what model of HP 9000 they should be, whether they are HP BASIC/WS or Pascal Workstation System clients, which connections (SRM or LAN) suit the existing environment, and what the shared resources will be.

Hardware Installation

The SRM/UX System Administrator plans and installs the SRM/UX system hardware. As mentioned at the beginning of this chapter, the HP-UX System Administrator should have already installed and configured the server workstation(s), the peripherals, and all of the underlying HP-UX system utilities to make the hardware work.

The hardware to be installed by the SRM/UX System Administrator consists of:

- Client workstations.
- Server and client interfaces (SRM and/or LAN).
- Connections (SRM and/or LAN).

Booting HP-UX on the Server(s)

If the SRM/UX and HP-UX system administration tasks will be performed by one person at your site, then he or she will be responsible for installing and booting-up HP-UX on the server computer(s). If the tasks will be performed by different people, then the HP-UX System Administrator should boot HP-UX version 7.0 or later on the server(s).

Customizing the Server's HP-UX Environment

You may need to customize the HP-UX kernel to handle SRM/UX. You should configure SRM and/or LAN drivers into the kernel—depending on the connections you are supporting—and if you will be using remote file systems, you must include the Network File System (NFS) driver in the kernel.

You must also ensure that the HP-UX kernel includes drivers for every printer and plotter you will be using in your system. See the "Reconfiguring the Kernel" chapter in the *HP-UX System Administration Tasks* manual for details.

Converting Discs to Long File Names

To ensure compatibility with SRM, we recommend that you convert all of the discs used by the SRM/UX file system to long file names. (See the *convertfs*(1M) man page and the "Managing the File System" chapter of the *HP-UX System Administration Tasks* manual for details on this subject.)

Establishing the SRM/UX File System

Part of the job of organizing the SRM/UX system involves defining the SRM/UX file system for your site. For details on the SRM/UX file system, see the SRM/UX system section towards the beginning of this chapter and chapter 4.

Obtaining and Assigning SRM/UX System Parameters

The SRM/UX System Administrator must obtain some parameters from the client workstations, such as the LAN interfaces' Link Level Addresses, and about the client workstations, such as the client Internet Address. The SRM/UX System Administrator must also assign some parameters to the client workstations, such as SRM client node addresses, LAN (emulated) node addresses, and client names. (The SRM/UX System Administrator will most likely obtain the entire client Internet Address from the HP-UX System Administrator.)

Configuring the SRM/UX Server

The vehicle for configuring the SRM/UX server is the configuration file: /etc/srmdconf. By filling out the tables in srmdconf, you are telling the server process, *srmd*(1M), the composition and details of the system that the server will be serving.

The srmdconf file is an HP-UX text file that you create or modify with an editor. A template srmdconf file is placed in your server's `/etc` directory and also in the /etc/newconfig/srmux directory by the SRM/UX installation process. There are five `tables` in srmdconf, that is, sections of the file which are organized into formatted tables which contain the configuration details *srmd*(1M) needs to run the SRM/UX system. The `tables` are:

- VOLUME-TABLE
- LAN-CLIENTS
- SRM-CLIENTS
- SPOOL-ENVIRONMENT
- SPOOL-TABLE

See chapter 4 of this manual for the details on /etc/srmdconf.

Establishing the SRM/UX Spooling Environment

SRM/UX uses the HP-UX spooling environment to emulate an SRM spooling environment for the SRM/UX client workstations. After the HP-UX spooling environment is ready, then the SRM/UX System Administrator must establish the spooling environment by adding the proper entries to the SPOOL-TABLE and modifying the SPOOL-ENVIRONMENT table in /etc/srmdconf.

Installing Client Booting

Installing client booting on the server involves performing procedures on both the server's side and the client's side.

It is not necessary to shut down the server's HP-UX system to install client booting.

You can boot both HP BASIC/WS and Pascal Workstation System clients from the server. You can boot these clients over the SRM connection and over the LAN connection.

See chapter 5 of this manual for details on client booting from the server's side, and from the client's side.

Dynamically Reconfiguring the Server

Some typical reasons for reconfiguring the server are: adding a new client, deleting a client, modifying the parameters of an existing client (for example, adding a new volume to a client's Volume List in either the LAN-CLIENTS or SRM-CLIENTS table of srmdconf), changing a spooling directory, adding or deleting a printer or plotter, etc. SRM/UX lets you make changes to individual client's configurations without disturbing all of the users on the overall SRM/UX system; that is, you can make the changes while the server process is still running.

To reconfigure a running server (also called dynamic reconfiguration) you simply make the changes in each relevant field of the appropriate tables in the /etc/srmdconf configuration file and then run *srmdconfcheck*(1M) to check the correctness of your entries, followed by *srmdreconfig*(1M) to do the reconfiguration.

See the next section for an introduction to *srmdconfcheck*(1M), *srmdreconfig*(1M), and the other "SRM/UX Front Panel Commands".

Using SRM/UX Front Panel Commands

SRM/UX front panel commands are executed from the server at the HP-UX shell prompt.

srmdrestore(1)

You should use *srmdrestore*(1) if you want to use your SRM files on SRM/UX. It moves the files from an SRM backup tape (SRM version 3.0 or later only) to the SRM/UX server's file system.

srmdstat(1)

You use *srmdstat*(1) to check on the status of the SRM/UX clients. It reports such things as whether clients are "active" or "not active", which clients have open files, and which clients have files locked.

srmdconfcheck(1M)

You run *srmdconfcheck*(1M) to verify your initial srmdconf file, and every time you make changes to srmdconf. Make a copy of the srmdconf file, make your changes, run *srmdconfcheck*(1M) on the copy, and then move the copy to the "real" /etc/srmdconf file. Using this method will keep you from adversely impacting the clients with mistakes in /etc/srmdconf.

srmdreconfig(1M)

You can run *srmdreconfig*(1M) to reconfigure a running server without shutting it down. *Srmdreconfig*(1M) tells the SRM/UX daemon process, *srmd*(1M), to reread the /etc/srmdconf file and reconfigure itself, without shutting down and restarting.

srmdreset(1M)

You can run the *srmdreset*(1M) command to unlock and close open files for the clients specified in the command line of *srmdreset*(1M). For example, if a client has malfuctioned (say, due to a hardware failure) and the malfunction is tying up shared resources that other clients are waiting to use, then the SRM/UX System Administrator can use *srmdreset*(1M) to free the shared resources.

srmdshut(1M)

You can execute the *srmdshut*(1M) command to do a "graceful" shutdown of the SRM/UX daemon process, *srmd*(1M). It replaces the SRM's SYSTEM DOWN command. For details on what a "graceful" shut down means, see the "Stopping the Server Process" section in chapter 4 of this manual.

From SRM to SRM/UX

If you have an operating SRM system that you want to upgrade to SRM/UX, then you will find this section helpful. If you are starting an SRM/UX system from scratch, then proceed to the "System Planning Guide" section later in this chapter.

SRM/UX is a functional replacement of the SRM. Most of the SRM functionality is provided by the SRM/UX server; other functionality is provided more directly by HP-UX, especially in the areas of configuring and administering the server. SRM/UX has capabilities that SRM didn't have:

- You can access remote file systems using NFS (Network file system)
- You can use LAN connections in addition to SRM connections
- You can use pipes on HP BASIC/WS clients

The SRM/UX server emulates SRM operations, such as opening files, writing and reading data to and from files, etc. by performing the corresponding operations on the SRM/UX file system. Most operations are functionally identical with SRM from the HP BASIC/WS and Pascal Workstation System clients' perspectives.

Many of the SRM commands have no direct SRM/UX equivalent (for example, MOUNT and UNMOUNT), but the tasks they accomplish are handled by the server's underlying HP-UX operating system. The table below shows which SRM/UX front panel commands accomplish the functions of the remaining SRM commands.

Table 2-1. SRM Commands in SRM/UX

SRM Command	SRM/UX Command
REMOVE_USER	*srmdreset*(1M)
SYSTEM DOWN	*srmdshut*(1M)
FILES USERS NODES	*srmdstat*(1)
RESTORE	*srmdrestore*(1)

The SRM server hardware and operating system have been replaced by an HP-UX process named *srmd*(1M) which runs in the background on an HP 9000 computer that is running as a multi-tasking HP-UX workstation. As a result, under SRM/UX, some SRM system administration tasks (such as backups) have been moved into the domain of HP-UX system administration.

Controlling File Access (Passwords)

The main difference between SRM and SRM/UX from the HP BASIC/WS and Pascal Workstation System clients' perspectives is how the server controls file access. That is:

- SRM uses a password-per-file scheme, with each password controlling up to 6 different properties.
- SRM supports multiple passwords per file or directory, each with different property control sets. The SRM file access control properties are unrelated to the HP-UX read/write/execute file protection properties; that is, SRM passwords and the HP-UX file protection scheme are not compatible. (See the "File System" chapter in the *HP-UX System Administration Concepts* manual for a complete discussion of HP-UX file protection.)

  SRM/UX uses the HP-UX file protection properties, NOT the SRM file protection properties.
- The HP-UX file protection properties must be observed by SRM/UX to protect the non-SRM/UX HP-UX files and processes running on the server

platform. The SRM/UX server ignores SRM passwords specified in path names. Therefore, the client will not have to modify any applications if they presently use passwords in path names, with the exception of HP BASIC/WS application programs that use the PROTECT statement.

Long File Names

SRM supports file names up to 16 characters long. The default HFS file system under HP-UX has 14 character file names. Therefore, to ease the migration of applications and data from SRM to SRM/UX, HP recommends converting all of the HP-UX discs that contain files that are accessed by SRM/UX clients to a long-file-name system by using *convertfs*(1M). Refer to the *HP-UX Reference Vol 3:Sections 1M,5,7, and 9* manual for details on *convertfs*(1M). Also, read the discussion on converting to long file names in the "Managing the File System" chapter of the *HP-UX System Administration Tasks* manual.

The SRM/UX server hides file names longer than 16 characters from the clients.

Note HFS disks configured for long file names cannot be connected directly to Pascal Workstation System or HP BASIC/WS systems because their HFS drivers support only short-file-name HFS disks.

File Locking

On a given SRM/UX server, a client can "lock" a file, and other SRM/UX clients on that server will not be able to access it; for example, an HP BASIC/WS user can lock a file using the HP BASIC/WS *LOCK* statement (See the *HP BASIC Language Reference* for details on *LOCK*). This emulates SRM functionality exactly. However, the SRM/UX server does NOT prevent other HP-UX processes from accessing the file.

Note The file lock will work only if the clients involved are accessing the file through the same SRM/UX server. If one client has the file locked on one server, then a client of another server can access the same file through an NFS connection.

On the other hand, HP-UX applications can open files exclusively to lock out SRM/UX clients by using the open(2) and lockf(2) system calls. Refer to the *HP-UX Reference Vol 2:Sections 2 and 3* manual for details.

Accessing Remote File Systems

Through the Network File System (NFS), an industry standard networking tool available through HP-UX, the SRM/UX server can mount remote file systems. This added capability means that parts of or all of the SRM/UX file system may be located on mass storage that is physically remote from the SRM/UX server.

As SRM/UX System Administrator, you should communicate with the HP-UX System Administrator about files you ported from SRM to SRM/UX. You want to ensure that all of those files work on SRM/UX; so it may be necessary to convert the remotely mounted mass storage to long file names.

Porting Files From SRM Backup Tapes To SRM/UX

There is a way to use your SRM files and data on SRM/UX: you can bring the files over (port them) using a command called *srmdrestore*(1). You can restore the files you have on SRM version 3.0 or later backup tapes.

Use *srmdrestore*(1) after your SRM/UX server is installed and configured.

See chapter 8, *Migrating Applications and Data from SRM to SRM/UX* for a detailed description of *srmdrestore*(1).

System Planning Guide

Before you build your SRM/UX system you must plan it, so that it will meet your user's needs. This guide will help you with that task. With the above SRM/UX orientation you are ready to make the decisions about how you will configure your SRM/UX system.

SRM, LAN, or Both Connections?

Does your SRM/UX system include SRM connections, LAN connections, or a mix of both?

Be aware that the LAN connection has a higher data rate than the SRM connection, which results in higher performance for SRM/UX operations. As a result, HP recommends using the LAN connection for all new SRM/UX systems which don't contain HP 9000 Series 200 clients.

Servers

This manual assumes that the HP-UX System Administrator at your site has already installed the server workstation and all of its peripherals. As the SRM/UX System Administrator, you should identify what peripherals have been installed on the server.

The supported workstations that can be used as servers are: HP 9000 Series 300, Series 400, and Series 800.

Your SRM/UX system should contain the correct number of servers to handle the user's needs. Some considerations for how many servers you need in the system follow:

- program and data isolation (security)
- loading—how many SRM/UX clients, and other HP-UX processes will be running on the server
- physical proximity of the resources
- the need for redundancy

Decide on how you are going to install the SRM/UX file system (i.e, what will its structure be) on the server's HP-UX file system.

Clients

The supported workstations that you can use as clients are HP 9000 Series 200, and Series 300. The only Series 200 workstations you can use over the LAN connection are Model 226 and Model 236. However, you can not boot over LAN from any Series 200 workstation.

Identify how many HP BASIC/WS clients you will have. Identify how many Pascal Workstation System clients you will have.

Completing the Plan

After you have made all of the above decisions, more must be done to bring the SRM/UX system plan together: you must assign some parameters and collect the rest of the parameters for the clients and servers on both the SRM connection(s) and the LAN connection(s).

This section provides four tools to help you plan your system: the SRM/UX system schematic, the SRM/UX Client Descriptions worksheet, the SRM/UX SRM Worksheet, and the SRM/UX LAN Worksheet. Their use is described in the "System Planning Tools" section later in this chapter.

On the SRM connection you must:

- Assign unique client node addresses.
- Assign unique server node addresses. (Node addresses are physically set with switches on the SRM interface cards.)
- Assign Uids, Gids, and Umasks for each client.
- Assign unique client names.
- Assign specific volumes within the SRM/UX file system that each client will be allowed to access.
- Obtain the name of each SRM interface device file that will be used in each server.

On the LAN connection you must:

- Obtain each client interface's unique link level address from the client's Boot ROM. (For Series 300s: it is displayed on the monitor after cycling the power. For Series 200s: it is read from a label on the interface card.)
- Obtain or assign each client's Internet Address from your site's allocated Internet Addresses. (See your HP-UX System Administrator or the HP-UX manuals and/or the server's /etc/hosts file.) Note: Clients must be on the same LAN subnet as the server.
- Assign Uids, Gids, and Umasks for each client.

- Assign unique emulated client node addresses.
- Assign unique emulated server node addresses.
- Assign unique client names.
- Assign specific volumes within the SRM/UX file system that each client will be allowed to access.

Note HP recommends setting up a simple, single-server, single-client system first. Then after you have set the simple system up, you can proceed to bring up the entire SRM/UX system more easily and successfully.

The following subsections give you some guidelines for accomplishing the above tasks on either or both of the system connections.

Guidelines for Assigning Server and Client Node Addresses

Note The server's allowed ranges are 0..126 for server node addresses and 1..126 for client node addresses. However, HP BASIC/WS can only access servers with server node addresses in the range 0..63. (HP BASIC/WS can access node-specific AUTOST files in the full range of client node addresses, e.g., /SYSTEMS/AUTOST126).

With the above restrictions in mind, the following guidelines for assigning node addresses on SRM connections and LAN connections are adequate for most systems and ensure reliable operation, with minimum complexity and confusion:

- Assign a unique node address to each server in the system, beginning at node address 0 (zero). Use incrementing values (0,1,2,3, ...) for each additional server on the SRM/UX system.
- Assign unique node addresses for each client workstation, beginning at address 10 (assuming there are no more than 10 servers in the system). Assign addresses from 10 to a maximum of 63 (for the SRM connection) and 99 (for the LAN connection), one address for each workstation. (Assigning addresses higher than 99 may cause problems with compatibility of client applications.)

- Although it is not recommended, duplicate node addresses can be assigned in the same SRM/UX system, provided that no two computers having identical node addresses are connected to the same SRM connection. If you ever move a computer or rearrange the SRM or LAN cabling, you run the risk of a system malfunction if you have assigned duplicate node addresses.
- If any server or client workstation has more than one interface card, then all interfaces in the workstation may be assigned the same node address, provided the address does not conflict with any other computer on the same connection.

Note Never assign node address 0 (zero) to any client workstation. That would cause a system malfunction, since address 0 is reserved as a default server address.

Guidelines for Creating Unique Node Names

SRM/UX node names are optional, but we recommend using them to make system administration easier. You can base the node names on what the workstation is being used for (i.e., a certain product), or on its model type, or on the user's name, or on any combination of all of these; for example:

- Product use: prodxyz
- Model type: 236
- User's name: fred_smith
- Combination: fred236xyz

The information in the server's /etc/hosts file and, if one exists on your system, the /etc/hosts.local file should help avoid collisions with Node Names assigned to other HP-UX computers.

Guidelines for Assigning Volumes to Clients

As the SRM/UX System Administrator you must decide which client workstations will have access to various pieces of the SRM/UX file system. You should set up a directory structure that places programs and files in locations that are logical and efficient. Then you can control which users access various pieces of the SRM/UX file system through the /etc/srmdconf file's VOLUME-TABLE, LAN-CLIENTS table, and SRM-CLIENTS table.

Guidelines for Assigning Uid, Gid, and Umask Values

- The `Uid` field appears in three tables in srmdconf: the VOLUME-TABLE, the LAN-CLIENTS table, and the SRM-CLIENTS table. The Uid field specifies the HP-UX User ID or Owner ID that SRM/UX assigns to all new files and directories that are created in a specific volume (when Uid is specified in the VOLUME-TABLE) or that are created by a specific client workstation (when Uid is specified in the LAN-CLIENTS or SRM-CLIENTS tables). If Uid is specified in both the VOLUME-TABLE and on the lines in the LAN-CLIENTS or SRM-CLIENTS tables, the value in the VOLUME-TABLE overrides. The Uid is used for all file permission checking. The Uid is also used for jobs submitted to the HP-UX print spooler by each client; therefore, you must specify a Uid value for each client in the LAN-CLIENTS and SRM-CLIENTS tables.
- The `Gid` field appears in the same three tables as Uid. It specifies the HP-UX Group ID that SRM/UX assigns to all new files and directories. The rules for the Gid field are the same as listed above for the Uid field.

  See the "File System" chapter of the *HP-UX System Administration Concepts* manual for details on setting effective user and group IDs.
- The `Umask` field appears in only two of the srmdconf tables: LAN-CLIENTS and SRM-CLIENTS. The value you specify for Umask sets the default file permission bits for all new files and directories that each client workstation creates through SRM/UX.

  See the "File System" chapter of the *HP-UX System Administration Concepts* manual and the *umask*(1) man page in the *HP-UX Reference, Vol 1* for details on setting the correct Umask.

System Planning Tools

Three tools are suggested to help you organize all of this information: the SRM/UX system schematic, the SRM/UX Client Descriptions worksheet, the SRM/UX SRM Worksheet, and the SRM/UX LAN Worksheet. The SRM/UX system schematic will be unique for each system. An example SRM/UX system schematic is shown in Figure 2-3. Templates are provided for the SRM/UX Client Descriptions worksheet, the SRM/UX SRM Worksheet, and the SRM/UX LAN Worksheets.

To construct an SRM/UX system schematic, do the following:

- Draw an SRM/UX system schematic.
- Lay out the system on the page with related components close to each other.
- Draw the server(s) and each SRM connection and LAN connection with all of the clients connected to each connection.
- Draw the peripherals connected to the server(s).
- Label each server with its SRM `Server Node Address` and/or LAN `Server Node Address`, and its workstation type (i.e., Model number).
- Label each client on the SRM connection with its `Client Node Name` and `Client Node Address`, and its workstation type (i.e., Model number).
- Label each client on the LAN connection with its `Client Node Name`, `Client Link Level Address`, and (emulated) `Client Node Address`, and its workstation type (i.e., Model number).

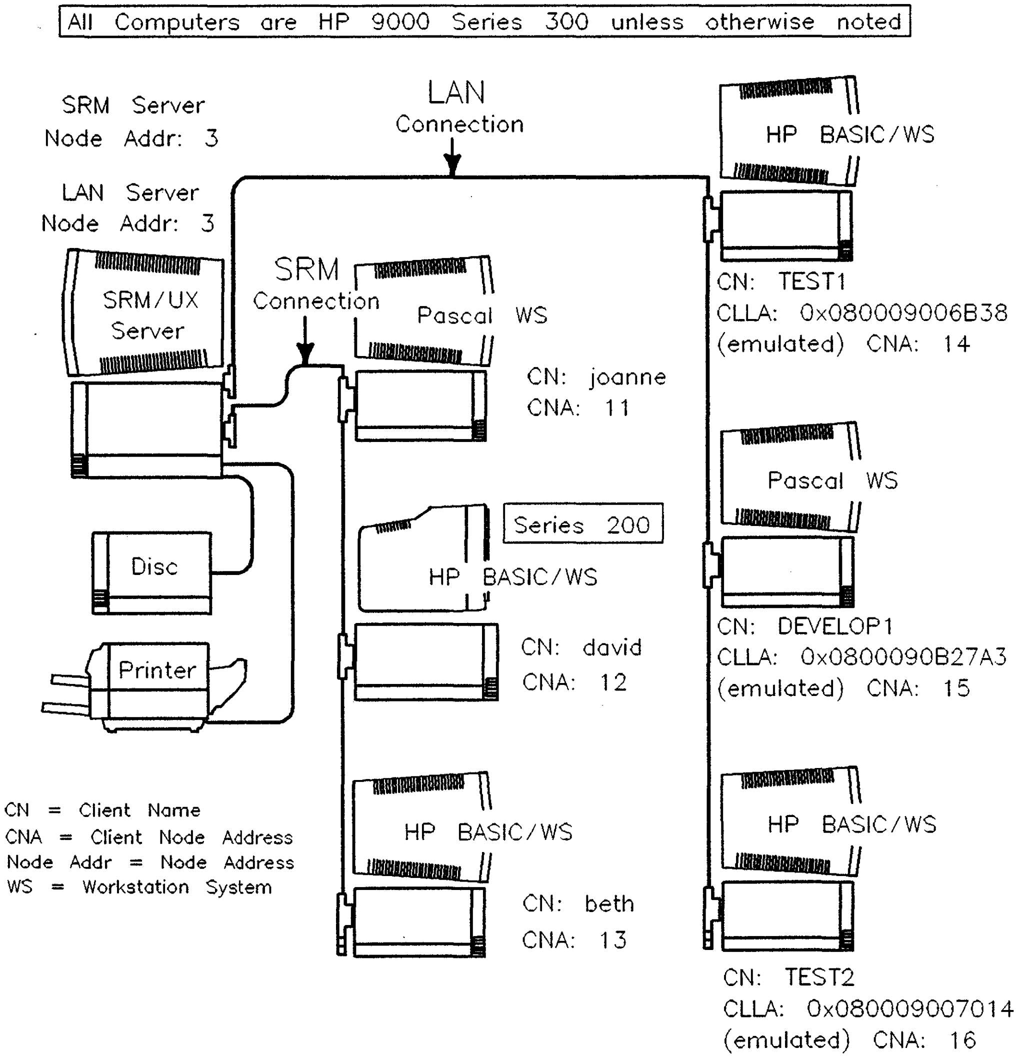


Figure 2-3. Example SRM/UX System Schematic

Next, make photocopies of the three worksheets provided here. Then if you need to make changes or expand your SRM/UX system, you will have blank worksheets to work with.

Fill out the SRM/UX Client Descriptions worksheet, the SRM/UX SRM Worksheet, and the SRM/UX LAN Worksheet by following the guidelines above. The worksheets provide an organized place to record the system parameters for all of your servers and client workstations. Walk around to each workstation and record all of the parameters while you are there and have easy access to the state of the hardware, and to such parameters as the LAN interfaces' `Client Link Level Addresses`, or the SRM interface cards' `Client Node Addresses`.

Then, when it comes time to edit the server's /etc/srmdconf configuration file, you can just copy all of the necessary information right off of the worksheets.

Table 2-2. SRM/UX Client Descriptions

SERVER NAME:			
Client Name	Comments	Connection Type	Volume List

Table 2-3. SRM/UX SRM Worksheet

<table>
<tr><td colspan="7">SERVER NAME:</td></tr>
<tr><td>Client Name</td><td>Server Node Address[1]</td><td>Server's SRM Device File</td><td>Client Node Address</td><td>Uid</td><td>Gid</td><td>Umask</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

[1] This field is not entered in the /etc/srmdconf SRM-CLIENTS table.

Table 2-4. SRM/UX LAN Worksheet

SERVER NAME:							
Client Name	Client Link Level Address	Client Internet Address	Client Node Address[1]	Server Node Address	Uid	Gid	Umask

[1] The Client Node Address and Server Node Address are emulated on the LAN connection

Where To Go From Here

Now that you understand the fundamental concepts of the SRM/UX system and have planned your specific system, we recommend reading through:

- Chapters 4, 5, and 6 if you have only HP BASIC/WS clients.
- Chapters 4, 5, and 7 if you have only Pascal Workstation System clients.
- Chapters 4, 5, 6, and 7 if you have both HP BASIC/WS and Pascal Workstation System clients.

to learn the details of the server and the clients.

Then, go to chapter 9 and follow the step-by-step instructions in the first single-server, single client (on both an SRM connection and a LAN connection) example SRM/UX system. After you get the simple system operating correctly it will be much easier to get your complete SRM/UX system running.

After the simple SRM/UX system example you will find more complex system examples in chatper 9 that will give you a more detailed understanding of all of the issues involved in bringing up a complete SRM/UX system.

The instructions in chapter 9 will point you to the appropriate sections of this manual and other HP manuals that discuss the concepts you should understand and tasks you must perform to set up your complete SRM/UX system.

3

Installing the Hardware

Hardware System Overview

Introduction

The SRM/UX server process will run on any HP 9000 Series 300, Series 400, or Series 800 computer which runs HP-UX version 7.0 or later and is a standalone workstation or a diskless HP-UX server.

A Series 300 server may support both LAN and SRM interfaces simultaneously, but the Series 800 and Series 400 servers support LAN interfaces only. LAN interfaces provide increased performance compared to the SRM cards.

You could replace an existing SRM server with a Series 300 HP-UX computer (with SRM card installed) and no other hardware changes would be required.

An Example SRM/UX System

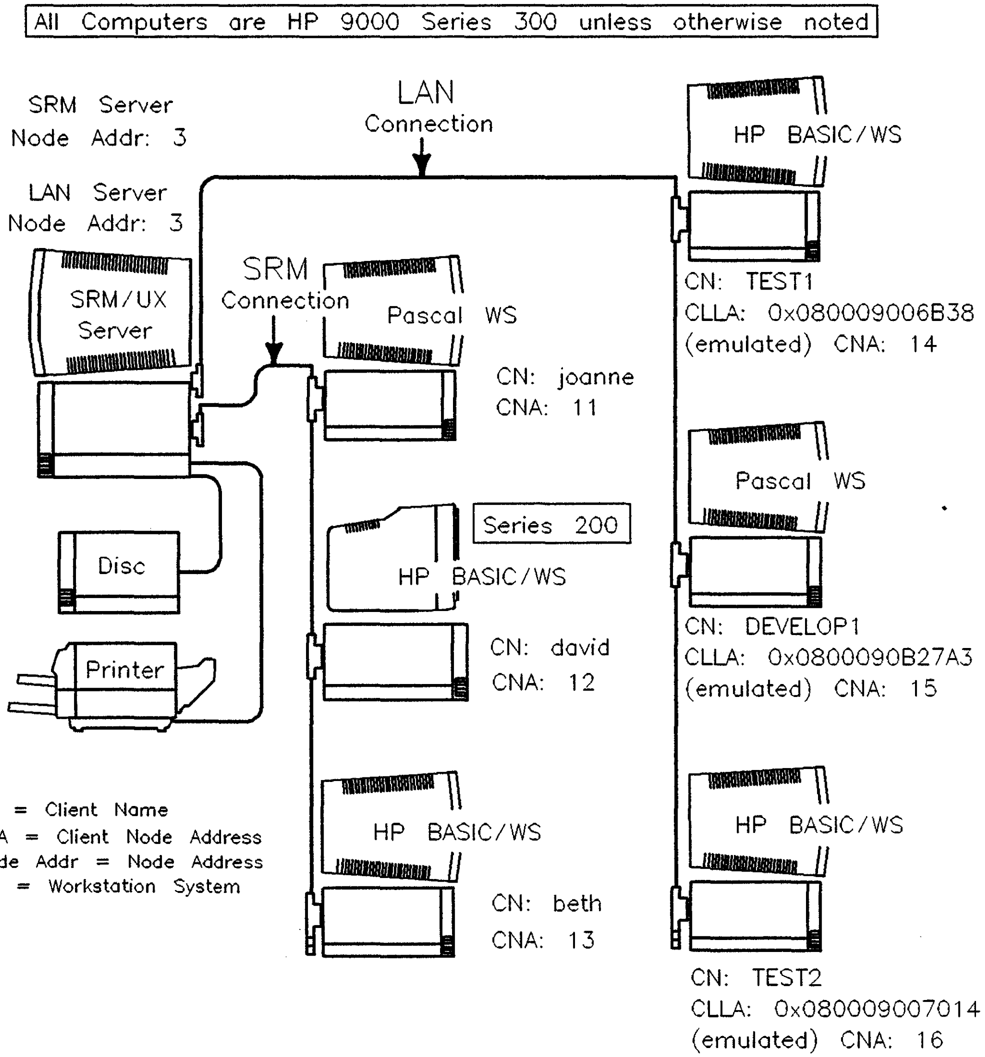


Figure 3-1. Example SRM/UX system (Series 300 server, both connections)

The System Hardware Components

The SRM/UX system is made up of four categories of hardware components: the server and client workstations, the SRM connection, the LAN connection, and the shared peripherals.

Communication over the SRM/UX system is achieved through either or both of two cabling schemes: the SRM connection and the LAN connection. For both of these cabling schemes, interfaces must be installed, one in the server and one in each client workstation for each connection to which they are attached (built-in LAN interfaces can be used on Series 300). The SRM/UX system can be made up of a number of SRM connections and/or a number of LAN connections. The limits on the number of interfaces are:

- On Series 300 servers: 9 SRM interfaces and/or 5 LAN interfaces
- On Series 800 servers: On versions 7.x of HP-UX, 3 LAN interfaces
- On Series 800 servers: On versions 8.x of HP-UX, 5 LAN interfaces

Note LAN and SRM coax cable must NEVER be connected together. The protocols and electrical specifications are different for the two types of coax cable.

Other HP-UX networking software may use the server's LAN interfaces simultaneously with the SRM/UX server process. However, the server acquires exclusive access to the SRM cards. In particular, the existing SRM Access Utilities (e.g., srmcp) cannot be used on the same SRM cards used by the SRM/UX server. (They may be used on other SRM cards in the same machine, which are not used by the SRM/UX server process.)

Throughout this manual, the terms "SRM card" and "LAN interface" mean any of the interfaces listed in the table below.

Table 3-1. Valid Interfaces

——	Series 300	Series 800	Series 400
SRM	98629A 98629 + 50961U 50962A	not supported	not supported
LAN	98643A 98171A built-in	36967A 91786B	98171A built-in

Server and Client Workstations

The heart of the SRM/UX system is the resource management server hardware-platform, running special software—the *srmd*(1M) daemon process—which manages the data communication with the client workstations. In the SRM/UX system, the term "server" typically refers to the entirety of the HP-UX platform running the *srmd*(1M) process. The server also handles all mass storage and I/O activities related to the operation and use of shared system resources. Unlike the SRM server, the SRM/UX server can be simultaneously used as an HP-UX workstation.

HP-UX Host Platforms that run as servers:

- HP9000 Series 300 which can have SRM cards and/or LAN interfaces installed
- HP9000 Series 400 which can have only LAN interfaces installed
- HP9000 Series 800 which can have only LAN interfaces installed
  - □ The Series 400 and Series 800 can act as a server only over the LAN connection because SRM card support is not included in HP-UX on those workstations. As a result, some Series 200 clients could not use a Series 400 or 800 as a server because there would be no way to connect them together.

Platforms that run as clients:

- HP 9000 Series 300

- HP 9000 Series 200
  - □ Some Series 200 computers don't support LAN interfaces; they must use SRM cards. LAN interfaces are supported only for 226's and 236's.

The SRM Connection

The SRM connection has two different configurations: the SRM coax connection and the SRM multiplexer connection, which are functionally identical. Both connections are supported.

For instructions on how to correctly install your SRM interface cards and SRM cabling, see the "Installing the SRM/UX System Hardware" section later in this chapter.

The LAN Connection

See the "Inter-system Communication" chapter of the *HP9000 Workstations Configuration Guide* for details on how. to correctly install your LAN interface cards and the LAN cabling.

The Shared Peripherals

The shared peripherals on the SRM/UX system which are attached to the server(s), will be installed and configured by the HP-UX system administrator at your site. Installing and configuring peripherals is no longer in the province of the SRM/UX system administrator, as it was for the SRM system administrator. The SRM/UX system administrator needs only to configure the SRM/UX system spooling environment to work within the HP-UX spooling environment. See chapters 2 and 4 for details about doing this.

The shared peripherals are connected to the SRM/UX server and include: printers, plotters, and disc drives. The number and kind of shared peripherals are dictated by user requirements. The number of disc/tape drives, printers, and plotters the SRM/UX server can support, is dependent on the constraints of your HP-UX system. See the *HP 9000 Workstations Configurations Guide* for details. Additional peripherals can be connected to individual client workstations for special needs, but they cannot be shared by other client workstations.

Glossary of Hardware Terminology

SRM: Stands for Shared Resource Management, typically used to refer to the SRM server (HP50960A), its operating system software, and the peripherals attached to the SRM server. SRM/UX is an emulation of the SRM on HP-UX.

SRM card (or **SRM interface**): Refers to one of the following HP SRM card part numbers: HP98629A, HP98629 + HP50961U combination, or HP50962A.

SRM Coax Adapter: Refers to the HP 50961 Resource Management Coax Adapter. This adapter attaches to an HP 98629 to allow the use of this SRM interface on a coax connection.

SRM coax connection: Describes the SRM hardware configuration that uses SRM coax interfaces or HP 50961 Resource Management Coax Adapters at the client and server workstations, and uses coax cabling to connect the client(s) and server(s) in a bus fashion.

SRM coax interface: Refers to the HP50962A SRM card. This interface is functionally equivalent to the HP98629 SRM card and HP50961U Coax Adapter combination, which is also used on a coax connection.

SRM multiplexer connection: Describes the SRM hardware configuration in which the HP 98028A Resource Management Multiplexer and HP 97061 cables are used to connect clients and the SRM server. This configuration is obsolete.

SRM/UX server: An SRM/UX server is an HP Series 300, 400 or 800 HP-UX computer running a special application (*srmd*(1M)) that provides booting, file system, and spooling services to SRM/UX clients. It provides these services such that in most cases SRM/UX clients do not have to know that they are using an SRM/UX server instead of an SRM.

Installing the SRM/UX System Hardware

Site Preparation Checklist

To help make the hardware setup go smoothly, use the following checklist to verify the preparatory tasks:

- If needed, has EMI/RFI shielding been installed?
- Depending on local building codes, coaxial cables should either be run through conduits and ducts, or placed carefully out of the way of foot traffic. Have the T connectors installed, or ready to be installed, at the server and client workstation locations. Both ends of the entire coax link should be terminated with 50 ohm terminators, and one end should be at earth ground.
- Ensure at least one meter of coax cable exists between adjacent workstations and servers on the connection.
- Provide adequate clearances at the front, back, and sides of all workstations to allow for service and operator access, as well as for ventilation.
- It is highly recommended that you check for proper grounding before you make initial connections between the coax cable and the SRM/UX servers and client workstations.

Note SRM coax cable and LAN coax cable have different specifications. In general, LAN coax can be used for both connections, but SRM coax can NOT be used for LAN.

Installing Series 300 and Series 200 Workstations On SRM

Series 300 and Series 200 workstations can communicate with the SRM/UX system through the HP 50962A SRM coax interface or the HP 98629 Resource Management interface. The HP 98629 can use an SRM coax connection configuration after you attach an HP 50961 Resource Management coax Adapter.

Individual client workstations are identified to a server by the workstation's unique node address. This address is set by the DIP switches on the SRM interface card, before the interface is installed in the workstation. This address is emulated in the SRM/UX software for the LAN connection. See chapter 4 of this manual for details on how this is done. Use your hardware configuration worksheets to record information as you configure your SRM/UX servers and clients.

Assigning Node Addresses (SRM Connection)

Once you have established the physical layout of the SRM/UX system, you should assign "node addresses" to every client workstation and server on the SRM connection; follow the procedures in the system Planing Guide in chapter 2 of this manual.

For the SRM connection, select codes and node addresses are physically set via switches on the SRM interface cards that plug into each client and server.

Handling interfaces

Note the following guidelines to avoid equipment damage when handling any interface cards:

- Make sure the workstation's power is OFF before installing or removing interface cards.

Caution Inserting or removing an interface with the power on may damage the interface and the workstation.

- Most interfaces contain components that are sensitive to damage from electrostatic discharge. Therefore, use protective measures, including anti-static workstation and personnel grounding devices. Be especially careful when working in carpeted areas.

Caution Whenever you install, remove, or handle an interface, hold it by its extractors, edges, or cover plate. Do NOT touch its electronic components or circuit board traces.

- If you remove an interface from an anti-static bag to view or to set switches, place the bag on a flat, dry surface and place the interface on top of the bag.

Set the interface Select Code and Node Address

The interface configuration switches are manufactured as clusters of two to eight individual switches combined in a single molded plastic housing. Rocker switches may be flush with the housing, or they may protrude above the housing on one side or the other. The switch value is always determined by which end of the rocker is depressed. If the switch is a slider or a flip switch, the switch position is determined by the position of the tab on the switch.

The following figures show how to interpret switch settings correctly. Each of the four clusters have the identical setting: 01110000.

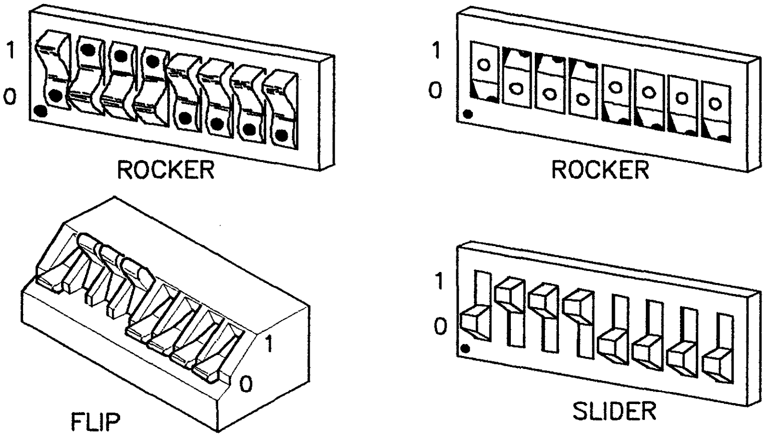


Figure 3-2. Interface card DIP Switch Types

To set rocker switches, use a ball-point pen or other pointed tool to depress each rocker until it is fully seated. To set slide and flip switches, use your fingers or a suitable tool, such as a small screwdriver.

If the rocker or slider is not fully seated, the SRM/UX system may malfunction.

Installing the Interface

1. If you haven't already done so, unpack your Series 300 or Series 200 workstation from its shipping carton. Set up the computer according to its instructions. Situate the computer so that its backplane is accessible.

2. Carefully unpack the HP 50962A SRM coax interface from its shipping carton, leaving it in its protective envelope.
3. Handling the interface carefully by its cover plate and edges only, remove it from the protective envelope, place the envelope on a dry, flat surface, and place the interface on the bag, oriented as shown in the figure below.

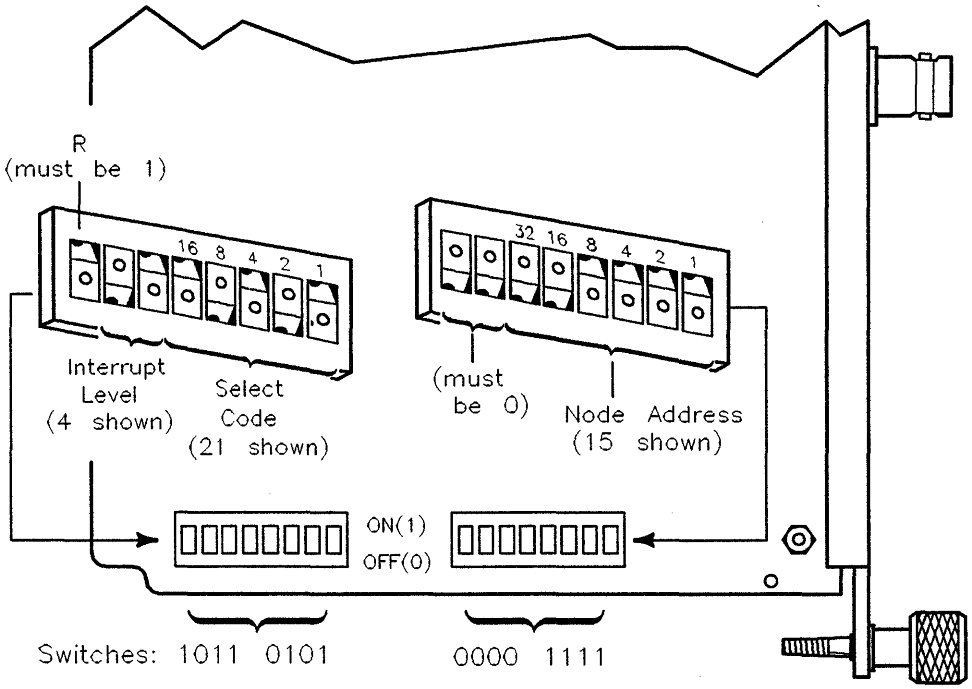


Figure 3-3. SRM Interface Card—Set Select Code and Node Address

4. Locate and set the clusters of switches as follows:

Table 3-2.
HP 50962A SRM coax interface Configuration Switches

Switch Function	Value	Switch Setting
Remote (R)	true	1
Interrupt Level (INT)	4 (for the client's SRM cards) 3 (for the server's SRM cards)	01 (for the client's SRM cards) 00 (for the server's SRM cards)[1]
Select Code (Sel. Code)	21 is recommended, unless you have a LAN interface whose select code has also been set to 21. To avoid a conflict, change the select code for one of the interfaces to a value other than 21.	10101
Node Address (Node Addr.)	unique address in the range 10 through 63	see table

[1] Note that the two switches governing the interrupt level do NOT follow binary numbering conventions.

Note that the leftmost switch in the left cluster *must be set to 1*. Also, the leftmost two switches in the right cluster *must be set to 00*.

The following tables illustrate the correct switch settings for node addresses and select codes.

Table 3-3. Node Address Switch Settings

Decimal Value	2^6 2^5 2^4 2^3 2^2 2^1 2^0	Decimal Value	2^6 2^5 2^4 2^3 2^2 2^1 2^0
10	0 0 1 0 1 0	37	1 0 0 1 0 1
11	0 0 1 0 1 1	38	1 0 0 1 1 0
12	0 0 1 1 0 0	39	1 0 0 1 1 1
13	0 0 1 1 0 1	40	1 0 1 0 0 0
14	0 0 1 1 1 0	41	1 0 1 0 0 1
15	0 0 1 1 1 1	42	1 0 1 0 1 0
16	0 1 0 0 0 0	43	1 0 1 0 1 1
17	0 1 0 0 0 1	44	1 0 1 1 0 0
18	0 1 0 0 1 0	45	1 0 1 1 0 1
19	0 1 0 0 1 1	46	1 0 1 1 1 0
20	0 1 0 1 0 0	47	1 0 1 1 1 1
21	0 1 0 1 0 1	48	1 1 0 0 0 0
22	0 1 0 1 1 0	49	1 1 0 0 0 1
23	0 1 0 1 1 1	50	1 1 0 0 1 0
24	0 1 1 0 0 0	51	1 1 0 0 1 1
25	0 1 1 0 0 1	52	1 1 0 1 0 0
26	0 1 1 0 1 0	53	1 1 0 1 0 1
27	0 1 1 0 1 1	54	1 1 0 1 1 0
28	0 1 1 1 0 0	55	1 1 0 1 1 1
29	0 1 1 1 0 1	56	1 1 1 0 0 0
30	0 1 1 1 1 0	57	1 1 1 0 0 1
31	0 1 1 1 1 1	58	1 1 1 0 1 0
32	1 0 0 0 0 0	59	1 1 1 0 1 1
33	1 0 0 0 0 1	60	1 1 1 1 0 0
34	1 0 0 0 1 0	61	1 1 1 1 0 1
35	1 0 0 0 1 1	62	1 1 1 1 1 0
36	1 0 0 1 0 0	63	1 1 1 1 1 1

Table 3-4. Select Code Switch Settings

Decimal Value	2^4 2^3 2^2 2^1 2^0
21	1 0 1 0 1
22	1 0 1 1 0
23	1 0 1 1 1
24	1 1 0 0 0
25	1 1 0 0 1
26	1 1 0 1 0
27	1 1 0 1 1
28	1 1 1 0 0
29	1 1 1 0 1
30	1 1 1 1 0
31	1 1 1 1 1

Note Use these values in the above tables as guidelines; values other than those in the Select Code table may be used. Also, numbers below 10 can be used for client node addresses (except for 0) if there are less than 10 servers.

Connecting the SRM Coax Cable

This section describes how the coax cable should be connected to form the SRM coax connection. It does not discuss the installation and configuration of the workstations and peripherals. For that information refer to the installation and operating manuals that were shipped with the device.

The RG 58C/U cable connects the servers and client workstations in bus fashion, as shown in the figure below. Place BNC T-connectors at points along the cable, that is, at both ends of each segment, where you plan to connect a server or a client workstation.

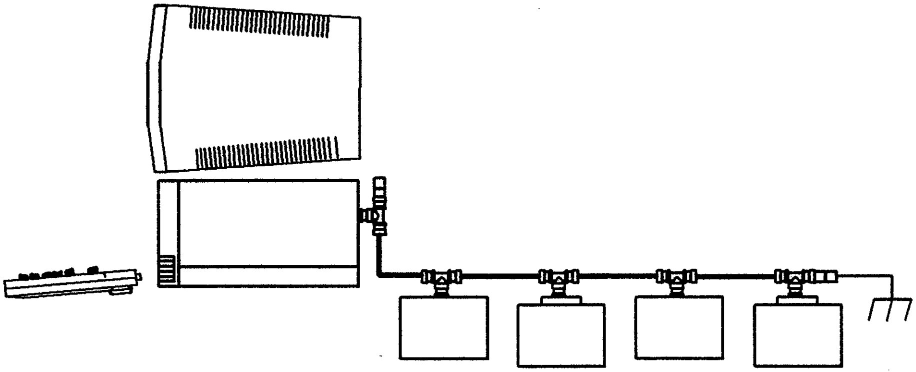


Figure 3-4. SRM Coax Connection Example

The SRM coax connection configuration consists of the following components: coaxial cable, with or without connectors attached (HP offers precut lengths with attached connectors, or uncut lengths to which you attach the connectors), end connectors, barrel connectors, T-connectors, terminators, grounding wire, and plastic insulators.

There are no special power requirements for the SRM coax connection configuration.

Correct and Incorrect Coax Connections

The following figure shows the correct SRM coax connections: the T-connectors are attached to the coax interface or the coax adapter; the cables are attached to the ends of the T-connectors; and both ends of the coax connection are correctly terminated, one at ground and the other with a 50-ohm terminator.

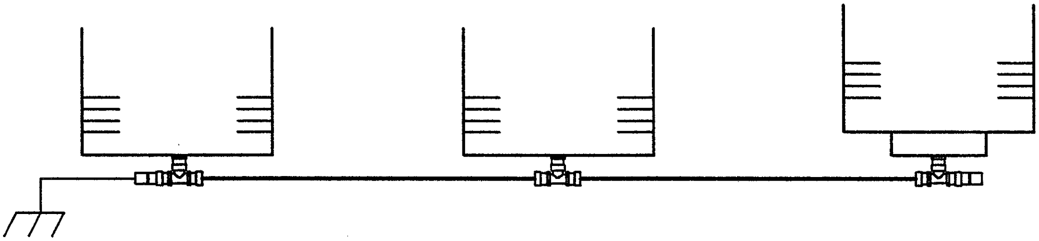


Figure 3-5. Correct SRM coax connection configuration

Caution An unterminated cable can cause permanent damage to HP 50961 or 50962A coax interface cards. Make sure that both ends of your SRM coax connection cable are properly terminated, and that one end is grounded. Do NOT ground both ends of the SRM coax cable.

To properly ground one end of the SRM coax connection, attach a coax terminator with a coated wire one meter long that has a flat, U-shaped end (i.e., a spade lug) to a grounded source, such as an AC wall-outlet or a water pipe. A section of coax cable can be used between the last T connector and a convenient grounding source. The cable should then be fitted with a grounded terminator and the grounding wire attached to the ground source.

The following two figures show incorrect SRM coax connections. The first configuration is incorrect because you cannot connect the cable directly to a coax adapter, and the second because you cannot place a length of cable between the coax interface and the T-connector or between the coax adapter and the T-connector.

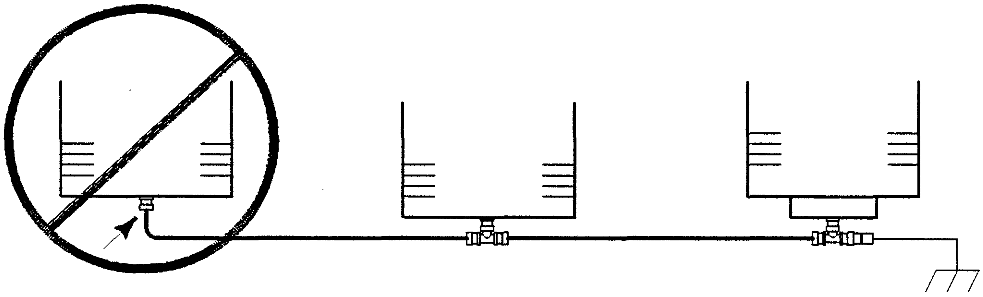


Figure 3-6. Incorrect SRM coax connection configuration—Example 1

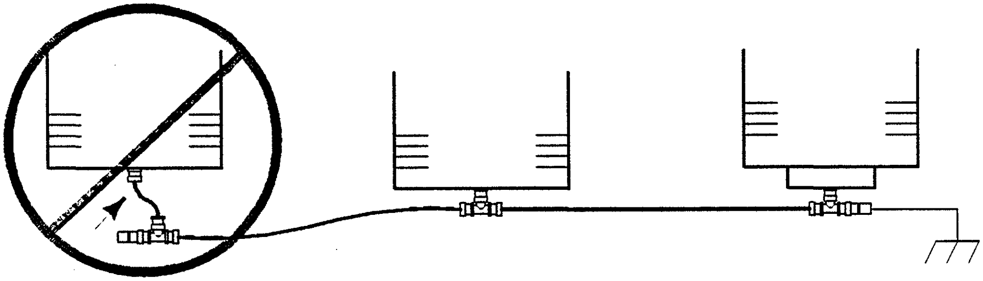


Figure 3-7. Incorrect SRM coax connection configuration—Example 2

To assure product and personnel safety, cover any exposed connectors—T-connectors, barrel connectors, or terminators—on the SRM coax connection. Use the covers that come with the devices.

The following figure shows a close-up view of how the interface card or adapter, T-connector, and coax cable are assembled.

Caution To assure personnel and product safety, an insulating "boot" must be used to cover the T-connector.

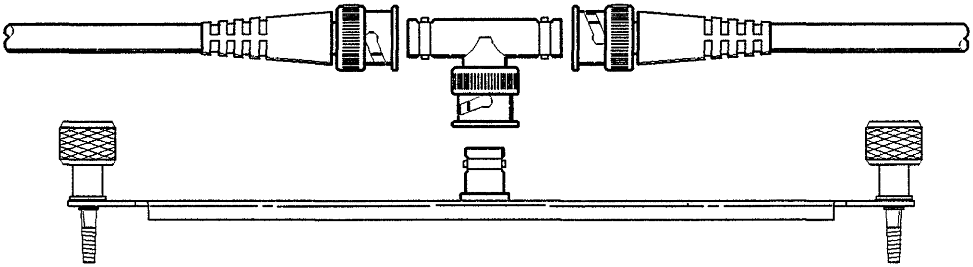


Figure 3-8. Close-up of SRM coax connection at the interface

If it becomes necessary to physically disconnect either a server or client workstation from the SRM coax connection, do so by unplugging the T-connector from the interface card. Leave the cables, or cable and terminator, connected to the T to preserve the SRM coax connection's continuity for other servers and clients.

Assigning Node Addresses (LAN Connection)

Individual client workstations are identified to a server by the workstation's unique node address. This address is emulated in the SRM/UX software for the LAN connection. See chapter 4 of this manual for details on how this is done.

See chapter 2 of this manual for an overview of the SRM/UX system and the details on system planning and assigning node addresses.

Connecting the LAN Coax Cable

See the "Inter-system Communication" chapter of the *HP9000 Workstations Configuration Guide* for details on how to correctly install your LAN interface cards and the LAN cabling.

Installing the Peripherals

The installation of peripheral hardware on the server is done entirely by the HP-UX system administrator and is not in the province of the SRM/UX system administrator.

If you are not only the SRM/UX system administrator but the HP-UX system administrator as well, see the "Series 300, 400, and 800 Interfacing and Accessories" chapters of the *HP 9000 Workstations Configuration Guide* for instructions on how to install the SRM/UX system peripherals.

4

Installing, Configuring, and Running the SRM/UX Server

Overview

This chapter covers the details of the SRM/UX server software. The assumption at this point in the manual is that you have read the first three chapters and, therefore, understand the basics of the SRM/UX system, have planned your site's specific SRM/UX system, have installed the hardware, and have HP-UX version 7.0 or later up and running on the server. This chapter fills in the technical details of the server installation and configuration with respect to SRM/UX file system management and spooling; then, the following chapter covers everything you need to know about clients booting from the SRM/UX server.

Chapters 6 and 7 concentrate on the SRM/UX client issues (HP BASIC/WS and Pascal Workstation System), such as spooling from the client's side and using the SRM/UX file system.

As mentioned earlier, there are a number of tasks that cross the boundaries between SRM/UX system administration and HP-UX system administration. Because there is variety in the way different sites will install computer resources, sometimes both sets of tasks will be handled by one person, and other times there will be several people handling different tasks. It is, therefore, important to closely coordinate among all of the administrators to ensure that all of the necessary tasks get done.

Installing the SRM/UX Server Software

Read the "Updating HP-UX" chapter in the *HP-UX System Administration Tasks* manual to learn about the process of installing new HP-UX applications.

Run *update*(1M) on your server to install SRM/UX from the installation tape. You will find the following partitions on the SRM/UX installation tape:

- SRMUX_COM—SRM/UX man pages and HP BASIC/WS 6.1 installation files
- SRMUX_300—SRM/UX Series 300/400 binaries
- SRMUX_800—SRM/UX Series 800 binaries

Depending on your SRM/UX server's hardware you will want to select either the SRMUX_300 or SRMUX_800 partition, but not both. During the Update process you will be asked the following question:

```
Are the files on this media unit for a Series 300? (y or n)
```

If you are loading the partitions onto a Series 300 or 400 answer: `y`, or if you are loading the partitions onto a Series 800 answer: `n`. Update will then install the SRM/UX files into the correct locations on your server's HP-UX file system.

The SRM/UX installation process places some new HP-UX configuration, command, and device driver files in /etc/newconfig/srmux files on your HP-UX file system. Specifically, the files it places under /etc/newconfig/srmux are: boottab, rbootd, services, srmdconf and srm.o (srm.o on Series 300/400 only). The SRM/UX customization script moves, replaces and modifies, if necessary, the following files:

- /etc/boottab
- /etc/rbootd
- /etc/srmdconf
- /etc/conf/libdil_srm.a
- /etc/services

The SRM/UX customization script saves old files into "save" directories (for example: on an HP-UX revision 7.00 system, /etc/conf/libdil_srm.a is saved as /etc/conf/HPUX7_00/libdil_srm.a). The customization script updates configuration files if they exist or it uses the latest possible version of the file. For example, if /etc/services exists, then the script modifies it. Otherwise it checks /etc/newconfig/services (the current HP-UX version). If /etc/newconfig/services exists, then the script copies it to /etc/services and then modifies it. Finally, if the above two checks fail, the script copies /etc/newconfig/srmux/services (the HP-UX version 7.0) to /etc/services and then modifies it.

Below is an example of what the SRM/UX installation process will put in the file /tmp/update.log for a Series 300/400 SRM/UX server as you run through the installation process:

```
=======  12/18/90  12:02:48 MST  BEGINNING UPDATE PROGRAM (interactive)
         @(#) $Revision: 64.508 $

         Beginning to load fileset "BASIC_WS".
         Successfully loaded fileset "BASIC_WS".
         Beginning customize script for fileset "BASIC_WS" using the command:
         /system/BASIC_WS/customize HP-MC68020
         Customize script for fileset "BASIC_WS" succeeded.
         Beginning to load fileset "SRMUX_300".
         Successfully loaded fileset "SRMUX_300".
         Beginning customize script for fileset "SRMUX_300" using the command:
         /system/SRMUX_300/customize HP-MC68020
A.B7.03.0A
WARNING: If your network is using Yellow Pages (YP), you must add the
         following line to /etc/services on the Yellow Pages server and
         rebuild the Yellow Pages services database:
         "lansrm          570/udp                   # SRM/UX server"
NOTE:    Saved "rbootd" in "/etc/HPUX7_00".
NOTE:    "/etc/rbootd" replaced with "/etc/newconfig/srmux/rbootd".
NOTE:    Saved "libdil_srm.a" in "/etc/conf/HPUX7_00".
NOTE:    A new SRM driver has been installed in libdil_srm.a. In order
         for this new driver to be used by your system the kernel must
         be rebuilt and the system rebooted. Please use SAM to include
         the SRM driver in your kernel.
NOTE:    See the update information in "/etc/newconfig/Update_info/srmux_notes".
         Customize script for fileset "SRMUX_300" succeeded.
         Beginning to load fileset "SRMUX_MAN".
         Successfully loaded fileset "SRMUX_MAN".
         Beginning customize script for fileset "SRMUX_MAN" using the command:
         /system/SRMUX_MAN/customize HP-MC68020
         Customize script for fileset "SRMUX_MAN" succeeded.

=======  12/18/90  12:05:22 MST  COMPLETED UPDATE PROGRAM (interactive)
```

Configuring the Server's HP-UX Environment For SRM/UX

The SRM/UX server is implemented as a daemon process running in the background of a multitasking HP-UX host workstation. For SRM/UX to function correctly you must customize the server's HP-UX environment. This customization involves some combination of the following tasks—depending on your specific system needs (Depending on your specific HP-UX configuration, many of these tasks may already be complete.):

Note There are tasks that require the server to be shut down and tasks that can be done while the server continues to run.

Tasks to be done with the server shut down:

- plugging in interfaces

Tasks to be done with the server up and running:

- mounting file systems
- reconfiguring the HP-UX kernel
- converting all or part of the HP-UX file system to handle long file names
- installing the capability to mount remote file systems (if desired)
- establishing a working HP-UX spooling environment

Reconfiguring the HP-UX Kernel

For the Initial SRM/UX System

To create the needed HP-UX kernel configuration on your server, you will use the procedures detailed in the "Reconfiguring the Kernel" chapter of the *HP-UX System Administrator Tasks* manual.

First, establish what your HP-UX kernel needs are based on the SRM/UX system you want to configure. That is:

- Do you have a LAN connection?
  - □ If you do, then you need the LAN driver in the kernel.

- Do you have an SRM connection (Series 300 only)?
  - □ If you do, then you need the SRM driver in the kernel.
- Do you need to mount remote file systems?
  - □ If you do, then you need the nfs driver in the kernel.
- What printers and plotters will you be using?
  - □ You need the correct drivers in the kernel for each spooling device.

Next, check your server's current HP-UX kernel configuration with the HP-UX system administration program: SAM. If your HP-UX kernel is missing any of the needed drivers, then you must reconfigure the kernel.

You may reconfigure the HP-UX kernel using SAM, the HP-UX system administration program. Use the procedures from the "Reconfiguring the Kernel" chapter in the *HP-UX System Administration Tasks* manual.

For an Existing SRM/UX System

If you have reconfigured the HP-UX kernel to fit your original SRM/UX system needs and later you need to change the SRM/UX system configuration, you may have to reconfigure the HP-UX kernel again. Refer to the *HP-UX System Administrator Tasks* manual for details on how to reconfigure the kernel.

- If you want to add a LAN connection to the SRM/UX system, then verify that the kernel contains the LAN driver. If it doesn't, reconfigure the kernel.
- If you want to add an SRM connection to the SRM/UX system, check to see if you have the correct SRM driver in your kernel by executing the following command:

  ```
  what /hp-ux
  ```

  You should search for the following line in the output of the `what` command:

  ```
  NEW SRM DRIVER 12/14/90
  ```

  You have the correct version if the date is 12/14/90 or later.

  The correct version of the new SRM driver (that replaces the standard 7.0X version) was installed by the SRM/UX installation process in /etc/conf/libdil_srm.a (for Series 300 and 400 only).

If you do not have the correct version, then reconfigure the kernel.

- If you want to add printers or plotters that don't have drivers in the standard kernel, first check `/etc/master` to see if HP supports the device. If the device appears in `/etc/master`, then add the appropriate driver to the kernel.
- If you need NFS Mounts, then add the nfs driver to the kernel.

HP-UX System Considerations

The srm Device File

If you are running on the SRM connection, then you must create a device file, with a unique name within its server, for each SRM interface card in each server. We recommend using the name `/dev/srm` if you have only one SRM interface card, and using `/dev/srm1`, `/dev/srm2`, etc. if you have multiple SRM cards. It is standard HP-UX practice to put all device files in the `/dev` directory.

As superuser (root), set up an `srm` device file by executing the command:

```
/etc/mknod /dev/srm c 13 0xnn0000
```

where nn(hex) is the SRM card's select code

See the "Device Files" chapter in the *HP-UX System Administration Concepts* manual for a complete discussion of device files, hardware addresses, and major/minor numbers.

Note While the server process *srmd*(1M) is running, any SRM interface cards in the server that are referenced by their device files (in /etc/srmdconf), will be "locked". While they are locked, the SRM Access Utilities, HP BASIC/UX, and other HP-UX processes will not be able to access the card, through the device file in /etc/srmdconf or any other device file.

Note If you are also using the SRM Access Utilities, then there is a different convention for the minor number. See the *srm*(1) man page that comes with the SRM Access Utilities for details. It is also possible to use multiple device files with the SRM Access Utilities.

Single LAN Interface Support

The SRM/UX installation process adds the following line to /etc/services to allow the SRM/UX server to support LAN interfaces (if it does not already exist):

```
    lansrm          570/udp                 # SRM/UX server
```

Note To set up an SRM/UX server on an HP-UX system that is, running Yellow Pages, you must take the additional steps listed below:

- add the line: `lansrm 570/udp` to /etc/services on BOTH the SRM/UX server computer and the Yellow Pages server
- on the Yellow Pages server type:

  ```
  /etc/yp/ypmake services
  ```

This will add the changes to the services file into the Yellow Pages data base and *srmd*(1M) will work properly.

Multiple LAN Interface Support

If you need to configure the SRM/UX server to handle multiple LAN cards, do the following:

- Perform the procedure for installing a single LAN interface detailed above
- Install the appropriate "ifconfig" invocation for each LAN card in the /etc/netlinkrc file. The names of the LAN cards specified in the ifconfig commands MUST include either of the strings "lan" or "ieee" (lower case) to be recognized by SRM/UX.

  Valid name examples: lan0 lan1 ieee2

Refer to the "Managing an HP-UX Cluster" chapter in the *HP-UX System Administration Tasks* manual for details.

- Set up *rbootd*(1M) to listen to multiple LAN cards by changing the shell variable, RBOOTD_DEVICES, in the initialize() function in /etc/rc to accommodate all of the LAN cards.

  For example, with two LAN cards, the assignment statement might look like this:

  ```
  RBOOTD_DEVICES = "/dev/lan0 /dev/lan1"
  ```

File Name Length

SRM supports file names up to 16 characters long. The default HFS file system under HP-UX has 14 character file names. Therefore, to ease the migration of applications and data from SRM to SRM/UX, HP recommends converting all of the HP-UX discs that contain files that are accessed by SRM/UX clients to a long-file-name system by using *convertfs*(1M). See the *HP-UX Reference, Vol 3* for details on *convertfs*(1M). Also, read the discussion on converting to long file names in the "Managing the File System" chapter of the *HP-UX System Administration Tasks* manual.

The SRM/UX server hides file names longer than 16 characters from the clients.

Note HFS disks configured for long file names cannot be connected directly to Pascal Workstation System or HP BASIC/WS systems because their HFS drivers support only short-file-name HFS disks.

Remote File System Access

Through the Network file system (NFS), an industry standard networking tool available through HP-UX, the SRM/UX server can mount remote file systems. Refer to the "Networking" chapter in the *HP-UX System Administration Concepts* manual for details on NFS.

This capability is provided entirely by the HP-UX operating system on the SRM/UX server, and is completely transparent to the HP BASIC/WS and Pascal Workstation System clients, and to SRM/UX itself. It provides access to both HP-UX and non-HP-UX machines, including machines which don't themselves support SRM/UX. It is independent of which interface card, LAN or SRM, the client uses to access SRM/UX.

This added capability means that parts of or all of the **SRM/UX file system** may be located on discs that are physically remote from the SRM/UX server's disc(s). It allows more flexibility in configuring an SRM/UX system, but has the added responsibility for the SRM/UX System Administrator to ensure that each disc which contains SRM/UX files is converted to long file names and to ensure (via the HP-UX System Administrator) that all of the remote discs are included in backups.

To provide remote file system access for SRM/UX, the root user must first make sure that NFS is installed on the SRM/UX server computer. After NFS is installed, the root user can, at any time, mount remote file systems for which the server has permission. For example, in order to mount the directory /users on the remote system hpxyzzy, and access it via the local path /srmuxroot/usersxyzzy, make sure /srmuxroot/usersxyzzy exists on the SRM/UX server (using *mkdir(1)* to create directories), then execute:

```
/etc/mount hpxyzzy:/users /srmuxroot/usersxyzzy
```

Now, SRM/UX client workstations can access /users on hpxyzzy via path /usersxyzzy on the volume associated with /srmuxroot in the VOLUME-TABLE in /etc/srmdconf. The directories' and files' uids and gids should match on both the SRM/UX server and the remote NFS-mounted computer, in order to minimize confusion over permissions.

Spooling

The SRM/UX System Administrator must investigate to see what spooling devices are supported on the server. He or she must then set up the SRM/UX spooling environment to support those devices. See your site's HP-UX System Administrator and/or the *HP-UX System Administration Tasks* manual for details on setting up the server's HP-UX spooling environment.

The SRM/UX server supports printer and plotter spooling by using the underlying HP-UX spooling functionality. In addition, a front-end to the HP-UX `lp(1)` command (called *srmdlpfilter*(1)) pre-processes HP BASIC/WS and Pascal Workstation System file types. The pre-processing includes stripping off WS headers (which are hidden binary headers in HP BASIC/WS and Pascal Workstation System files). The operation of this front-end is transparent to the clients. Therefore, Pascal Workstation System and HP BASIC/WS users are able to do spooling to SRM/UX exactly as they do for the SRM.

Establishing the SRM/UX File System

The SRM/UX file system is defined from two different perspectives. From the client's perspective the SRM/UX file system is a set of emulated SRM volumes. From the server's perspective the SRM/UX file system is one subtree or a set of subtrees of the HP-UX file system that is presented to the SRM/UX clients. The SRM/UX file system is established by the SRM/UX System Administrator under the HP-UX root directory and is configured into the mass storage of the HP-UX server (host) computer.

Through the Network File System (NFS), an industry standard networking tool available through HP-UX, the SRM/UX server can mount remote file systems. This added capability means that parts of or all of the SRM/UX file system may be located on mass storage that is physically remote from the SRM/UX server.

If you want to transfer your SRM file system directly to SRM/UX with minimum effort, because of naming conventions in the Boot ROM and in HP BASIC/WS and Pascal Workstation System, HP recommends placing all of the directories and files that are related to booting SRM/UX clients (such as the SYSTEMS, WORKSTATIONS, and RMBBINS directories) under one directory, and calling that directory the **SRM/UX root directory** (`srmuxroot`).

You can place the SRM/UX files that are not related to client booting at any location under the HP-UX root, if you set up the appropriate volumes in /etc/srmdconf.

As the SRM/UX System Administrator, you can allow SRM/UX clients access to only the SRM/UX root directory at first. This is the most straight-forward way to serve the clients. As your client's needs grow and you gain more experience with SRM/UX, you may customize the SRM/UX file system to more closely match your SRM/UX system requirements.

You can create the SRM/UX root directory anywhere under the HP-UX root directory that makes sense for your particular site. That is, you might wish to place the SRM/UX root directory at /srmuxroot or /srmroot, or it may fit your requirements better to place it at another location, such as /usr/local/srmuxroot.

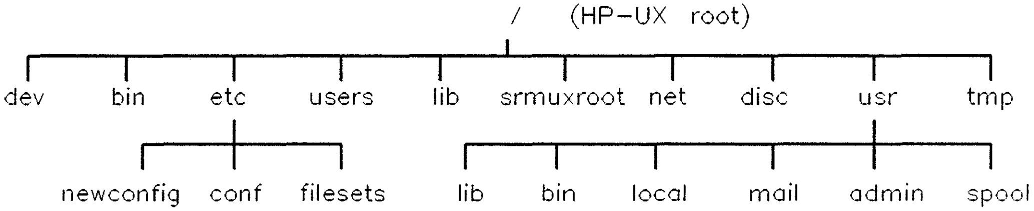


Figure 4-1. Example HP-UX File System

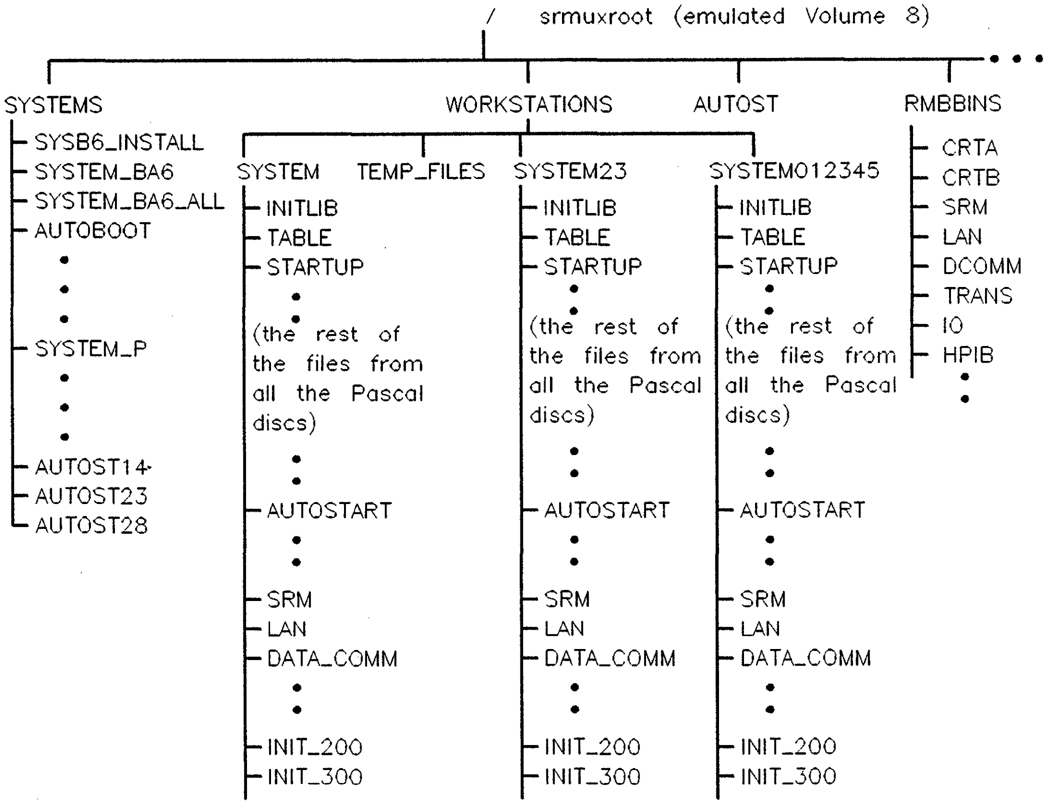


Figure 4-2. Example SRM/UX File System

The server process, *srmd*(1M), manages the SRM/UX file system based on the information you give it in the configuration file, srmdconf. That is, the SRM/UX System Administrator controls, through the entries in srmdconf, which HP BASIC/WS and/or Pascal Workstation System clients can access which directories (i.e., emulated SRM volumes) on the HP-UX file system.

Establishing the SRM/UX Spooling Environment

The SRM/UX spooling environment uses the existing HP-UX spooling environment to emulate SRM spooling.

The SRM/UX server supports printer and plotter spooling. To properly configure your SRM/UX spooling environment, first you must configure your HP-UX spooling environment, and then configure the SRM/UX spooling environment to pass files to the HP-UX spooling environment. To learn how to configure your HP-UX spooling environment, refer to the *HP-UX System Administration Tasks* manual.

If something goes wrong with the spooling environment, error messages, warnings, and other general information are sent to the /usr/lib/srmux/srmdspool.log file.

Both Pascal Workstation System and HP BASIC/WS client users do spooling to SRM/UX exactly as they do for SRM.

Sending Files From a Client Workstation to a Printer or Plotter

Printers and plotters are accessed through the SRM/UX system's spooler for SRM/UX clients and can also be accessed through HP-UX for the server. Every printer or plotter must have its own spooler directory, which the SRM/UX System Administrator configures into the /etc/srmdconf file by adding a line to the SPOOL-TABLE. A spooler directory stores files from clients in order and sends the files, one-by-one, to the associated printer or plotter. To the extent that a client workstation must treat printers and plotters as directories for spooling files, access to these shared peripherals is transparent.

To use a spooler, a client user places a copy of the file to be printed or plotted into a spooler directory (for priority printing, the file is placed into a special "PRIORITY" subdirectory in the spooler directory). Any SRM/UX data file may be spooled. (Non-HP-UX files are handled by *srmdlpfilter*(1); details on *srmdlpfilter*(1) appear later in this chapter.)

A spool file is automatically sent to the printer or plotter associated with the spooler directory when the file meets the following criteria:

- The file is not empty.
- The file is closed.

- The file is either:
  - □ the head of the queue of files in the PRIORITY subdirectory of the spooler directory.
  - □ the head of the queue in the spooler directory and the PRIORITY queue is empty.

Printer Environments and Priority Spooling

SRM/UX provides both ENVIRONMENT and PRIORITY spooling. These features and their use are described in the next sections.

Printer Environment Spooling

Some printers allow you to send commands and data that set up a desired printing environment, e.g., to load a certain font or to select a printing orientation. An SRM/UX system spooler allows files that contain printer environment information (environment files) to be held in an ENVIRONMENTS subdirectory of the spooler directory. When a print file is sent to a printer, the spooler prefixes an environment file (that you specify) to the print file, or it prefixes the default environment file, if one exists. The environment file's printer information defines the printer environment in which the print file is then printed.

Files spooled without ENVIRONMENT files are handled this way:

- ASCII files are sent with an automatic carriage return and line feed after each record.
- The byte stream contained in files of any non-ASCII type are NOT printed or plotted exactly as they were in the file, *srmdlpfilter*(1) does a translation on the files.

To set up environment spooling, follow these steps:

1. Use a client workstation on the SRM/UX system to create a subdirectory called ENVIRONMENTS (upper case required) in the desired printer spooler directory, or you can create the ENVIRONMENTS directory from the server with an HP-UX *mkdir*(1) command.
2. Determine from the particular printer's manual what environment information is required.

3. Place the appropriate information in a file in the ENVIRONMENTS subdirectory. Environment files:
   a. Must be of type ASCII, BDAT, or Pascal Data. Do not use Pascal TEXT. Files of type ASCII may end in .ASC or .asc.
   b. Must have valid file names.
   c. Must have unique filenames. The spooling environment does not distinguish environment filenames with different extensions.
4. Repeat steps 2 and 3 to create a file for each printer or plotter on your system.
5. Optionally, place default environment information in a file in the ENVIRONMENTS subdirectory. Name this file DEFAULT, DEFAULT.ASC, or DEFAULT.asc. This environment is used when no other environment file is specified.

Here is an example spool directory:

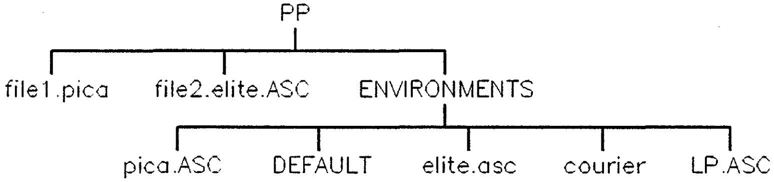


Figure 4-3. Example Spool Directory

To use environment spooling:

Specify the name of the desired environment file as part of the file name of the spooled file. A spool file name then takes the form:

spool_filename.environment_filename[.ASC]

where a period is used to separate *spool_filename* from *environment_filename*, and the suffix [.ASC] or [.asc] applies to the spool file itself and not to the environment file.

In the example spool directory above, for example: `file2.ASC` is to be printed in the printer environment specified by the environment file `elite.asc`. To do this, a file with the following filename is placed in the spool directory:

```
file2.elite.ASC
```

where the .ASC extension is part of the original filename, `file2.ASC`.

When an environment file is not specified in the spool file name, the spooler prefixes the DEFAULT environment file to the spool file. In the absence of a DEFAULT environment file, the spool file is printed without an environment.

Note If you create an ENVIRONMENTS subdirectory in a spooler directory, you should thereafter include the name of an environment file as part of every file you send to that spooler, unless you want the default environment.

An Example Environment File

The data below could be placed in an ASCII environment file called "LP" or "LP.ASC". In this example, the unprintable ASCII character 27 (escape) is replaced by `^[`. You would have to substitute ASCII 27 for every occurrence of `^[` in the example below for it to work.

```
[&l1O^[(8U^[(s0P^[(s16.66H^[(s8.5V^[(s0S^[(s-3B^[(s0T^[&l5C
```

Do not confuse `0` (zero) with `O` (upper case "o") or `l` (lower case "L") with `1` (the number one).

This environment works with the HP LaserJet Series II printer and sets up the printer to print in landscape orientation; "Line Printer" is the primary typeface. This environment is good for printing program listings.

To use this environment file, place `file_name.LP.ASC` in the spooling directory and the environment file "LP" described above will be prefixed to *file_name* for spooling.

Priority Spooling

The SRM/UX system's spooler allows you to print "rush" jobs before others. Print or plot files sent to the PRIORITY subdirectory in a spooler directory will print or plot before other files in the spooler directory.

To set up priority spooling, use a client workstation on the SRM connection to create a subdirectory called PRIORITY in the desired spooler directory, or you can create the PRIORITY directory from the server using the HP-UX *mkdir*(1) command.

To use priority spooling, place the file to be printed or plotted with higher priority into the spooler directory's PRIORITY subdirectory. Any file in the PRIORITY subdirectory is printed before the files placed directly in the spooler directory if the Spool Command in /etc/srmdconf for this printer uses the $PRIORITY shell variable in its lp invocation.

Priority spooling is compatible with environment spooling, as described in the subsection above. Below is an example spooler directory set up for both priority spooling and environment spooling:

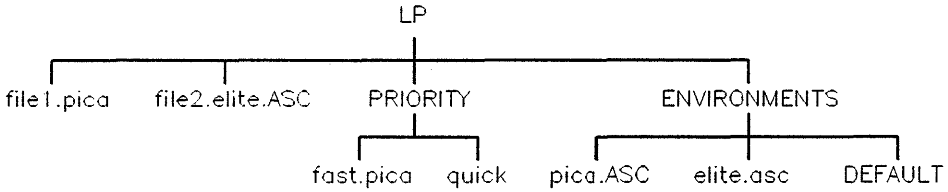


Figure 4-4. Example Spool Directory—Environment and Priority Spooling

Note You should limit the use of the PRIORITY subdirectory. If all spool files are placed in the PRIORITY subdirectory, the priority advantage is lost. You may want to protect the PRIORITY directory by only allowing special users to access it. To do this, the SRM/UX System Administrator should modify the HP-UX permissions on the PRIORITY subdirectory.

Configuring the SRM/UX Server Process

The file, /etc/srmdconf, is used to configure *srmd*(1M) and defines how it emulates an SRM host for each Pascal Workstation System and HP BASIC/WS client.

Changing the /etc/srmdconf file is the province of the SRM/UX system administrator; client users are not allowed to change any of the items in /etc/srmdconf.

The SRM/UX installation process will place a "template" srmdconf in your server's /etc directory, with its permissions already set correctly. To maintain the integrity of the srmdconf file, the root user should own it. If for some reason the file's permissions have been changed, set its permissions to:

```
rw-r--r--     # i.e., mode 644
```

so that other users can examine it but not modify it.

The SRM/UX Configuration File—srmdconf

The functions of srmdconf are:

- Specifying file access control

  The Administrator controls which HP BASIC/WS and Pascal Workstation System client workstations can access which pieces of the HP-UX file system on the SRM/UX server computer. The set of allowed accesses may be different for each client.

  The Administrator also controls file permissions via the Uids, Gids, and Umasks specified in /etc/srmdconf.

- Declaring the emulation of SRM node addresses for LAN clients

  LAN and SRM interfaces are managed in different ways. For the LAN card, node addresses do not exist, therefore, *srmd*(1M) emulates an SRM node address for each LAN card it communicates with. Two node addresses need to be emulated: the SRM host (server) node address and the client node address.

  Such emulation is not necessary for SRM interfaces, because the required node addresses are set using "DIP" switches on the SRM cards.

  As a result of the differences between SRM and LAN interfaces, the format of the LAN-CLIENTS and SRM-CLIENTS tables of /etc/srmdconf are different.

- Specifying spooling services

  The SRM/UX server provides printer and plotter spooling services to the HP BASIC/WS and Pascal Workstation System clients through entries in /etc/srmdconf. The *srmd*(1M) process controls SRM/UX spooling operations.

The SRM/UX installation process installs a template srmdconf file in the /etc directory on your server. This template appears below:

```
VOLUME-TABLE

# Volume Descriptions
#
# Volume         |Volume|     |     |Temp     |Root
# Name           |Number| Uid | Gid |Directory|Directory
#----------------+------+-----+-----+---------+-------------------------------------
 SRMUXROOT       :   8  :     :  9  : tmp      :/srmuxroot

LAN-CLIENTS

# LAN Client Descriptions
#
# Client       |Client         |Client|Server|           |     |     |     |
# Link Level   |Internet       |Node  |Node  |Client     |     |     |     |Volume
# Address      |Address        |Addr  |Addr  |Name       | Uid | Gid |Umask|List
#--------------+---------------+------+------+-----------+-----+-----+-----+--------
#0x080009001234:   15.99.99.100:  11  :   0  :examplelan:   102:    9: 002 :SRMUXROOT # Scott's hp9000s332

SRM-CLIENTS

# SRM Client Descriptions
#
#                |Client|          |     |     |     |
# Server's SRM   |Node  |Client    |     |     |     |Volume
# Device File    |Addr  |Name      | Uid | Gid |Umask|List
#----------------+------+----------+-----+-----+-----+------------------------------
#/dev/srm        :  10  :EXAMPLESRM:  101:    9: 002 :SRMUXROOT

SPOOL-ENVIRONMENT

# The following environment variables are pre-initialized for spooling
# processes.
#
#       SHELL                   - defaults to /bin/sh, unless overridden
#       TZ                      - inherited from the OS, if it is set there,
#                                 otherwise defaults to MST7MDT
#       SRMSPOOLFILENAME        - defaults to the full pathname of the file
#                                 being spooled (e.g. /srmuxroot/PRINTER/xyzzy)
#       PATH                    - defaults to /bin:/usr/bin. Recommend that
#                                 it be set to include paths needed for all
#                                 commands used in Spool Commands.
#--------------------------------------------------------------------------------
---
PATH=/bin:/usr/bin:             # srmdlpfilter is in /usr/bin

SPOOL-TABLE

# Spooling Commands
#
# Spool               |        |Spool
# Directory           |Priority|Command
#----------------------+--------+---------------------------------------------------
 /srmuxroot/PRINTER    :     2  :srmdlpfilter | lp $PRIORITY
```

Format of srmdconf—Keyword Tables

There are 5 keywords that introduce the various tables of srmdconf. They are: VOLUME-TABLE, LAN-CLIENTS, SRM-CLIENTS, SPOOL-ENVIRONMENT, and SPOOL-TABLE. They must appear in the order given, and in upper-case, exactly as shown in the template srmdconf file that appears above. (The SRM-CLIENTS table appears in Series 800 and Series 400 server's /etc/srmdconf files, but must be empty because Series 400's and Series 800's do not support SRM Cards.) Each table in srmdconf describes a different part of the configuration.

For all tables, fields within each line of the table are separated by ":" characters. Some fields are optional, which means that you can omit the value for the parameter, but you must include both colons on either side of the field. You can put any amount of "white-space" between the colons.

Tables and the fields within the tables must appear in the order seen in the example (the same order in which they are discussed below). A comment can appear as the last item in any line of the file, and is introduced by the "#" character; anything after a "#" is ignored by *srmd*(1M), and *rbootd*(1M), until the start of the next line. Blank lines or comment lines may be inserted anywhere in the file to improve readability. The non-comment part of a line is limited to 255 characters.

The VOLUME-TABLE

The VOLUME-TABLE contains names and descriptions of "volumes" (emulated SRM volumes) that the server "presents" (i.e., allows access to) to Pascal Workstation System and HP BASIC/WS clients. Each SRM/UX volume is a subtree of the currently mounted tree of the HP-UX host (including any accessible remote-file-systems previously mounted by NFS utilities) and is described by one line of text in the VOLUME-TABLE.

As the SRM/UX System Administrator you must decide which client workstations will have access to various pieces of the SRM/UX file system. You should set up the SRM/UX file system and place programs and files in locations that follow your system plan (as developed in chapter 2). Then you can control which users access various pieces of the SRM/UX file system through the /etc/srmdconf file's VOLUME-TABLE, LAN-CLIENTS table, and SRM-CLIENTS table.

If the client users have an HP-UX account on the server and, therefore, have a login or "home" directory, you can give those users access to that directory from the SRM/UX client workstations. You do this by making entries in both the VOLUME-TABLE and the LAN-CLIENTS or SRM-CLIENTS table. You would do this if the client workstation had one user most of the time. If a client is used by many people, you probably would not want to do this.

Create a volume name in the VOLUME-TABLE that represents the user and assign it to the user's HP-UX home directory in the `Root Directory` field. For example, if Joe Smith has an HP-UX account and his login directory is `/users/joes`, then you could create a `Volume Name` of `JOES` with a `Root Directory` of `/users/joes`. Then in the LAN-CLIENTS table (assuming the client is on the LAN connection) you would include `JOES` in the `Volume List`.

In this indirect way, then, Joe Smith can access his HP-UX directories from an HP BASIC/WS or Pascal Workstation System SRM/UX client workstation.

Below you will find the detailed explanation of the fields in the VOLUME-TABLE and valid entries in them. Directly after the explanations you will find a summary table which provides quick access to VOLUME-TABLE configuration information.

The `Volume Name` field is the name of the volume as seen by the client. It simulates the Volume Label of an SRM disc. An HP BASIC/WS user uses this name as the LABEL specifier in the MSI statement, and can see it as part of the header on CAT output. A Pascal Workstation System user can see this name by using the Filer List command to list the directory "/" for the unit to which the volume is associated. Each line of the VOLUME-TABLE *MUST* have a unique volume name.

The `Volume Number` field is the emulated disk volume number as it would be seen on an SRM. Volume 8 is special: booting from SRM/UX is supported only from this volume. Only one volume 8 is allowed in the VOLUME-TABLE. There may be multiple volumes with other volume numbers, but a given client may access only one volume with a given number. See the *Installing and Configuring Pascal Workstation Clients* chapter in this manual for details of how this field is used in CTABLE. HP BASIC/WS does not use this field (except that the Boot ROM uses Volume 8); it only uses volume labels for SRM disks.

The Uid field is a User or Owner ID value that will be assigned to files and directories created through the *srmd*(1M) server. The Uid and Gid fields are used for all file permissions checking. This Uid is an override of the value specified in the LAN-CLIENTS and SRM-CLIENTS tables of srmdconf. If the uid field in the VOLUME-TABLE is empty, the value in the LAN-CLIENTS or SRM-CLIENTS tables will control.

Note Using a Uid of 0 (zero) is NOT recommended. The Uid of 0 is used for the HP-UX super-user. In general, you should not give SRM/UX Clients access to HP-UX super-user privileges on the SRM/UX Server.

The Gid field is similar to the Uid field, providing group ID assignment for newly created files and directories and file permissions checking.

The Temp Directory field provides a mapping for the file name /WORKSTATIONS/TEMP_FILES for Pascal clients. (The Temp Directory field is NOT relevant for HP BASIC/WS clients.) For many operations, the Pascal Workstation System SRM driver creates temporary files which it stores on the SRM/UX file system in the directory /WORKSTATIONS/TEMP_FILES. It is often not convenient to provide the directory /WORKSTATIONS/TEMP_FILES at the root of each SRM/UX subtree. To provide flexibility, this field tells *srmd*(1M) to convert the directory /WORKSTATIONS/TEMP_FILES to one which is more convenient. If the path you specify here starts with a leading "/", it is an absolute path name in the HP-UX file system (for example: `/tmp`). Otherwise, the path is evaluated relative to the Root Directory field for its volume (see next paragraph). If the field is empty (or is all spaces), no conversion is done.

Note Many of the Pascal Workstation System tools expect to create files in /WORKSTATIONS/TEMP_FILES and then link them to their final name in the current directory using hard links. If there are HP-UX mount points inside of a logical SRM/UX volume, this may cause problems because hard links are not allowed across physical disks. (The SRM can also encounter these problems if the SRM's MOUNT command is used to mount another disk within an SRM logical volume.)

See the "Pascal-Specific File Serving Problems" in Appendix B in this manual for a description of the work-around for this problem.

The `Root Directory` field provides the path on the HP-UX file system to use as the SRM/UX root directory for this volume. For example, the SRM/UX installation process will create a volume 8, called SRMUXROOT, which will use the HP-UX directory /srmuxroot as its `Root Directory`. This emulated SRM volume contains all of the directories and files that are related to booting SRM/UX clients (such as the SYSTEMS, WORKSTATIONS, and RMBBINS directories). When a client specifies the root of this SRM/UX volume ("/"), the SRM/UX server process will translate the client's path into the HP-UX path /srmuxroot. (This translation is completely invisible to the clients.) Attempting to go above the SRM/UX `Root Directory` from the clients (e.g., by issuing an `MSI ".."` command in BASIC) will leave the user at the SRM/UX `Root Directory`; that is, going above the SRM/UX `Root Directory` from the client is not allowed.

The following table summarizes the configuration rules for the VOLUME-TABLE:

Table 4-1. VOLUME-TABLE Configuration Guidelines

The string VOLUME-TABLE must appear exactly as shown here	
Six fields are required in Each VOLUME-TABLE Entry	
Field	**Rules for Valid Entries**
Volume Name	1) must be 16 characters or less 2) must be non-blank 3) must be unique
Volume Number	1) must be between 1 and 127, inclusive 2) it is not necessary to have a unique Volume Number for each Volume Name (but a given client cannot access multiple volumes with the same Volume Number)
Uid	1) entry not required 2) must be between 0 and 65535, inclusive (0 NOT recommended)
Gid	1) entry not required 2) must be between 0 and 65535, inclusive
Temp Directory	1) entry not required 2) the directory specified must exist on the system 3) if the entry has a leading slash, the path is absolute otherwise the path is relative to the Root Directory of the volume
Root Directory	1) entry must be non-blank 2) must be entered as an absolute path 3) the directory specified must exist on the system

The LAN-CLIENTS Table

Below you will find the detailed explanation of the fields in the LAN-CLIENTS table and valid entries in them. Directly after the explanations you will find a summary table which provides quick access to LAN-CLIENTS table configuration information.

The `Client Link Level Address` field specifies the LAN Link Level Address of the LAN Interface on the Pascal Workstation System or HP BASIC/WS client to which service is to be provided. It is a 12 digit hex number (6-bytes) which is factory-set by HP to be unique. Many Series 300 computer BOOTROMs report this number; if it is reported, it will appear on the left side of the screen, on the line where the existence of the LAN Interface is reported. You should enter the attended boot mode to keep the Link Level Address on the screen long enough to record it. If the BOOTROM does not report the Link Level Address, either use the HP BASIC/WS Status Registers 21 through 26 on the LAN Interface, or see the installation manual that came with the LAN Interface (or the computer's installation manual if the LAN Interface is built in) for information on determining the Link Level Address.

The `Client Internet Address` field specifies the Internet Address (IP Address) of the client. Each LAN client must have a unique Internet Address AND the client's Internet Address must be on the same subnet as the SRM/UX server (meaning that the network address part of the server and client IP addresses are the same). For example, if the SRM/UX server is at IP Address 15.99.99.1 (with a network address of 15.99.99, and a host address of 1), then SRM/UX clients could use any IP Addresses from 15.99.99.2 through 15.99.99.254 that were not already in use by other machines. (Host addresses of 0 and 255 are reserved addresses.)

If there are also HP-UX computers (i.e., workstations running HP-UX that are unrelated to SRM/UX) on the same subnet, care is required to avoid duplication of Internet Addresses. You should make entries for your SRM/UX clients in your /etc/hosts file and, if one exists on your system, your /etc/hosts.local file, so that HP-UX System Administrators for other machines on your network will know which IP Addresses are assigned to your SRM/UX Clients. (If multiple computers attempt to operate with duplicate Internet Addresses, your SRM/UX System and/or your entire HP-UX network, may malfunction.)

Note If a given LAN client will be connected to multiple SRM/UX servers from one LAN Interface, it MUST have the same Client Internet Address on all of the SRM/UX servers.

Note If a given workstation runs HP-UX at some times, and is an SRM/UX Client (running HP BASIC/WS or Pascal Workstation System) at other times, it should be assigned the same Internet Address for both cases (there should be just one entry in /etc/hosts).

Note For a more detailed discussion of assigning Internet Addresses, see the "Managing an HP-UX Cluster" chapter of the *HP-UX System Administration Tasks* manual or the *Installing and Administering LAN/9000 Series 300* or *Installing and Administering LAN/9000 Series 800* manuals.

The `Client Node Addr` field specifies the emulated SRM client node address that the SRM/UX server will supply to the SRM/UX client. On SRM cards, the Node Address is set with 6 switches. SRM clients use their Node Address to generate node-specific file names. For example, HP BASIC/WS uses this number to support node-specific AUTOST files (such as /SYSTEMS/AUTOST23). User applications can also use this number to generate node-specific file names. (HP BASIC/WS returns this number in response to a "STATUS *sc*, 6" statement, where *sc* is the select code of the client's SRM or LAN interface.) For example, multiple SRM clients, with Node Addresses 34 and 41, which are running the identical program, could write their output to different files, such as RESULTS34 and RESULTS41. The SRM/UX server supplies LAN clients with an emulated SRM Node Address so that programs which used the Node Address on the SRM can easily be ported to SRM/UX.

Client Node Addresses on LAN must be between 1 and 126, inclusive. However, numbers less than 10 are (by convention) reserved for servers, not clients, and numbers higher than 99 may cause problems for client applications which assume that Node Addresses use only 1 or 2 digits (which is the case on the SRM). Therefore, you may wish to only use values between 10 and 99, unless you know that your applications will handle 3 digit Node Addresses correctly. (HP BASIC/WS does handle node-specific AUTOST files which contain 3 digits correctly, for example: /SYSTEMS/AUTOST105.) Also, the Pascal Workstation System does NOT know its Client Node Address during boot, and when assigning a "*" directory, and uses instead the unique 6-digit hex number taken from the last six digits of the LAN Link Level Address (e.g., 0012B5).

Note If a given LAN client will be connected to multiple SRM/UX servers from one LAN interface, it should have the same Client Node Address on all of the SRM/UX servers.

The `Server Node Addr` field specifies the emulated SRM server node address that the SRM/UX client can use to connect to this SRM/UX server. Both HP BASIC/WS and Pascal Workstation System clients use this number. HP BASIC/WS users put the `Server Node Address` in MSI specifiers, as in ":REMOTE 21,[Server Node Address]"; for example, ":REMOTE 21, 4". The `Server Node Address` is used in the `ba` parameter of the `tea_srm` function-calls in the Pascal Workstation System CTABLE program (see chapter 7 for details on the `ba` parameter).

Note We strongly recommend assigning ONLY ONE Server Node Address to each server in your SRM/UX system. That is, all entries in the column labeled "Server Node Addr" in each server's /etc/srmdconf LAN-CLIENTS table should contain the same address. Also, each server in your SRM/UX system should have a unique Server Node Address. (If one SRM/UX client is a client of multiple servers that have the same Server Node Address, your SRM/UX system may malfunction.)

The `Client Name` field is optional. Its value is a string, up to 10 characters long, that assigns a unique name to each client, and that can be used by the SRM/UX "front-panel commands", such as *srmdreset*(1M) and *srmdstat*(1), to refer to a client by name rather than by node address. The client name must be unique across both the LAN-CLIENTS and SRM-CLIENTS tables. If a name is not given for a client, then this field must be empty or all blanks, but the colon field separators must remain on both sides of the `Client Name` field.

Note We recommend that you assign a `Client Name` to all clients, especially for larger configurations.

The `Uid` field specifies the HP-UX User Id which will be the owner of all new files and directories created by this Client on this SRM/UX Server (unless overridden by the optional `Uid` field in the VOLUME-TABLE). The Uid is used for all file permissions checks done by the Server. The Uid will also be used for jobs submitted to the HP-UX print spooler for this Client. (A value must be specified in this field, even if overridden in all VOLUME-TABLE entries.)

The `Gid` field specifies the HP-UX Group Id which will be assigned to all new files and directories created by this Client on this SRM/UX Server (unless overridden by the optional `Gid` field in the VOLUME-TABLE). The Gid is used for all file permissions checks done by the Server. The Gid will also be used for jobs submitted to the HP-UX print spooler for this Client. (A value must be specified in this field, even if overridden in all VOLUME-TABLE entries.)

The `Umask` field is optional. If specified, it must be an octal number between 0 and 777. The `Umask` field specifies the HP-UX Umask which will be applied to all new files and directories created by this client on this SRM/UX server. Two common values of Umask are 022, which restricts write permission to the owner of the file, or 002, which restricts write permission to either the owner, or a member of the group specified in the Gid field. If a value for Umask is not specified, then a default value of 0 is used, which will allow anyone to have write permission on the file. (Refer to the *umask*(1) man page in the *HP-UX Reference* for more details.)

The `Volume List` field specifies a list of emulated volumes chosen from the VOLUME-TABLE above. It specifies which volumes in the VOLUME-TABLE can be accessed by each client. The primary way that the SRM/UX System Administrator controls which pieces of the HP-UX file system each client may access is through entries in the `Volume List`, together with the SRM/UX volume structure set up in the VOLUME-TABLE.

Make sure that no two `Volume Names` in a client's `Volume List` have the same `Volume Number` in the VOLUME-TABLE.

Note

If the SRM/UX server allows LAN clients on more than one LAN interface in the server, then you must also modify /etc/rc and /etc/netlinkrc. For instructions on how to do that, see the "HP-UX System Considerations" section earlier in this chapter.

The following table summarizes the configuration rules for the LAN-CLIENTS Table:

Table 4-2. LAN-CLIENTS Table Configuration Guidelines

<table>
<tr><th colspan="2">The string, LAN-CLIENTS, must appear exactly as shown</th></tr>
<tr><th colspan="2">Nine fields are required in each LAN-CLIENTS table entry</th></tr>
<tr><th>Field</th><th>Rules for Valid Entries</th></tr>
<tr><td>Client Link Level Address</td><td>1) must be 0x followed by 12 hexadecimal digits
2) must be unique</td></tr>
<tr><td>Client Internet Address</td><td>1) must be an Internet Address on the same subnet as the Server's Internet Address
2) must be unique</td></tr>
<tr><td>Client Node Addr</td><td>1) must be an integer between 1 and 126, inclusive (we recommend 10..99)
2) recommend using a unique number, but not required</td></tr>
<tr><td>Server Node Addr</td><td>1) must be between 0 and 126, inclusive (we recommend 0..9) (HP BASIC/WS only supports Server Node Addresses in the range 0 to 63, inclusive)</td></tr>
<tr><td>Client Name</td><td>1) must be 10 characters or less
2) must be unique
3) entry not required</td></tr>
<tr><td>Uid</td><td>1) entry is required
2) must be between 0 and 65535, inclusive (0 NOT recommended)</td></tr>
<tr><td>Gid</td><td>1) entry is required
2) must be between 0 and 65535, inclusive</td></tr>
<tr><td>Umask</td><td>1) entry not required (defaults to 0)
2) must be between 000 and 777 (octal), inclusive</td></tr>
<tr><td>Volume List</td><td>1) must contain between 1 and 15 volumes which are defined in the VOLUME-TABLE
2) must not contain duplicate Volume Names
3) each volume in a client's Volume List must have a unique Volume Number</td></tr>
</table>

The SRM-CLIENTS Table

This table in /etc/srmdconf will have entries only on Series 300 servers, because Series 400 and Series 800 servers do not support SRM cards.

Below you will find the detailed explanation of the fields in the SRM-CLIENTS table and valid entries in them. Directly after the explanations you will find a summary table which provides quick access to SRM-CLIENTS table configuration information.

The `Server's SRM Device File` field specifies the full path to the device-special file for the SRM interface on the server's HP-UX file system. The SRM/UX System Administrator should set up the device file by following the instructions in the "Configuring the Server's HP-UX Environment for SRM/UX" section earlier in this chapter.

The `Client Node Addr` field specifies the node address that is set on the switches on the SRM card installed in an HP BASIC/WS or Pascal Workstation System client workstation. Each SRM card on a given SRM connection MUST have a unique Node Address; otherwise the SRM hardware will malfunction.

`Client Node Addresses` in the SRM-CLIENTS table and the LAN-CLIENTS table are completely separate with respect to setting up communication between the server and clients. Also, they do not have to be unique for SRM clients which access the server using different SRM device files. However, having multiple clients with the same `Client Node Address` may cause problems with multiple HP BASIC/WS clients accessing the same node-specific AUTOST files (/SYSTEMS/AUTOSTnn), and may confuse applications which use their `Client Node Address` to generate unique file names, which the application expects to be accessed by only one client. Therefore, HP recommends that all clients in both the LAN-CLIENTS and SRM-CLIENTS tables have unique `Client Node Addresses`, if at all possible.

Note The `Server's SRM Device File` and `Client Node Addr` fields are the (SRM connection) client's core parameters; that is, they uniquely define the client's SRM card. If you change either of these two, you have essentially removed the existing client and have created a new one.

The remaining fields (`Client Name`, `Uid`, `Gid`, `Umask`, and `Volume List`) have the same definitions as the corresponding fields in the LAN-CLIENTS table. Please read the LAN-CLIENTS section of this chapter for details.

The following table summarizes the configuration rules for the SRM-CLIENTS Table:

Table 4-3. SRM-CLIENTS Table Configuration Guidelines

The string, SRM-CLIENTS, must appear exactly as shown	
Seven fields are required in each SRM-CLIENTS table entry	
Field	**Rules for Valid Entries**
Server's SRM Device File	1) entry is required 2) must have an absolute path (starts with /) 3) the file must exist 4) the Device must be character special; if errors here then the *mknod*(1M) command was incorrectly issued 5) the Device must be an SRM Card; if errors here then the *mknod*(1M) command was incorrectly issued 6) the Server must be able to open the Device File
Client Node Addr	1) must be an integer between 1 and 63, inclusive (we recommend 10..63) 2) MUST be unique on a given SRM connection or the hardware will malfunction
Client Name	1) must be 10 characters or less 2) must be unique 3) entry not required
Uid	1) entry is required 2) must be between 0 and 65535, inclusive (0 NOT recommended)
Gid	1) entry is required 2) must be between 0 and 65535, inclusive
Umask	1) entry not required (defaults to 0) 2) must be between 000 and 777 (octal), inclusive
Volume List	1) must contain between 1 and 15 volumes which are defined in the VOLUME-TABLE 2) must not contain duplicate Volume Names 3) each volume in a client's Volume List must have a unique Volume Number

The SPOOL-ENVIRONMENT Table

Below you will find the detailed explanation of the fields in the SPOOL-ENVIRONMENT table and valid entries in them. Directly after the explanations you will find a summary table which provides quick access to SPOOL-ENVIRONMENT table configuration information.

The SPOOL-ENVIRONMENT table declares HP-UX environment variables used by the SRM/UX spooling processes. When a file is closed in a Spooling Directory, *srmd*(1M) will pass an open file descriptor for the file to an HP-UX shell process specified in the SPOOL-TABLE (next section), and then the file is spooled.

The server sets up default values for the following variables: SHELL, TZ, SRMSPOOLFILENAME, PATH, and PRIORITY. You may add additional entries to the SPOOL-ENVIRONMENT table to define additional environment variables for use by the Spool Commands in your SPOOL-TABLE, or you may override the default values for SHELL, TZ, or PATH. This capability provides more power and flexibility for your SRM/UX spooling environment. (See the "Bourne Shell" section of the HP-UX manual *Shells and Miscellaneous Tools* for details on shell and environment variables, and using pipes.)

The first and only field on each line (except for an optional trailing comment) is the variable assignment field. It always has the format:

variable=value

where the allowed *values* depend on which environment variable is being defined. For example, PATH accepts a list of one or more path names separated by colons, as in:

```
PATH=/bin:/usr/bin:/usr/local/bin
```

The following table summarizes the configuration rules for the SPOOL-ENVIRONMENT table:

Table 4-4. SPOOL-ENVIRONMENT Table Configuration Guidelines

<table>
<tr><th colspan="2">The string, SPOOL-ENVIRONMENT, must appear exactly as shown here</th></tr>
<tr><td colspan="2"><b>There are no required entries in the spool-environment table</b></td></tr>
<tr><td colspan="2"><b>Lines in this table must be 255 characters or less</b></td></tr>
<tr><td colspan="2"><b>The total number of environment variables must be 126 or less</b></td></tr>
<tr><td colspan="2"><b>The correct format for entries in this table is: variable=value; however, <i>srmdconfcheck</i>(1M) checks only for the presence of an equals sign in the line</b></td></tr>
<tr><td colspan="2"><b>There are five, pre-initialized environment variables:</b></td></tr>
<tr><td><b>Variable:</b></td><td><b>Comments:</b></td></tr>
<tr><td>1) SHELL=/bin/sh</td><td></td></tr>
<tr><td>2) TZ=MST7MDT</td><td>or the TZ value inherited from the OS</td></tr>
<tr><td>3) SRMSPOOLFILENAME=<i>defaults to the full path name of the file being spooled</i></td><td></td></tr>
<tr><td>4) PATH=/bin:/usr/bin</td><td>PATH should be set to include the paths needed for all commands used in spooler execution. The default PATH includes all utilities supplied with SRM/UX</td></tr>
<tr><td>5) PRIORITY=
<i>or</i>
PRIORITY=-p2 (<i>for example</i>)</td><td>PRIORITY will be set to null for files spooled from the main Spooling Directory, or it will be set to the value specified in the <code>Priority</code> field of the SPOOL-TABLE for files spooled from the PRIORITY subdirectory.</td></tr>
</table>

The SPOOL-TABLE

Below you will find the detailed explanation of the fields in the SPOOL-TABLE and valid entries in them. Directly after the explanations you will find a summary table which provides quick access to SPOOL-TABLE configuration information.

This table in srmdconf provides directions for SRM/UX to use to emulate SRM printer and plotter spooling.

The Spool Directory field specifies the name of the spool directory. It is an absolute path in the SRM/UX file system from which files are to be print-spooled by *srmd*(1M).

The Priority field, which is optional, specifies the Priority SRM/UX will use for jobs from the PRIORITY sub-directory of the Spool Directory. It must have a value between 0 and 7, inclusive. The SRM server supports a "high-priority spooling mechanism" by using an optional PRIORITY sub-directory in Spooling Directories. Files placed in a PRIORITY sub-directory are spooled ahead of files placed in the main Spooling Directory. The same mechanism is supported by the SRM/UX server. *Srmd*(1M) checks for files to be spooled in the PRIORITY sub-directory (if it exists) for each Spool Directory. If the Priority field has a value of 7, for example, then the shell variable PRIORITY is set to '-p7' for each job from the PRIORITY sub-directory. This value (-p7) can be used as an argument to *lp*(1) to specify a higher priority for jobs from the PRIORITY sub-directory, if the $PRIORITY argument is included in the *lp*(1) invocation in the Spool Command (discussed next). *Srmd*(1M) sets the PRIORITY shell variable to the null string for jobs from the main Spool Directory, so they will be submitted with the default priority. If no value is specified in the Priority field, or if the $PRIORITY argument is not included in the *lp*(1) invocation in the Spool Command, then all jobs from that Spool Directory, including the PRIORITY sub-directory, will be submitted with the default priority.

The Spool Command field specifies an HP-UX shell command sequence, to be executed in order to perform the spooling operation. (It will be a Bourne shell sequence unless the SHELL variable has been set to something else in the SPOOL-ENVIRONMENT table. The command sequence is passed to */bin/sh* (or the current value of SHELL) by *srmd*(1M), along with the contents of the spool file. The full path name of the file (e.g. /srmuxroot/PRINTER/Abc-job.UX) is passed in to this command sequence in the environment variable SRMSPOOLFILENAME. See the HP-UX man page for *sh*(1) for shell programming information. The spool file will be removed after the shell command sequence completes.

Note DO NOT put an explicit: rm $SRMSPOOLFILENAME command in the Spool Command field.

The Spool Command is executed with the client's Uid. Therefore, if it is necessary to cancel an *lp*(1) request (for example), root user capability is not required, because jobs are submitted under the client's Uid.

How print spooling is done depends partly on how the HP-UX system's own printer spooling is set up. However, from the HP BASIC/WS or Pascal Workstation System user's point of view, all that is required for proper spooling is to place a file into a Spool Directory on SRM/UX.

The example SPOOL-TABLE from the /etc/srmdconf template shown earlier in this chapter illustrates SRM/UX spooling:

```
SPOOL-TABLE

# Spooling Commands
#
# Spool                 |        |Spool
# Directory             |Priority|Command
#-----------------------+--------+---------------------------------------------------
 /srmuxroot/PRINTER     :     2  :srmdlpfilter | lp $PRIORITY
```

In this example, the PRINTER subdirectory in Volume 8 has been designated as a Spool Directory. The example Spool Command is: srmdlpfilter | lp $PRIORITY. *Srmdlpfilter*(1) is a command, provided as part of the SRM/UX product, which translates non-HP-UX file types (such as LIF ASCII or .TEXT files) to standard HP-UX *vi*-compatible text, suitable for printing by the HP-UX spoolers, such as *lp*(1). It will also prefix the appropriate ENVIRONMENT file to the output, if an ENVIRONMENT subdirectory exists in the Spool Directory.

For example, from a Pascal Workstation System's EDITOR, when a file is ready for printing, you could Quit, Save, and specify a file name such as #5:/PRINTER/ABC-JOB.TEXT. The file would then be printed on the Server's default printer, lp, after *srmdlpfilter*(1) translated the .TEXT format to *vi*-compatible format. The file /PRINTER/ABC-JOB.TEXT will be removed after it is spooled. An HP BASIC/WS user could execute a SAVE or COPY command to place the file to be printed in the /PRINTER directory.

Note The SRM/UX System Administrator does not have to include *srmdlpfilter*(1) in the `Spool Command` pipeline; however, a `Spool Command` which invokes *lp*(1) directly will successfully print only standard HP-UX files. (Standard HP-UX files are created by using the suffix ".UX" in Pascal Workstation System version 3.2 and later, or by using the CREATE statement in HP BASIC/WS.) Also, if *srmdlpfilter*(1) is not included in the `Spool Command` pipeline, ENVIRONMENT files will not be supported.

The following table summarizes the configuration rules for the SPOOL-TABLE:

Table 4-5. SPOOL-TABLE Configuration Guidelines

The string SPOOL-TABLE must appear exactly as shown here	
Three fields are required in each entry in the SPOOL-TABLE; however, no entries are required if no spooling will be done.	
Field	**Rules for Valid Entries**
Spool Directory	1) entry is required 2) must have an absolute path (starts with /) 3) the Spool Directory must exist 4) at least one volume must have access to this Spool Directory
Priority	1) entry is optional 2) must be between 0 and 7, inclusive 3) the default will be whatever your HP-UX spooling priority is set to (default 0; see *lpadmin*(1M))
Spool Command	The Spool Command may be an arbitrary shell pipeline. No error checking is performed on the Spool Command. If you add your own customized Spool Commands, we recommend debugging them as standalone commands before adding them to the SPOOL-TABLE.

An Example /etc/srmdconf File

The following is an example /etc/srmdconf file which illustrates the format and principles for configuring SRM/UX. The discussion following the example explains the details of the entries in each table.

```
VOLUME-TABLE

# Volume Descriptions
#
# Volume         |Volume|     |     |Temp     |Root
# Name           |Number| Uid | Gid |Directory|Directory
#----------------+------+-----+-----+---------+------------------------------------
SRMUXROOT        :  8   : 17  :  9  :         : /srmuxroot
HPUXROOT         :  9   :     :     : /tmp    : /
ABC              :  10  :     :     : /tmp    : /users/abc
DEF              :  10  :     :     : /tmp    : /users/def
GHI              :  10  :     :     : /tmp    : /users/ghi
JKL              :  10  :     :     : /srmtmp : /users/jkl
MNO              :  10  :     :     : /tmp    : /users/mno

LAN-CLIENTS

# LAN Client Descriptions
#
# Client       |Client         |Client|Server|          |     |     |     |
# Link Level   |Internet       |Node  |Node  |Client    |     |     |     |Volume
# Address      |Address        |Addr  |Addr  |Name      | Uid | Gid |Umask|List
#--------------+---------------+------+------+----------+-----+-----+-----+--------
0x080009001234 : 15.99.99.10   : 11   :  0   : Abc      : 101 : 99  : 022 :SRMUXROOT,
HPUXROOT, ABC # User Abc's Workstation

0x080009005678 : 15.99.99.11   : 12   :  0   : Def      : 102 : 99  : 022 :SRMUXROOT,
DEF # User Def's Workstation

0x0800090012AB : 15.99.99.12   : 13   :  0   : Ghi      : 200 : 99  : 022 :SRMUXROOT,
GHI # User Ghi's 1st Workstation

0x080009009876 : 15.99.99.13   : 14   :  0   : Ghi2     : 200 : 99  : 022 :SRMUXROOT,
GHI # User Ghi's 2nd Workstation
```

```
SRM-CLIENTS     # enter SRM clients only on Series 300 Servers

# SRM Client Descriptions
#
#                |Client|          |     |     |     |
# Server's SRM   |Node  |Client    |     |     |     |Volume
# Device File    |Addr  |Name      | Uid | Gid |Umask|List
#----------------+------+----------+-----+-----+-----+-----------------------------
/dev/srm         :  20  :  jkl     : 103 : 99  : 022 : SRMUXROOT,JKL
/dev/srm         :  21  :  mno     : 123 : 99  : 022 : SRMUXROOT,MNO

SPOOL-ENVIRONMENT

# The following environment variables are pre-initialized for spooling
# processes.
#
#       SHELL                  - defaults to /bin/sh, unless overridden
#       TZ                     - inherited from the OS, if it is set there,
#                                otherwise defaults to MST7MDT
#       SRMSPOOLFILENAME       - defaults to the full pathname of the file
#                                being spooled (e.g. /srmuxroot/PRINTER/xyzzy)
#       PATH                   - defaults to /bin:/usr/bin. Recommend that
#                                it be set to include paths needed for all
#                                commands used in Spool Commands.
#----------------------------------------------------------------------------------
SHELL=/bin/ksh

SPOOL-TABLE

# Spooling Commands
#
# Spool                |        |Spool
# Directory            |Priority|Command
#----------------------+--------+---------------------------------------------

/srmuxroot/PRINTER     :   3    : srmdlpfilter | lp -dljet -oraw $PRIORITY
```

Discussion of the Example /etc/srmdconf File

In the above example /etc/srmdconf file, a Pascal Workstation System or HP BASIC/WS client using a LAN interface with Link Level Address 0x080009001234 (remember: the "core" parameter of an SRM/UX client on the LAN connection is its Link Level Address), has access to three volumes: SRMUXROOT, HPUXROOT, and ABC, as specified by this client's `Volume List` in the LAN-CLIENTS table.

Volume SRMUXROOT is defined in the VOLUME-TABLE to be the HP-UX directory /srmuxroot. This client (0x080009001234) always boots up in this emulated Volume 8 (named SRMUXROOT). After boot up, users can move around in Volume 8 with "MSI" commands in HP BASIC/WS or with Udir or Pdir commands in the Pascal Workstation System's Filer.

The client user is not allowed to move above the root of the emulated SRM volume. If the user issued the command to go above the root of the volume—for example: in HP BASIC/WS, `MSI "../../.."`, from /SYSTEMS—he or she would be left at `"/"`(the volume's root directory) from the HP BASIC/WS system's perspective, and at `/srmuxroot` from the server's HP-UX perspective. Trying to move above a volume's `Root Directory` will cause no problems and no error messages will be issued; the attempt will simply be unsuccessful.

The client with Link Address 0x080009001234 has access to two other volumes as well, HPUXROOT and ABC as shown in the `Volume List`. HPUXROOT is assigned to "/" on the HP-UX file system (the HP-UX root directory). This volume provides access to the entire HP-UX file system, if read/write/execute permissions on the HP-UX system so allow. Access to the HP-UX root directory should not be given to clients casually; only those who need it for special reasons should be permitted to access it. HPUXROOT is a volume intended for special uses of SRM/UX; such as SRM/UX clients data-sharing with HP BASIC/UX or other HP-UX applications, or accessing remote file systems mounted via NFS on the SRM/UX server.

Notice, for all clients, that no two `Volume Names` in their `Volume List` may have the same `Volume Number` in the VOLUME-TABLE. (In this example, a client can have access to only one of the volumes: ABC, DEF, GHI, JKL, or MNO because each has a `Volume Number` of 10.)

In this example /etc/srmdconf, in the lines headed by `Volume Names`: ABC, DEF, GHI, JKL, and MNO, the SRM/UX System Administrator has set up volumes for the client users that give them access to their HP-UX "home" directories.

For example: `Volume Name` ABC is assigned to the HP-UX directory /users/abc that user abc gets as his or her "home" directory when logging in to the HP-UX host computer that is acting as the SRM/UX server. Hence, while working as, say, an HP BASIC/WS user he or she could access files under the /users/abc directory by issuing the command: `MSI ":REMOTE 21, 0;LABEL ABC"` (assuming the client's LAN interface's `Select Code` is 21. The 0 in this command represents `Server Node` 0 as specified in the `Server Node Addr` field of this client's line in the LAN-CLIENTS table.)

Files and directories that the SRM/UX clients create on `Volume Name` SRMUXROOT (Volume 8) will be created with Uid=17 and Gid=9, according to the override specified in the VOLUME-TABLE. The VOLUME-TABLE override Uid and Gid values will also be used for all file permissions checks on accesses to SRMUXROOT. For all other Volumes in this example, the client's Uid and Gid fields will control file access permissions and ownership of newly created files and directories, because none of the other Volumes have Uid or Gid fields specified.

Files and directories that client 0x080009001234 creates on both of the other **Volume Names** it has access to (i.e., HPUXROOT and ABC) will be created with Uid=101 and Gid=99, as specified in the Uid and Gid fields of this client's line in the LAN-CLIENTS table.

Any directories client 0x080009001234 creates via any of the volumes in its **Volume List** will have permissions 755 (rwxr-xr-x), because the Umask for the client is set to 022 in the LAN-CLIENTS table; ordinary files will have permissions 644 (rw-r--r--).

The Pascal Workstation System's SRM driver creates temporary files in the /WORKSTATIONS/TEMP_FILES directory under Volume 8. The Temp Directory field of the VOLUME-TABLE provides the flexibility to have SRM/UX place the client's temporary files in a directory other than Volume 8 /WORKSTATIONS/TEMP_FILES; the temp files can be placed anywhere under the server's HP-UX root directory as specified in this field. Each volume listed in the VOLUME-TABLE can have its own directory to store temporary files in.

An HP BASIC/WS client user would use the LABEL field in his or her MSI statement to choose between Volumes SRMUXROOT, HPUXROOT, and ABC for any given mass storage operation.

A Pascal Workstation System client user would not have to modify CTABLE to talk to volume SRMUXROOT, because it is disk Volume 8 on **Server Node Address** 0, which matches the default CTABLE values. However, to access HPUXROOT and/or ABC, he or she would have to modify and recompile CTABLE; because disk volumes 9 and 10 are not defaults of CTABLE. (See chapter 7 of this manual for details on how to do this.)

Users on the other client workstations on the LAN connection see an entirely different view of the same SRM/UX server. They have access to different volumes, and files and directories they create will have different Uids and Gids.

On the SRM connection, this example shows two clients: Client Name jkl and Client Name mno. They are both uniquely determined as clients by the combination of their Server's SRM Device File and Client Node Addr fields.

SRM client jkl, at Client Node Addr 20, has access to volumes SRMUXROOT and JKL. Both clients on the SRM connection are using the same server's device file: /dev/srm. There must be a uniquely named (within its server) device file for each SRM interface in the server. In this example, the server has only one SRM interface and one device file /dev/srm.

Each client on the SRM connection also sees an entirely different view of the same SRM/UX server; in that they have access to different volumes, and files and directories they create will have different Uids and Gids.

In the SPOOL-ENVIRONMENT table, the default value for SHELL has been overridden. The Spool Command will be executed in the Korn shell instead of the default Bourne shell. The variables PATH, TZ and SRMSPOOLFILENAME will be set to their default values.

The example SPOOL-TABLE has configured the directory /srmuxroot/PRINTER as a Spool Directory, with a corresponding Spool Command:

```
srmdlpfilter | lp -dljet -oraw $PRIORITY
```

If a client places a file Abc-job.TEXT in /srmuxroot/PRINTER, then the server will set the shell variable SRMSPOOLFILENAME to /srmuxroot/PRINTER/Abc-job.TEXT and pass an open file descriptor for Abc-job.TEXT to the Spool Command shell sequence. This Spool Command first invokes *srmdlpfilter*(1), which will translate the .TEXT file to *vi*-compatible format, and pipe its output to the HP-UX spooler, *lp*(1), which will print the file on the ljet printer in raw mode.

Validating the Configuration File (/etc/srmdconf)

The SRM/UX "front-panel command", *srmdconfcheck*(1M), validates the new or newly updated /etc/srmdconf file. Use it when initially starting-up the SRM/UX daemon process and when dynamically reconfiguring the Server (i.e., reconfiguring without shutting down the server process).

Note All of the SRM/UX "front panel commands" (EXCEPT *srmdstat*(1)) can be executed ONLY by the super-user (root), including: *srmdconfcheck*(1M), *srmdreconfig*(1M), *srmdreset*(1M), and *srmdshut*(1M).

Srmdconfcheck(1M) attempts to discover all of the errors in /etc/srmdconf on the first pass, however, that may not always be possible. Some errors will mask other errors so that, after fixing the first-level errors, more will be caught while rerunning *srmdconfcheck*(1M). Specifically, if you have more than one error on any line in /etc/srmdconf, *srmdconfcheck*(1M) reports the first error it finds (which may or may not be the first error that occurs in the line from left to right) and quits checking the rest of the line. Hence, it may take several iterations through *srmdconfcheck*(1M) to completely debug and validate your /etc/srmdconf file; depending on how many nested errors there are in the configuration file originally.

To validate your `/etc/srmdconf` file, proceed as follows:

- Execute *srmdconfcheck*(1M) by executing the command:

  ```
  /etc/srmdconfcheck
  ```

  The output will be sent to stdout, however, if there are multiple errors you may want to redirect the *srmdconfcheck*(1M) output to a file so that you have a record of the errors. For example, issue the command:

  ```
      /etc/srmdconfcheck > confcheck.out
  ```

- Below is an example output from *srmdconfcheck*(1M) that results from the /etc/srmdconf file just below the output:

  ```
  Checking SRM/UX configuration file /etc/srmdconf ...
  Illegal link address (0x08000900jfjk) on line 21. Line ignored.
  Srm device /dev/fake_srm on line 32: No such file or directory. Line ignored.
  No client descriptions found.
  Could not complete checking SRM/UX configuration file, 3 errors found.
  ```

  Example /etc/srmdconf that generated the previous error message:

  ```
  VOLUME-TABLE

  # Volume Descriptions
  #
  # Volume         |Volume|     |     |Temp     |Root
  # Name           |Number| Uid | Gid |Directory|Directory
  #----------------+------+-----+-----+---------+------------------------------------
  SRMUXROOT        :  8   :     :     : /tmp    : /srmuxroot
  HPUXROOT         :  9   :     :     : /tmp    : /
  DTESTV           :  10  :     :     : /tmp    : /users/geoff/dtestv

  LAN-CLIENTS

  # LAN Client Descriptions
  #
  # Client       |Client         |Client|Server|          |     |     |     |
  # Link Level   |Internet       |Node  |Node  |Client    |     |     |     |Volume
  # Address      |Address        |Addr  |Addr  |Name      | Uid | Gid |Umask|List
  #--------------+---------------+------+------+----------+-----+-----+-----+--------
  #** "Illegal link level address" error on the line below***************
  0x08000900jfjk : 15.99.99.61   : 11   :  10  : newclient: 3824: 33  : 022  :SRMUXROOT,
  HPUXROOT,DTESTV

  SRM-CLIENTS

  # SRM Client Descriptions
  #
  #                |Client|          |     |     |     |
  # Server's SRM   |Node  |Client    |     |     |     |Volume
  # Device File    |Addr  |Name      | Uid | Gid |Umask|List
  #----------------+------+----------+-----+-----+-----+------------------------------
  #** "non-existent device file" error on the line below***************
  /dev/fake_srm    : 16   :newsrmclnt: 3824: 53  : 022 : SRMUXROOT,DTESTV
  ```

```
SPOOL-ENVIRONMENT

# The following environment variables are pre-initialized for spooling
# processes.
#
#      SHELL                 - defaults to /bin/sh, unless overridden
#      TZ                    - inherited from the OS, if it is set there,
#                              otherwise defaults to MST7MDT
#      SRMSPOOLFILENAME      - defaults to the full pathname of the file
#                              being spooled (e.g. /srmuxroot/PRINTER/xyzzy)
#      PATH                  - defaults to /bin:/usr/bin. Recommend that
#                              it be set to include paths needed for all
#                              commands used in Spool Commands.
#------------------------------------------------------------------------------
PATH=/bin:/usr/bin:          # srmdlpfilter is in /usr/bin

SPOOL-TABLE

# Spooling Commands
#
# Spool                |        |Spool
# Directory            |Priority|Command
#----------------------+--------+----------------------------------------------
 /srmuxroot/PRINTER    :     2  :srmdlpfilter | lp $PRIORITY
```

Notice the error message: `No client descriptions found.` Since there is only one entry in each of the LAN-CLIENTS and SRM-CLIENTS tables, and each of those entries is incorrect, *srmdconfcheck*(1M) reports that there are no valid clients.

You must fix all of the errors. We recommend that you also fix all of the warnings that *srmdconfcheck*(1M) gives.

- Examine the *srmdconfcheck*(1M) output on your screen or in the output file (confcheck.out in this example) and correct the errors reported there.
- Rerun *srmdconfcheck*(1M) until it reports:

  `Checked SRM/UX config file /etc/srmdconf, 0 errors and 0 warnings found.`

You now have a validated configuration file on your SRM/UX server.

If you are starting the SRM/UX server for the first time, and you have filled out the tables in /etc/srmdconf and have run *srmdconfcheck*(1M) successfully, then you are ready to get the server process up and running on the server computer. To do so, see the next section: "Starting the srmd Process on the Server."

Starting the srmd Process on the Server

After you have made the necessary entries in /etc/srmdconf you are ready to start the *srmd*(1M) process on the server.

Manually Starting srmd

To manually start up the *srmd*(1M) server process:

1) You must be logged in as root (super-user)

2) Type `/etc/srmd` at the HP-UX prompt.

Note Only ONE *srmd*(1M) process is allowed at any one time on a server. If you attempt to start multiple *srmd*(1M) processes, you will not get an error message on your terminal or window. However, an entry will be made in `/usr/adm/srmd.log` that looks like this:

```
Mon Dec 10 17:15:29 1990 pid=15313: STARTUP
Mon Dec 10 17:15:29 1990 pid=15313: ABORT: only 1 srmd process allowed at a time
.
Mon Dec 10 17:15:29 1990 pid=15313: TERMINATING
```

Automatically Starting srmd

You can start the server process manually, any time, using the procedures detailed above, and stop it by executing *srmdshut*(1M) (See the section later in this chapter for details on *srmdshut*(1M).) You can also arrange for *srmd*(1M) to come up running, automatically, every time you reboot HP-UX on the server.

The server is started by invoking the SRM/UX server program: `/etc/srmd`.

The following discussion assumes your server's HP-UX system has a standard, "as shipped" version of the `/etc/rc` file installed. If you have customized your `/etc/rc` script, then you probably have a good idea of where the following modifications should be placed in it, based on what events must happen before and after the *srmd*(1M) invocation.

Do the automatic *srmd*(1M) startup by adding the following line to the /etc/rc script:

```
/etc/srmd                   # start-up the SRM/UX server
```

If your server is NOT an HP-UX cluster server, and you want to have SRM/UX clients booting over the LAN connection from your server, then you will also need to add an additional line to the /etc/rc script:

```
rbootd_start                # start-up the HP-UX boot daemon
```

Some of the networking initialization must occur before starting /etc/srmd. Locate the section of /etc/rc shown here in the standard HP-UX /etc/rc file.

```
several hundred lines exist above this...

        # Actions based on system type:
        case $state in
        standalone)     # Not a member of a diskless cluster
                echo "Starting up standalone system"
                swap_start
                set_date
                setmount
                hfsmount
                syncer_start
                lp_start
                clean_ex
                clean_uucp
                net_start
                /etc/srmd     #<<<<<add this line without this comment
                rbootd_start  #<<<<<add this line without this comment
                cron_start
                pty_start
                vt_start
                list_tmps
                clean_adm
                save_core
                delog_start
                audit_start
                ;;

        localroot)      # This is a root server in a Diskless system
                echo "Starting up DISKLESS server ($rootname)"
                swap_start
                set_date
                setmount
                hfsmount
                syncer_start
                lp_start
                clean_ex
                clean_uucp
                net_start
                csp_start
                /etc/srmd   # <<<<<add this line without this comment
                rbootd_start
                cron_start
                pty_start

more lines exist below here...
```

Make the following changes to the shell "case" statement:

- In the "case" where $state is "standalone" (meaning that your server is NOT an HP-UX cluster server), add the /etc/srmd and rbootd_start lines immediately after the line which calls net_start.
- In the "case" where $state is "localroot" (meaning that your server IS an HP-UX cluster server), add the /etc/srmd line between the lines which call csp_start and rbootd_start. (The rbootd_start will already be present in this case; you will not need to add it.)

Since the *update*(1M) procedure for new HP-UX versions does not normally delete an existing /etc/rc file, it is possible that your server is running with an /etc/rc file which pre-dates HP-UX 7.0. In this case, you may wish to update your /etc/rc file to the HP-UX 7.0 version, and make the changes detailed above to automatically start *srmd*(1M), OR you can locate the networking initialization code in your /etc/rc, and add the /etc/srmd line after the networking initialization, but before the diskless boot process, *rbootd*(1M), is started.

Checking the Status of the SRM/UX Clients

The SRM/UX "front panel command", *srmdstat*(1), provides the SRM's FILES, USERS, and NODES functionality. *Srmdstat*(1M) reports the status of the SRM/UX file system; that is, it reports on the status of the client workstations' files.

Srmdstat(1M) reports whether clients are "active" or "not active", and reports on other system status as well (see the discussion below).

A client is "active" if the client has submitted "requests" (e.g., has booted from the server, has opened or closed files, has spooled a file) to the server since the server's latest invocation (i.e., since the SRM/UX System Administrator last issued the /etc/srmd command). If a client has submitted requests to the server since its latest invocation, then it remains active even if you power down the client workstation.

A client is "not active" if the client has NOT submitted requests to the server since the server's latest invocation. There are several reasons why a client might be "not active":

- The client workstation has not powered up since the last server invocation.
- The client has booted from some place other than this server and has not submitted any requests.
- The client is using a local disc and has sent no file system requests to this server.
- The client is using another (i.e., not this server's) remote file system.

Below is a discussion of the *srmdstat*(1) command. It shows example *srmdstat*(1) command lines using all of the options available and the output from those commands.

Note For all of the SRM/UX "front panel commands", you can add `/etc` in your PATH variable and then type just the command name. If `/etc` is not in your PATH, then you must provide the full path as you execute each command: for example,

With `/etc` in PATH: `srmdstat`

Without: `/etc/srmdstat`

Example 1 - (srmdstat)

If you execute:

```
/etc/srmdstat
```

The output might look like this:

```
   test1 at 15.99.99.19 not active
```

```
design1 at 15.99.99.23 has open files:
        /srmuxroot/STATTEST
        /srmuxroot

design3 at 15.99.99.115 has open files:
        /srmuxroot/STATTEST (this client owns LOCK)
        /srmuxroot

JOESCLIENT on /dev/srm at node 16 not active

test5 on /dev/srm at node 53 has 0 open files
```

This example shows the current status of all clients on this server:

- test1 and JOESCLIENT are not active
- design1 has 2 open files: /srmuxroot/STATTEST and /srmuxroot
- design3 has the same 2 open files; in addition it has /srmuxroot/STATTEST LOCKed, so that other clients cannot access the file until design3 releases its lock.
- test5 has no open files

Example 2—(srmdstat -u)

If you execute:

```
/etc/srmdstat -u
```

The output might look like this:

```
design1 at 15.99.99.23 - open file count 2

design3 at 15.99.99.115 - open file count 2
```

The -u option produces a condensed version of the default *srmdstat*(1) output; it reports only those clients which have open files, and tells how many files each client has open.

Example 3—(srmdstat -n)

If you execute:

```
/etc/srmdstat -n design1
```

The output might look like this:

```
   design1 at 15.99.99.23 has open files:
           /srmuxroot/STATTEST
           /srmuxroot
```

The -n (*name*) option reports the status of a single client, the one with the specified name.

Srmdstat(1) reports the same information (for the named client) with this option as it reports with no options specified.

Example 4—(srmdstat -i)

If you execute:

```
/etc/srmdstat -i 15.99.99.115
```

The output might look like this:

```
   design3 at 15.99.99.115 has open files:
           /srmuxroot/STATTEST (this client owns LOCK)
           /srmuxroot
```

The -i option is similar to the -n option, except that the `Client Internet Address` is used to specify the LAN client you want a report on.

Example 5—(srmdstat -s)

If you execute:

```
/etc/srmdstat -s /dev/srm 53
```

The output might look like this:

```
   test5 on /dev/srm at node 53 has 0 open files
```

The -s option is similar to the -n option, except that the `Server's SRM Device File` and `Client Node Address` are used to specify the SRM client you want a report on.

Example 6—(srmdstat -f)

If you execute:

```
/etc/srmdstat -f /srmuxroot/STATTEST
```

The output might look like this:

```
/srmuxroot/STATTEST open by:
        design1 at 15.99.99.23
        design3 at 15.99.99.115 owns LOCK
```

The -f option is used to show which clients are using particular file(s). It also tells which client, if any, has the file LOCKed.

The SRM/UX System Administrator can use this option to see if one client is holding onto a LOCK inappropriately. In this example, the Administrator could do an *srmdreset*(1M) on design3 to clear the LOCK thereby allowing design1 use of the file /srmuxroot/STATTEST.

Notice how *srmdstat*(1) reports each client's `Client Name`, and `Client Internet Address` (for LAN) and `Client Name`, `Server's SRM Device File`, and `Client Node Addr` (for SRM); in addition to the other information detailed above.

Resetting SRM/UX Clients

Srmdreset(1M) gives the SRM/UX System Administrator explicit control over what files and LOCKs are in use by each Client. (*Srmdreset*(1M) emulates the SRM's REMOVE_USER command.) The SRM/UX System Administrator can use *srmdreset*(1M) to unlock and close specific files, or to unlock and close all of the files open by a particular client. The following examples illustrate the various ways to use *srmdreset*(1M):

Example 1—(srmdreset -n)

If you execute:

```
/etc/srmdreset -n design3 /srmuxroot/STATTEST
```

Then:

The above command closes the file /srmuxroot/STATTEST for client design3 and also releases any LOCKs that design3 had on this file. Any other clients that have the file open, will still have it open. Any other files that client design3 has open will remain open.

Example 2—(srmdreset -n)

If you execute:

```
/etc/srmdreset -n design3
```

Then:

The above command closes all files open by client design3. It will also release all LOCKs held by design3. All other clients are unaffected. (If other clients were kept from accessing files by LOCKs held by design3, they will now be able to access those files.)

Note *Srmdreset*(1M) does NOT disable future requests by the Clients which get reset. In order to completely disable a client from accessing the SRM/UX Server, the SRM/UX System Administrator must edit /etc/srmdconf to delete or "comment out" the line defining the Client, and then execute *srmdreconfig*(1M)

Example 3—(srmdreset -i)

If you execute:

```
/etc/srmdreset -i 15.99.99.115
```

Then:

Example 3 is the same as Example 2, except that the LAN client to be reset is specified here by using the Client Internet Address instead of the Client Name.

Example 4—(srmdreset -s)

If you execute:

```
/etc/srmdreset -s /dev/srm 53
```

Then:

Example 4 is similar to Example 3. The difference is that you specify the SRM client to be reset by using the server's SRM Device File and Client Node Address instead of the Client Name.

Example 5—(srmdreset -a)

If you execute:

```
/etc/srmdreset -a /srmserve/STATTEST
```

Then:

The above command closes the file /srmuxroot/STATTEST on ALL Clients that have it open. No other files are affected.

Example 6—(srmdreset -a)

If you execute:

```
/etc/srmdreset -a
```

Then:

The above command closes ALL files that ALL Clients have open. (Executing the above command is essentially the same as executing *srmdshut*(1M), and then restarting the server.)

Note

On your HP BASIC/WS clients, after you execute *srmdreset*(1M), you may have closed the directory file in which the client user was working. The BASIC/WS user can use the following command to re-open his current directory:

```
MSI SYSTEM$("MSI")
```

Reconfiguring a Running SRM/UX Server (Dynamic Reconfiguration)

Some typical reasons for reconfiguring the server are: adding a new client, deleting a client, modifying the parameters of an existing client (for example, adding a new volume to a client's `Volume List` in either the LAN- or SRM-CLIENTS table of srmdconf), changing a `Spooling Directory`, adding or deleting a printer or plotter, etc. SRM/UX allows making changes to individual client's configurations without disturbing other clients on the overall SRM/UX system.

Note

Once you have configured and started the server with the original SRM/UX system and are ready to make changes to that system, you are ready to reconfigure the running server.

As you perform all of the procedures detailed below, make a copy of the /etc/srmdconf file, then edit and validate (with *srmdconfcheck*(1M), using the -f option: for example, `/etc/srmdconfcheck -f srmdconf_copy`) the copy file. This will avoid complications caused by changing the clients' identities before the system "catches up" with what you've done.

The following discussion will give you a detailed understanding of what is involved in dynamic reconfiguration.

Principles of Dynamic Reconfiguration

1. The definition of an SRM/UX client varies, depending on whether the client workstation is connected to the server over the LAN connection or over the SRM connection. On the LAN connection the client is defined by its Link Level Address, (`Client Link Level Address` in the LAN-CLIENTS table in srmdconf). All of the other entries in the client's line are tied to this essential parameter. On the SRM connection the client is defined by both its device file (`Server's SRM Device File` in the SRM-CLIENTS table in srmdconf) and its client node address (`Client Node Addr`). All of the other entries in the client's line are tied to these two essential parameters.

   As a result, if you make any changes to the `Client Link Level Address` (for LAN) or the `Server's SRM Device File` and/or `Client Node Addr` (for SRM) you have created a *NEW CLIENT*. In other words, once you change (by editing the appropriate field in the srmdconf file) one of these "core" client parameters, you have eliminated the previously existing client and have created an entirely new client, even if the rest of the line in srmdconf remains the same.

2. Clients can be "active" or "not active".

   A client is "active" if the client has submitted "requests" (e.g., has booted from the server, has opened or closed files, has spooled a file) to the server since the server's latest invocation (i.e., since the SRM/UX System Administrator last issued the /etc/srmd command). If a client has submitted requests to the server since its latest invocation, then it remains active even if you power down the client workstation. To discover if clients are "active" or "not active", run *srmdstat*(1).

   A client is "not active" if the client has NOT submitted requests to the server since the server's latest invocation. There are several reasons why a client might be "not active":

   a. The client workstation has not powered up since the last server invocation.

   b. The client has booted from some place other than this server and has not submitted any requests.

   c. The client is using a local disc and has sent no file system requests to this server.

d. The client is using another (i.e., not this server's) remote file system.

3. Several modifications to the /etc/srmdconf file are effective ONLY for *new requests* to the server, not for already existing requests, such as already open files (details below).

4. You will use a combination of the following SRM/UX "front panel commands" to accomplish the dynamic reconfiguration of your SRM/UX system: *srmdconfcheck*(1M), *srmdreset*(1M) and *srmdreconfig*(1M).

Use of srmdconfcheck(1M)

Always run *srmdconfcheck*(1M) after you make changes to /etc/srmdconf. It verifies the correctness of the entries in the tables and ensures that the changes you have made will work.

Use of srmdreconfig(1M):

Srmdreconfig(1M) tells the SRM/UX daemon, *srmd*(1M), to reread the /etc/srmdconf file and reconfigure itself, without shutting down and restarting. The advantage of dynamic reconfiguration is that clients whose configurations have not been modified by the revised /etc/srmdconf can continue normal operations, uninterrupted by an otherwise necessary shut-down of *srmd*(1M).

If you make modifications to a client's line in the LAN or SRM-CLIENTS tables, the "core" client parameters (as described above) must be correct when you run *srmdreconfig*(1M), otherwise:

- All of the client's files are closed.

- No new requests will be accepted from that client.

Srmdreconfig(1M) reports any modifications that it cannot carry out in the `/usr/adm/srmd.log` file.

If you are modifying the configuration of any client that you wish to reboot from this server, be aware that the HP-UX boot daemon, rbootd, continuously scans the /etc/srmdconf file for valid, bootable clients. Therefore, if you change the "core" parameter for a LAN connection client (i.e., the client Link Level Address), and then you try to reboot the applicable client, its Boot ROM will no longer report this server as a bootable system host, and you will have effectively deleted your client's ability to boot from this SRM/UX server over the applicable LAN connection. This is one of the main reasons to modify a *copy* of srmdconf and verify it first, before you rename it to /etc/srmdconf.

Details of Dynamic Reconfiguration

Note The changes discussed below take effect only after restarting *srmd*(1M) or executing *srmdreconfig*(1M).

The VOLUME-TABLE

You can modify any field in the VOLUME-TABLE and the changes will take immediate effect for *new* client requests, but the changes will not take effect for already open files.

The LAN-CLIENTS Table

The `Client Link Level Address` is a permanent part (i.e., is kept in ROM) of the LAN interface installed in the client workstation and "anchors" the client's identity when communicating over the LAN. That is, an SRM/UX LAN connection client is bound to the `Client Link Level Address` as set in its interface card and as correctly entered in the /etc/srmdconf file. Therefore, any change to the `Client Link Level Address` field constitutes deleting the old client and adding a new client.

If you make changes to fields other than the `Client Link Level Address` field, then whether the client is "active" or "not active" affects the reconfiguration process.

You can change the following three fields in the LAN-CLIENTS table only when the client's state is "not active". These fields are:

- `Client Internet Address`
- `Client Node Addr`
- `Server Node Addr`

Note

Srmd(1M) will NOT acknowledge the changes to the `Client Internet Address`, `Client Node Address`, or `Server Node Address` fields in the LAN-CLIENTS table after you issue the command *srmdreset*(1M), followed by *srmdreconfig*(1M). There are two ways to force *srmd*(1M) to acknowledge changes in these fields:

1. The preferred way is to:
   a. Delete the client by:
      i. Editing the /etc/srmdconf file (either delete the client's line or comment it out with a # at the beginning of the line)
      ii. Running *srmdreconfig*(1M)
   b. Then add the (modified) client by:
      i. Editing the /etc/srmdconf file to include the desired changes
      ii. Running *srmdreconfig*(1M)
2. A less desirable way is to:

   Execute *srmdshut*(1M) to shut down the server, then execute /etc/srmd to restart the server; upon restart *srmd*(1M) will acknowledge the changes. This method is undesirable because you must shut down the server for all clients, not just the one(s) you are reconfiguring.

You can change the rest of the fields in the LAN-CLIENTS table any time, whether the client is active or not active, and the changes will be effective for new requests immediately upon executing *srmdreconfig*(1M), but not for existing open files. To make them effective for all requests, first execute *srmdreset*(1M) on that client using the -n <name> or -i <internet> options (to close the client's open files), then execute *srmdreconfig*(1M).

These fields (which are coupled with open files) are:

- `Uid`
- `Gid`

- `Volume List`

The SRM-CLIENTS Table

The `Server's SRM Device File` and `Client Node Addr` pair "anchor" the client's identity when communicating over the SRM connection. That is, to SRM/UX an SRM client is bound to the `Server's SRM Device File` (created with the *mknod*(1M) command) and `Client Node Addr` (set on its interface card with DIP switches) and the correct setting for both being entered in the /etc/srmdconf file. Therefore, any change to either of these fields constitutes deleting the old client and adding a new client.

You can change the rest of the fields in the SRM-CLIENTS table any time, whether the client is "active" or "not active", and the changes will be effective for new requests immediately upon executing *srmdreconfig*(1M), but not for existing requests. To make them effective for all requests, first execute *srmdreset*(1M) on the client using -n <name> or -s <device file> options (to close the client's open files), then execute *srmdreconfig*(1M).

These fields (which are coupled with open files) are:

- `Uid`
- `Gid`
- `Volume List`

The SPOOL-ENVIRONMENT Table

You can change any line in the SPOOL-ENVIRONMENT table in /etc/srmdconf and the changes will automatically take place upon the next spool request.

The SPOOL-TABLE

Files are spooled from the spool directory when they are closed.

There are two special cases when dynamically reconfiguring the SPOOL-TABLE:

1. If you delete a `Spool Directory` by deleting or modifying a line in the SPOOL-TABLE in /etc/srmdconf and there are open files in the old `Spool Directory` when you execute *srmdreconfig*(1M), then the open files will not be spooled when they are closed.

2. If you define a new `Spool Directory` by adding or modifying a line in the SPOOL-TABLE in /etc/srmdconf, and there are already open files in the new `Spool Directory` when you execute *srmdreconfig*(1M), then the open files will be spooled when they are closed.

You can change any of the fields in the SPOOL-TABLE and the changes will take effect immediately, with the above considerations.

Stopping the Server Process

You will want to stop and restart the server process, *srmd*(1M), if you have made configuration changes that cannot be easily handled with dynamic reconfiguration. Stopping and restarting *srmd*(1M) will be the easiest way to handle large reconfiguration tasks, such as changing the `Client Internet Addresses` for all of your LAN Clients.

The SRM/UX "front panel command" *srmdshut*(1M) provides the SRM's SYSTEM DOWN functionality. However, the SRM's powerfail option is NOT supported.

Srmdshut(1M) will do a "graceful" shutdown of the SRM/UX daemon. Incoming client requests will be disabled, and then all open files will be unlocked and closed before terminating the daemon process. The daemon will wait for a brief time for spooling activity to complete so that spool files can be removed from spool directories. If spooling operations have not completed when the daemon terminates, then spool files will remain in the Spool Directory. (A list of these files will be put in /usr/adm/srmd.log before the daemon terminates.) *Srmdshut*(1M) will wait for the daemon to terminate.

Files in `Spool Directories` will not be spooled when they are closed by *srmdshut*(1M). If the `Spool Directory` has a subdirectory called "ABORTED", then the partial spool file will be moved to that directory (replacing any previous file with the same name). If there is no "ABORTED" sub-directory, or the move operation fails, then the partial spool file will be purged. (A list of spool files moved to an "ABORTED" sub-directory, or purged, will be written to /usr/adm/srmd.log before the daemon terminates.)

PRIORITY subdirectories can also have "ABORTED" subdirectories.

Only the super-user (root) can execute *srmdshut*(1M).

Stopping the Server's HP-UX System and Host Computer

Since SRM/UX is a daemon process running in the background on an otherwise fully usable HP-UX workstation, you can shut down SRM/UX without shutting down either HP-UX or the host computer's power.

If you need to shut down HP-UX and the host computer's power, see the *HP-UX System Administration Tasks* manual for those procedures.

Restarting the Server Process

If you have shut down both HP-UX and the host computer, follow the steps in the *HP-UX System Administration Tasks* manual to bring them back up.

To manually restart up the *srmd*(1M) server process:

1) You must be logged in as root (super-user)

2) Type `/etc/srmd` at the HP-UX prompt.

Note Only one instance of *srmd*(1M) is allowed at a time on a server.

If you previously performed the procedure detailed in the "Automatically Starting srmd" section of this chapter, then the *srmd*(1M) process will automatically restart when you bring up the host computer and boot HP-UX. If you did not perform that procedure, then you can do so after you reboot HP-UX and *srmd*(1M) will automatically restart the next time you boot HP-UX.

5

Booting Clients From the SRM/UX Server

Installing the capability for clients to boot from the server requires action from the server side and from the client side. The following is a discussion of the overall model. It talks about the whole process without separating out the server side from the client side. Then in the procedures section towards the end of the chapter, the server tasks are separated from the client tasks.

When you are installing the ability for clients to boot from the server, most of the procedures can be done on the server and one client.

Both the server administrator and the client users should read the first half of this chapter—up to "The Procedures" section. Only those who are responsible for the technical implementation details need to read "The Procedures" section.

The Model On the SRM Connection

HP BASIC/WS Clients

To begin the process of booting the HP BASIC/WS from the SRM/UX server, the client workstation's Boot ROM establishes communication with the server process, *srmd*(1M). The Boot ROM then looks at the /SYSTEMS directory in Volume 8 (as defined in the server's /etc/srmdconf file) and lists the available system boot files on the client's screen (if you enter the attended boot mode), or the Boot ROM boots the first system boot file it finds in /SYSTEMS (if allowed to run in the unattended mode).

For clients to automatically boot in unattended mode, the SRM/UX system administrator must ensure that the desired system boot file is the first one found in the server's /SYSTEMS directory under Volume 8. See the

Procedures section towards the end of this chapter for details on ensuring that a system boot file is the first file in an HP-UX directory.

There are two critical HP BASIC/WS binaries that are relevant to booting HP BASIC/WS clients on the SRM connection: the SRM and DCOMM binaries. It is essential that both of these binaries be loaded into the HP BASIC/WS language system so that it can communicate directly with *srmd*(1M) and, therefore, work on the SRM/UX system. A special system boot file named SYSB6_INSTALL has been installed by the SRM/UX installation process on your server in the /SYSTEMS directory under Volume 8. The boot file SYSB6_INSTALL contains the following binaries: SRM, DCOMM, LAN, CRTA, CRTB, HFS, ERR, DISC, CS80, HP9885, HPIB, FHPIB, and SCSI. It has been designed to get your first HP BASIC/WS client up on the SRM/UX system as quickly and easily as possible. After SYSB6_INSTALL has booted, you can load as many other binaries as you need (from your HP BASIC/WS local mass storage) to create the "customized" or "extended" system that works best for the client users.

Note SYSB6_INSTALL is used only for installing the 6.1 version of HP BASIC/WS.

Sometimes, you may have a system boot file which contains a "minimal" system; that is, just the "core" system, CRTA, and CRTB. You could name such a boot file, SYSTEM_BA6, SYS_MIN, etc. Since this minimal file does not contain the critical binaries SRM and DCOMM, the HP BASIC/WS system is unable to communicate directly with *srmd*(1M). So when the client user executes the necessary LOAD BIN commands to load both binaries, HP BASIC/WS invokes Boot ROM code to do the loading. Then, once it has loaded both binaries, HP BASIC/WS can communicate directly with *srmd*(1M). You can then load as many other binaries as you need to create the "customized" or "extended" system that works best for the client users.

HP recommends that you store the SRM and DCOMM binaries with any system boot file that you will be booting from SRM/UX over the SRM connection. Doing so allows the HP BASIC/WS client to communicate with the server at boot time.

Pascal Workstation System Clients

To begin the process of booting the Pascal Workstation System from the SRM/UX server, the client workstation's Boot ROM establishes communication with the server process, *srmd*(1M). The Boot ROM then looks at the /SYSTEMS directory in Volume 8 (as defined in the server's /etc/srmdconf file) and lists the available system boot files on the client's screen (if you enter the attended boot mode), or the Boot ROM boots the first system boot file it finds in /SYSTEMS (if allowed to run in the unattended mode).

For clients to automatically boot in unattended mode, the SRM/UX system administrator must ensure that the desired system boot file is the first one found in the server's /SYSTEMS directory under Volume 8. See the Procedures section towards the end of this chapter for details on ensuring that a system boot file is the first file in an HP-UX directory.

There are two critical modules needed for communication between a Pascal client and *srmd*(1M) on the SRM connection: the SRM and DATA_COMM modules. If these modules are not present in your INITLIB file the client can't communicate with the server after booting. In that case, you can execute DATA_COMM and SRM (in that order) on the client from local mass storage.

HP recommends placing the DATA_COMM and SRM modules in each user's INITLIB file so that the process of booting establishes connection to the server. In general, each client user's INITLIB file will be a customized version of the INITLIB file that is included on the client's BOOT: or BOOT2: flexible discs.

The Model On the LAN Connection

HP BASIC/WS Clients

To begin the process of booting the HP BASIC/WS over LAN from the SRM/UX server, the client workstation's Boot ROM establishes contact with the server by first contacting the HP-UX rbootd daemon. Rbootd looks at the `paws-srm` and `basic-srm` lines in the server's /etc/boottab file to see which system boot file names the SRM/UX system administrator has placed there, specifically in the fourth field of the line. The Boot ROM lists them on the client's screen in the order they appear in the line (if you enter the attended boot mode), or the Boot ROM boots the first system boot file listed in the **paws-srm** line in /etc/boottab (if allowed to run in the unattended mode). Putting a system boot file name first in the **paws-srm** line is what provides the capability of doing "unattended" or "automatic" client booting over LAN of either the HP BASIC/WS or the Pascal Workstation System.

There are two critical HP BASIC/WS binaries that are relevant to booting HP BASIC/WS clients on the LAN connection: the SRM and LAN binaries. It is essential that both of these binaries be loaded into the HP BASIC/WS language system so that it can communicate directly with *srmd*(1M) and, therefore, work on the SRM/UX system. A special system boot file named SYSB6_INSTALL has been installed by the SRM/UX installation process on your server in the /SYSTEMS directory under Volume 8 . The boot file SYSB6_INSTALL contains the following binaries: SRM, DCOMM, LAN, CRTA, CRTB, HFS, ERR, DISC, CS80, HP9885, HPIB, FHPIB, and SCSI. It has been designed to get your first HP BASIC/WS client up on the SRM/UX system as quickly and easily as possible. After SYSB6_INSTALL has booted, you can load as many other binaries as you need (from your HP BASIC/WS local disc or other mass storage) to create the "customized" or "extended" system that works best for the client users.

Note SYSB6_INSTALL is used only for installing the 6.1 version of HP BASIC/WS.

Sometimes, you may use a system boot file which contains a "minimal" system, that is, just the "core" system, CRTA, and CRTB. You could call such a boot file SYSTEM_BA6, SYS_MIN, etc. Since this minimal file does not contain the critical binaries SRM and LAN, the HP BASIC/WS system is unable to communicate with *srmd*(1M). So, when the client user executes the necessary LOAD BIN commands to load both binaries, HP BASIC/WS invokes Boot ROM code to do the loading with the help of rbootd. In this case, you must tell rbootd where the SRM and LAN binaries are on the server's SRM/UX file system. You tell rbootd the path name of both binaries, relative to the root directory of Volume 8, in the fifth (last) field of the `paws-srm` line of /etc/boottab. You can also do this in the fifth (last) field of the `basic-srm` line of /etc/boottab, in special cases.

Then, once it has loaded both binaries, HP BASIC/WS can communicate directly with *srmd*(1M) and you can load as many other binaries as you need to create the "customized" or "extended" system that works best for the client users.

If the system boot file that is first loaded by the Boot ROM and rbootd contains the SRM and LAN binaries, then the HP BASIC/WS client can communicate directly with *srmd*(1M) right away, and no entries are necessary in the fifth (last) field of the `paws-srm` and `basic-srm` lines. For this reason, HP recommends that you store the SRM and LAN binaries with any system boot file that you will be booting from SRM/UX over the LAN connection. Doing so allows the HP BASIC/WS client to communicate with the server at boot time.

Pascal Workstation System Clients

To begin the process of booting the Pascal workstation system over LAN from the SRM/UX server, the client workstation's Boot ROM establishes contact with the server by first contacting the rbootd daemon. Rbootd looks at the `paws-srm` line in the server's /etc/boottab file to see which system boot file names the SRM/UX system administrator has placed there, specifically in the fourth field of the line. Whichever system boot file names are there, the Boot ROM lists them on the client's screen in the order they appear in the line (if you enter the attended boot mode), or the Boot ROM boots the first system boot file listed in the `paws-srm` line in /etc/boottab (if allowed to run in the unattended mode). Putting a system boot file name first in the `paws-srm` line is what provides the capability of doing "unattended" or "automatic" client booting over LAN of either the HP BASIC/WS or the Pascal Workstation System. (See the "Automatically Booting a Preferred System" section towards the end of this chapter for details.)

The critical INITLIB modules needed for proper communication between *srmd*(1M) and the Pascal client are: DATA_COMM, IOMPX, LANDVR, and SRM. The DATA_COMM module is contained in the file DATA_COMM (the CONFIG: disc for single-sided media and the ACCESS: disc for double-sided media), the IOMPX and LANDVR modules are contained in the file LAN (the LIB: disc for single-sided media and the ACCESS: for double-sided media), and the SRM module is contained in the file SRM (the CONFIG: disc for single-sided media and the ACCESS: disc for double-sided media). You can execute these files manually to establish initial contact with *srmd*(1M) after you boot locally, however, these modules must be present in the client's INITLIB file to ensure correct booting from the SRM/UX server.

Both Clients—The General Case

You can insert two lines in the server's /etc/boottab file that relate to client booting over the LAN connection. They are: the `paws-srm` line (if there are any Pascal Workstation System clients booting over LAN), and the `basic-srm` line (if there are any HP BASIC/WS clients booting over LAN). At most, one line of each type is allowed in /etc/boottab.

You can simplify the task of modifying /etc/boottab by using only one line—the `paws-srm` line. All of the necessary functionality provided by the two lines (`basic-srm` and `paws-srm`) has been combined into one line called (by convention) `paws-srm`. Through this one line you can direct rbootd with information about both HP BASIC/WS and Pascal workstation system clients.

Note

HP recommends using only the `paws-srm` line in /etc/boottab to handle all client booting over LAN, unless you have special SRM/UX system needs, such as a large number of system boot files and/or autostart files. Read the restrictions below to see if you need to use more than just the `paws-srm` line.

Lines in /etc/boottab can have, at most, 1024 characters; therefore, at 80 characters per line, you may run into trouble if your boottab line is 12 or more lines long. The fourth field (which holds the names of the system boot files) can have a maximum of 32 comma-separated entries. The fifth field (which holds the names of the autostart files and/or the names of the HP BASIC/WS binaries and/or the names of the Pascal workstation system TABLE, STARTUP, and INITLIB files) can have a maximum of 32 comma-separated entries.

You will find two subsections below describing the /etc/boottab entries necessary for booting the clients over LAN. The first subsection (The Recommended Method of Configuring boottab) describes the recommended and preferred method for configuring the /etc/boottab file. It describes the concepts for modifying a single line—`paws-srm`—that handles all cases; that is, HP BASIC/WS client booting, Pascal workstation system client booting, and automatically booting a preferred system—either HP BASIC/WS or Pascal workstation system. The second subsection (Another Method of Configuring boottab) describes the concepts for modifying the **basic-srm** line to boot HP BASIC/WS clients, in the unlikely event that you need to use the **basic-srm** line (See the note above).

The Recommended Method of Configuring boottab

A typical **paws_srm** line looks like this:

```
paws-srm:HPS300: :SYSB6_INSTALL,SYSTEM_BA6,SYSCUSTOM_B,SYSTEM_P:
/AUTOST,/RMBBINS/SRM,/RMBBINS/LAN,INITLIB,STARTUP,TABLE
```

The printed output appears here on multiple lines, but there is only 1 line in the actual /etc/boottab file

The fields in this line are separated by colons. The last two fields allow multiple entries, separated by commas.

- The first field, **paws-srm**, tells rbootd that the file **/etc/srmdconf** exists and should be processed.
- The second field is the client machine type field, which must be HPS300 for both HP BASIC/WS and Pascal Workstation System clients.
- The third field is empty (spaces are allowed between the colons for this field, but none are required).
- The fourth field is a list of one or more system boot files, separated by commas. These files must be in directory /SYSTEMS on Volume 8 as declared in the SRM/UX **/etc/srmdconf** file (described in detail in chapter 4 of this manual). The /SYSTEMS directory in Volume 8 is the only valid location for system boot files for both HP BASIC/WS and Pascal workstation system clients, therefore, you do not have to specify a path name for the system boot files in this fourth field.

  The HP BASIC/WS system boot file names in the fourth field of the example line above are: SYSB6_INSTALL, SYSTEM_BA6, and SYSCUSTOM_B. They have been customized to contain special binaries and will be used for different clients. For example, SYSB6_INSTALL is a system that contains the "core" and all of the mass storage driver binaries.

SYSTEM_BA6 is a minimal file containing the "core" system, CRTA, and CRTB. SYSCUSTOM_B is an in-between file containing the "core" system and a particular set of binaries that one set of client users find useful.

The Pascal Workstation System boot file in the example line above is named SYSTEM_P.

- The fifth field contains the names of the three standard Pascal extension boot files, and the HP BASIC/WS general autostart file /AUTOST, together with the HP BASIC/WS binaries SRM and LAN (which can be anywhere under Volume 8 and so require the full path name from the Volume root directory—here the binaries are all kept under /RMBBINS). The SRM and LAN binaries are included to allow the minimal system (SYSTEM_BA6) to LOAD BIN and establish communication with SRM/UX over LAN.

  The rules for specifying path names in the fifth field are different for HP BASIC/WS and Pascal Workstation System clients: you must specify complete path names with respect to Volume 8 for HP BASIC/WS (e.g., specify /RMBBINS/SRM instead of SRM), but you must NOT specify complete path names for Pascal Workstation System (e.g., specify INITLIB instead of /WORKSTATIONS/SYSTEM/INITLIB).

The Pascal extension boot files must be on Volume 8, just like the SYSTEM_P file, and they must be in either /WORKSTATIONS/SYSTEM or /WORKSTATIONS/SYSTEMnnnnnn (where nnnnnn is the six character emulated client node address—see chapter 4 for details).

Note Any alpha characters MUST be capitalized in the name of this directory; for example: /WORKSTATIONS/SYSTEM012ABC is correct, but /WORKSTATIONS/SYSTEM012abc is not.

For a discussion of booting from the clients' local mass storage, and building system boot files, etc. see chapter 6 for HP BASIC/WS clients, and chapter 7 for Pascal workstation system clients.

Another Method of Configuring boottab

If you have special SRM/UX system requirements that make using only the `paws-srm` line in /etc/boottab impossible, then use the `basic-srm` line.

Note Even though you can place HP BASIC/WS boot information in the /etc/boottab paws-srm line, you may NOT place Pascal workstation system boot information in the basic-srm line.

A typical basic-srm line looks like this:

```
basic-srm:HPS300::SYSTEM_BA6,SYSBA6_SRM,SYSTEM_BA6_ALL:
/AUTOST,/RMBBINS/SRM,/RMBBINS/LAN
```

The fields in this line are separated by colons. The last two fields allow multiple entries, which are separated by commas.

- The first field, basic-srm, tells rbootd that the file /etc/srmdconf exists and should be processed.
- The second field is the client machine type. The only legal value for SRM/UX in this field is HPS300.
- The third field is always empty. (spaces are allowed between the colons for this field, but none are required)
- The fourth field is a list of one or more system boot files. These system boot files MUST be in directory /SYSTEMS on Volume 8 as declared in the /etc/srmdconf file (described in detail in chapter 4).
- The fifth field contains the HP BASIC/WS general autostart file /AUTOST, together with the HP BASIC/WS binaries SRM and LAN (which can be anywhere under Volume 8 and so require the full path name from the Volume root directory—here the binaries are all kept under /RMBBINS on Volume 8). The SRM and LAN binaries are included to allow the minimal system (SYSTEM_BA6) to LOAD BIN and establish communication with SRM/UX over LAN.

What the Client User Needs to Know About the Server

There is a special file on the SRM/UX server called /etc/srmdconf that can be modified only by the SRM/UX system administrator. Through entries in a table in this file, the administrator assigns "Volume Names" and "Volume Numbers" to directories in the server's HP-UX file system. In this table, Volume Number 8 is special: it is reserved as a place for the clients (on both the LAN and SRM connections) to store their files and directories that are used in the booting process.

Both HP BASIC/WS and Pascal workstation system clients should store their system boot files in /SYSTEMS under Volume 8.

Both clients should store their autostart files under the Volume 8 root directory.

For example:

- For a Pascal Workstation System LAN client, store /WORKSTATIONS/SYSTEM012345/AUTOSTART under Volume 8
- For a Pascal Workstation System SRM client, store /WORKSTATIONS/SYSTEM15/AUTOSTART under Volume 8
- For an HP BASIC/WS client on either connection, store /SYSTEMS/AUTOST23 under Volume 8

All HP BASIC/WS client users should ask the SRM/UX System Administrator what the volume name assigned to Volume 8 is, so they can MSI to Volume 8 /SYSTEMS before they create customized system boot files using LOAD BIN and STORE SYSTEM commands.

HP recommends storing the HP BASIC/WS driver and language system binaries in /RMBBINS under Volume 8.

The Pascal Workstation System INITLIB drivers are kept under /WORKSTATIONS/SYSTEM in Volume 8.

There are two other tables in the server's /etc/srmdconf file that the client users should know about for booting purposes: the LAN-CLIENTS table and the SRM-CLIENTS table. Client users should ask the SRM/UX System

Administrator to ensure that the `Volume Name` for Volume 8 appears in the `Volume List` on their client's line in either the LAN-CLIENTS table or the SRM-CLIENTS table, depending on which connection is applicable.

Clients should ask the SRM/UX System Administrator to ensure that the correct entries for their needs are in the `paws-srm` line (and for large systems the `basic-srm` line) in /etc/boottab.

What the Server Administrator Needs to Know About the Clients

About HP BASIC/WS Clients

The HP BASIC/WS system is a self-contained operating system and language system that runs on HP computers, and provides a BASIC language programming environment for various project applications.

The HP BASIC/WS system contains a "core" system which is augmented by various language system and driver binaries. Any number of binaries can be combined into a system boot file which the client workstation can boot up and run under.

Typical HP BASIC/WS binaries are called: CRTA, CRTB, SRM, LAN, DCOMM, IO, TRANS, etc. System boot files are called anything the user wishes to name them, with the restriction that they start with SYS. Some typical system boot file names are: SYSTEM_BA6, SYSB6_INSTALL, SYS_CUSTOM, etc.

The HP BASIC/WS client user can load a minimal system boot file and then rely on autostart files to dynamically load the desired binaries after the system boots up. The autostart files contain BASIC programs that do the `LOAD BIN` commands after the core system boot file has started. There is a general autostart file, named AUTOST, and node-specific autostart files, named AUTOSTnn, where nn is the two digit client node address (e.g., AUTOST23).

The HP BASIC/WS client will need to install the system boot files and autostart files on the SRM/UX file system under the Volume 8 (as specified in /etc/srmdconf) directory /SYSTEMS.

Only an HP BASIC/WS client system can create an HP BASIC/WS system boot file (using LOAD BIN and STORE SYSTEM commands); it can not be done from the server.

About Pascal Workstation System Clients

The Pascal Workstation System is a self-contained operating system and language system that runs on HP computers, and provides a Pascal language programming environment for various project applications.

The Pascal workstation system contains a "core" system boot file which is augmented by various programs, tables, and drivers. Usually the system boot file is named SYSTEM_P.

The three key programs in the Pascal workstation system for extending the booting environment are: TABLE, STARTUP, and INITLIB.

The Pascal workstation system clients also have autostart files that can be used to custom-configure individual client system environments.

The client user will need to install the system boot file, configuration programs, and autostart files under the SRM/UX file system Volume 8 directory, under the /WORKSTATIONS/SYSTEM, /WORKSTATIONS/SYSTEMnn, and/or /WORKSTATIONS/SYSTEMnnnnnn subdirectories (where nn is the two-character client node address on the SRM connection and nnnnnn is the client's six-character emulated node address on the LAN connection).

The Procedures

Establishing the Booting Environment

The procedure used to establish the booting environment is done once, before the SRM/UX system is ready to be used for day-to-day operations. It is done on the server and on one client of each type, i.e., HP BASIC/WS and Pascal Workstation System. It is performed either entirely by the SRM/UX System Administrator—if he or she is knowledgeable about the client language system—or by a team effort between the SRM/UX System Administrator and client users who are knowledgeable about their respective language systems.

You must have HP-UX superuser (root) privilege to perform the procedures from the server side.

Note To boot from the SRM/UX server over the SRM connection, client workstations must have BootROM version 3.0 or later. The BootROM version number is displayed while the workstation is powering up or rebooting.

To boot from the SRM/UX server over the LAN connection, client workstations must have BootROM version 3.0 Revision B or later.

If no BootROM message is displayed, the workstation's BootROM is an earlier version. In that case the workstation will not boot from the SRM/UX server over the SRM or LAN connection and must boot from its own mass storage.

General Information on Booting Over the LAN Connection

Note The steps below assume that you have already loaded the SRM/UX installation tape and run *update*(1M), as described in chapter 4 of this manual.

With respect to booting over the LAN connection there are two modes of operation that the HP-UX computer, which is acting as your SRM/UX server, can be in: *standalone* and *localroot*. In the *standalone* case, the computer is not a member of a diskless cluster. In the *localroot* case, the computer is configured as a root server in a diskless cluster. You do not have to be concerned about which mode your HP-UX server is in. The SRM/UX installation process will install the correct files in the correct places automatically.

With respect to client booting on the server there are two files that the installation process manipulates: /etc/rbootd and /etc/boottab.

The file /etc/boottab is the bootstrap configuration file for the rbootd daemon. The SRM/UX installation process "looks" at your server's current file system as it installs the boottab file in the following way:

- First, *update*(1M) looks in the /etc directory. If there is an /etc/boottab file, then *update*(1M) modifies it by installing the line:

  ```
  paws-srm:HPS300: :SYSB6_INSTALL,SYSTEM_BA6,SYSTEM_P:
  /AUTOST,/RMBBINS/SRM,/RMBBINS/LAN,INITLIB,TABLE,STARTUP
  ```

  The above `paws-srm` *line is one line in the file but appears as two lines here.*

- If there is no boottab file in /etc, then *update*(1M) looks in /etc/newconfig for a boottab file. If /etc/newconfig/boottab (which is the current HP-UX version of boottab) exists, then *update*(1M) copies it to /etc and modifies it by installing the `paws-srm` line above.
- If there is no /etc/newconfig/boottab file, then the /etc/newconfig/srmux/boottab file (the HP-UX version 7.0 boottab) is copied to /etc and modified by installing the `paws-srm` line above.

Note Although the SRM/UX installation process modifies /etc/boottab on your file system into a default configuration for SRM/UX, you will have to modify the `paws-srm` line (and possibly the `basic-srm` line) in /etc/boottab when you create new, custom system boot files and/or new autostart files. Refer to the section titled "The Model On the LAN Connection" earlier in this chapter for details.

The file /etc/rbootd is the executable, HP-UX remote boot server. The SRM/UX installation process installs the correct rbootd file in the /etc directory as follows:

- If the server computer is running HP-UX version 7.0*x*, then *update*(1M) saves the current rbootd file as /etc/HPUX7_0*x*/rbootd and replaces the current rbootd with the correct version.
- If the server computer is running HP-UX 8.0 or later, then *update*(1M) does nothing.

Note You can check which version of /etc/rbootd is running on the server by executing `what /etc/rbootd` from the server's keyboard. It will return:

```
/etc/rbootd:
         $Revision: 66.35 $
```

Revision 66.35 or later will work with SRM/UX.

After the installation process, the SRM/UX System Administrator needs only to perform the following steps to allow clients to boot from the server (in addition to the other steps detailed in chapter 4 which are indirectly related to client booting, such as reconfiguring the HP-UX kernel, modifying /etc/srmdconf, and starting /etc/srmd):

- Start /etc/srmd if it is not already running. Execute:

  ```
  /etc/srmd
  ```

- Start /etc/rbootd if it is not already running. Execute:

  ```
  /etc/rbootd
  ```

  □ If rbootd is already running, then kill the process:

  ```
  ps -ef | grep rbootd
  ```

  Note the PID (xxxx) for rbootd. Then,

  ```
  kill xxxx
  ```

  And start the daemon process. Execute:

  ```
  /etc/rbootd
  ```

Autostart File Search Rules. SRM/UX uses a set of rules when searching for autostart files. HP BASIC/WS clients have access to a node specific autostart file in the same way as provided by the SRM; that is, the autostart file is first looked for as /SYSTEMS/AUTOSTnn on Volume 8, where nn is the 2 digit emulated SRM node number of the client (see chapter 4 for a discussion of emulated SRM node numbers).

If /SYSTEMS/AUTOSTnn on Volume 8 does not exist, then the name /AUTOST is tried, where /AUTOST is a general autostart file for all HP BASIC/WS client nodes.

Note that the paths /SYSTEMS/AUTOSTnn and /AUTOST are relative to the SRM/UX Root Directory specified for Volume 8 in /etc/srmdconf. So the path /SYSTEMS/AUTOST12 as seen by the HP BASIC/WS client is actually /srmuxroot/SYSTEMS/AUTOST12 on the server's SRM/UX file system, when Volume 8 is assigned to /srmuxroot.

For the Pascal Workstation System client, the AUTOSTART file is searched for in the same directory in which the extension boot files were found, i.e., either /WORKSTATIONS/SYSTEMnn, /WORKSTATIONS/SYSTEMnnnnnn, or /WORKSTATIONS/SYSTEM.

General Information on Booting Over the SRM Connection

For clients to automatically boot in unattended mode, the SRM/UX system administrator must ensure that the desired system boot file is the first one found in the server's Volume 8 /SYSTEMS directory.

The concept of a file being the "first one" in a directory needs some explaining. HP-UX builds directories with "slots" for files, and these slots have an order. You can see the order of files in an HP-UX directory with the HP-UX `ls -f` command.

The order of the slots is not maintained in a simple fashion, it changes as files are removed and created. Here we want to ensure that the correct system boot file is in the first slot. One way to do this is the following. From the server:

1. create a new directory:

   ```
   mkdir /srmuxroot/SYS_NEW
   ```

2. copy or move the files into the new directory in the order you want them to appear:

   ```
   cp /srmuxroot/SYSTEMS/SYSTEM_BA6 /srmuxroot/SYS_NEW
   ```

   `cp` *source_file_name destination_file_name*, etc.

   OR

   ```
   mv /srmuxroot/SYSTEMS/SYSTEM_BA6 /srmuxroot/SYS_NEW
   ```

`mv` *source_file_name destination_file_name*, etc.

3. remove the old directory and rename the new directory to /srmuxroot/SYSTEMS:

   ```
   rm -r /srmuxroot/SYSTEMS
   ```

   Then:

   ```
   mv /srmuxroot/SYS_NEW /srmuxroot/SYSTEMS
   ```

Booting the First HP BASIC/WS Client Over LAN

This procedure is separated into tasks on the server and tasks on the client. Do the server tasks first.

Note During the SRM/UX installation process, the following system boot files and binaries were installed on your SRM/UX server:

- In /SYSTEMS under Volume 8:
  - □ SYSTEM_BA6—Minimal BASIC system boot file
  - □ SYSB6_INSTALL—BASIC system boot file used for installing 6.1
- In /RMBBINS under Volume 8:
  - □ LAN—LAN interface card driver
  - □ SCSI—SCSI interface card driver
  - □ SRM—Shared Resource Manager driver

These are the files which changed between the 6.0 and 6.1 releases of HP BASIC/WS.

Server Tasks. From the server workstation, do the following:

1. Bring up the server according to the procedures in chapter 4 of this manual. Ensure that:

   a. the /etc/srmdconf entries are correctly made (use /etc/srmdconfcheck to verify the correctness of /etc/srmdconf)

   b. /etc/services is modified to handle LAN interfaces

   c. /etc/srmd (the server process) is running

   d. /etc/rbootd is running

2. Ensure that the `paws-srm` line appears correctly in /etc/boottab.

   That is, ensure that this line is in /etc/boottab:

   ```
   paws-srm:HPS300: :SYSB6_INSTALL,SYSTEM_BA6,SYSTEM_P:
   /AUTOST,/RMBBINS/SRM,/RMBBINS/LAN,INITLIB,TABLE,STARTUP
   ```

   The printed output appears here on multiple lines, but there is only 1 line in the actual /etc/boottab file

   Refer to the "The Model on the LAN Connection" section earlier in this chapter for an explanation of the `SYSB6_INSTALL:/AUTOST,/RMBBINS/SRM,/RMBBINS/LAN` entries in this line.

3. Get the correct version of /etc/rbootd running (Use the instructions in the *General Information on Booting Over LAN* section above.)

Client Tasks.

Note The following is the HP BASIC/WS version 6.1 installation process. It does NOT apply to earlier or later versions of HP BASIC/WS.

After the SRM/UX System Administrator has done the above, do the following from one of the HP BASIC/WS client workstations:

1. SYSBOOT or cycle the computer's power.
2. Boot the SYSB6_INSTALL system boot file from the server.
3. Copy the binaries from your local mass storage onto the server's SRM/UX file system. To do this:

   a. Insert the version 6.0 BINARIES ONE disc into the client's flexible disc drive.

   b. MSI to the directory where the binaries are: e.g.:

   ```
   MSI ":,701,0"
   ```

   c. Turn wildcards on:

```
WILDCARDS UX; ESCAPE ""
```

d. Copy the binaries onto the server's file system, e.g., For a server at node address 0 and select code 21:

```
COPY "*" TO "/RMBBINS:REMOTE 21, 0;LABEL SRMUXROOT"
```

If your 6.0 discs are single-sided (option 042 or 044), then remove the BINARIES ONE disc from the drive and insert the BINARIES TWO disc. Then re-execute the COPY statement above.

Note Do NOT specify PURGE. This will ensure that you don't overwrite the binaries that the SRM/UX installation process already installed in /RMBBINS under Volume 8. Since you do not specify PURGE, you will see ERROR 293 reported after the COPY command is issued. This error is expected.

The binaries that are already installed are: SRM, LAN, and SCSI. They are the binaries that differ from version 6.0 to version 6.1.

---

4. Change the permissions on the binaries that you just copied. To do this:

a. MSI back to the SRM/UX directory. For example, execute:

```
MSI "/RMBBINS:REMOTE 21, 0;LABEL SRMUXROOT"
```

b. Change the permissions:

```
PERMIT "*";OWNER:READ; GROUP:READ; OTHER:READ
```

Since some of the binaries in this directory were installed during the SRM/UX installation process, you will see Error 293. This error is expected.

5. SYSBOOT to the minimal system, SYSTEM_BA6. To do this execute:

```
SYSBOOT "SYSTEM_BA6:REMOTE 21, 0;LABEL SRMUXROOT"
```

6. Use LOAD BIN commands to build the new system boot file, for example:

a. `LOAD BIN "/RMBBINS/SRM"`

b. `LOAD BIN "/RMBBINS/LAN"`

c. `LOAD BIN "/RMBBINS/IO"`

d. `LOAD BIN "/RMBBINS/GRAPH"`

e. `LOAD BIN "/RMBBINS/TRANS"`

f. *etc.*

You must load the SRM binary *FIRST* followed by the LAN binary.

The client users on your SRM/UX system may have different needs for binaries, so you should load and store as many custom system boot files as is necessary to provide the most efficient client environment. (See chapter 6 for more details on creating custom system boot files.)

HP recommends, at a minimum, storing the SRM and LAN binaries with each of your custom system boot files to be used on the LAN connection. Doing so will allow your client workstation to communicate directly with the SRM/UX server, rather than requiring it to communicate through the Boot ROM.

7. Store the custom system boot files in Volume 8 /SYSTEMS

   Use `STORE SYSTEM` commands, for example:

   a. `STORE SYSTEM "/SYSTEMS/SYSB6_JOE"`

   b. `STORE SYSTEM "/SYSTEMS/SYSB6_TEST3"`

   c. `STORE SYSTEM "/SYSTEMS/SYSB6_DESIGN1"`

   d. etc.

8. Have your SRM/UX System Administrator add the new system boot file names that you have created to the /etc/boottab file.

Verifying the Custom HP BASIC/WS System. —*LAN connection*

Client Task

To verify that you have performed the HP BASIC/WS client booting configuration correctly, execute a CAT command and a LIST BIN command. The CAT command should give you a listing of the files in the directory to which you have MSI'ed and the header should say "SRM/UX". The LIST BIN command should give you an accurate list of the language extension and driver binary files that the recently-stored system file contains. If the CAT and LIST BIN commands don't give you the results you expect, carefully repeat the above procedures. If you still have problems, refer to Appendix B of this manual for troubleshooting hints.

Booting the First HP BASIC/WS Client Over SRM

This procedure is separated into tasks on the server and tasks on the client. Do the server tasks first.

Note During the SRM/UX installation process, the following system boot files and binaries were installed on your SRM/UX server:

- In /SYSTEMS under Volume 8:
  - □ SYSTEM_BA6—Minimal BASIC system boot file
  - □ SYSB6_INSTALL—BASIC system boot file used for installing 6.1
- In /RMBBINS under Volume 8:
  - □ LAN—LAN interface card driver
  - □ SCSI—SCSI interface card driver
  - □ SRM—Shared Resource Manager driver

These are the files which changed between the 6.0 and 6.1 releases of HP BASIC/WS.

Server Tasks. From the server workstation, do the following:

1. Bring up the server according to the procedures in chapter 4 of this manual. Ensure that:

   a. the /etc/srmdconf entries are correctly made (use /etc/srmdconfcheck to verify the correctness of /etc/srmdconf)

   b. /etc/srmd (the server process) is running

Client Tasks.

Note The following is the HP BASIC/WS version 6.1 installation process. It does NOT apply to earlier or later versions of HP BASIC/WS.

After the SRM/UX System Administrator has done the above, do the following from one of the HP BASIC/WS client workstations:

1. SYSBOOT or cycle the computer's power
2. Boot the SYSB6_INSTALL system boot file from the SRM/UX server
3. Copy the binaries from your local mass storage onto the server's SRM/UX file system. To do this:
   a. Insert the version 6.0 BINARIES ONE disc into the client's flexible disc drive
   b. MSI to the directory where the binaries are: e.g.:

      ```
      MSI ":,701,0"
      ```

   c. Turn wildcards on:

      ```
      WILDCARDS UX; ESCAPE ""
      ```

   d. Copy the binaries onto the server's file system, e.g. For a server at node address 0 and select code 23:

      ```
      COPY "*" TO "/RMBBINS:REMOTE 23, 0;LABEL SRMUXROOT"
      ```

      If your 6.0 discs are single-sided (option 042 or 044), then remove the BINARIES ONE disc from the drive and insert the BINARIES TWO disc. Then re-execute the COPY statement above.

Note Do NOT specify PURGE. This will ensure that you don't overwrite the binaries that the SRM/UX installation process already installed in /RMBBINS under Volume 8. Since you do not specify PURGE, you will see ERROR 293 reported after the COPY command is executed. This error is expected.

The binaries that are already installed are: SRM, LAN, and SCSI. They are the binaries that differ from version 6.0 to version 6.1.

4. Change the permissions on the binaries that you just copied. To do this:
   a. MSI back to the SRM/UX root directory. For example, execute:

      ```
      MSI "/RMBBINS:REMOTE 23, 0;LABEL SRMUXROOT"
      ```

b. Change the permissions. For example, execute:

```
PERMIT "*";OWNER:READ; GROUP:READ; OTHER:READ
```

Since some of the binaries in this directory were installed during the SRM/UX installation process, you will see Error 293. This error is expected.

5. SYSBOOT to the minimal system, SYSTEM_BA6. To do this execute:

```
SYSBOOT "SYSTEM_BA6:REMOTE 23, 0;LABEL SRMUXROOT"
```

6. Use load bin commands to build the new system boot file, for example:

   a. `LOAD BIN "/RMBBINS/SRM"`

   b. `LOAD BIN "/RMBBINS/DCOMM"`

   c. `LOAD BIN "/RMBBINS/IO"`

   d. `LOAD BIN "/RMBBINS/GRAPH"`

   e. `LOAD BIN "/RMBBINS/TRANS"`

   f. *etc.*

   You must load the SRM binary *FIRST* followed by the DCOMM binary.

   The client users on your SRM/UX system may have different needs for binaries, so you should load and store as many custom system boot files as is necessary to provide the most efficient client environment. (See chapter 6 for more details on creating custom system boot files.)

   HP recommends, at a minimum, storing the SRM and DCOMM binaries with each of your custom system boot files to be used on the SRM connection. Doing so will allow your client workstation to communicate directly with the SRM/UX server, rather than requiring it to communicate through the Boot ROM.

7. Store the custom system boot files in Volume 8 /SYSTEMS

   Use `STORE SYSTEM` commands, for example:

   a. `STORE SYSTEM "/SYSTEMS/SYSB6_JOE"`

   b. `STORE SYSTEM "/SYSTEMS/SYSB6_TEST3"`

   c. `STORE SYSTEM "/SYSTEMS/SYSB6_DESIGN1"`

d. etc.

Verifying the Custom HP BASIC/WS System. *—SRM connection*

Client Task

To verify that you have performed the HP BASIC/WS client booting configuration correctly, execute a CAT command and a LIST BIN command. The CAT command should give you a listing of the files in the directory to which you have MSI'ed and the header should say "SRM/UX". The LIST BIN command should give you an accurate list of the language extension and driver binary files that the recently-stored system file contains. If the CAT and LIST BIN commands don't give you the results you expect, carefully repeat the above procedures. If you still have problems, refer to Appendix B of this manual for troubleshooting hints.

Booting the First Pascal Workstation System Client Over LAN

This procedure is separated into tasks on the server and tasks on the client. Do the server tasks first.

Server Tasks. From the server workstation, do the following:

1. Bring up the server according to the procedures in chapter 4 of this manual. Ensure that:

   a. the /etc/srmdconf entries are correctly made (use /etc/srmdconfcheck to verify the correctness of /etc/srmdconf)

   b. /etc/services is modified to handle LAN interfaces

   c. /etc/srmd (the server process) is running

   d. the directory /WORKSTATIONS/SYSTEM (relative to the root of volume 8 as specified in /etc/srmdconf) has been created

   e. the directory /SYSTEMS (relative to the root of volume 8 as specified in /etc/srmdconf) has been created

2. Make sure that the `paws-srm` line in /etc/boottab contains the appropriate Pascal client entries. If the server allows booting for Pascal clients only, ensure that the following line is in /etc/boottab:

   ```
   paws-srm:HPS300::SYSTEM_P:INITLIB,STARTUP,TABLE
   ```

If the server allows booting for both HP BASIC/WS clients and Pascal clients, see the discussion in the "Both Clients - The General Case" section earlier in this chapter for more on what a correctly formed entry in /etc/boottab looks like.

3. Get the correct version of /etc/rbootd running (Use the instructions in the "General Information on Booting Over LAN" section above).

Client Tasks. After the SRM/UX System Administrator has done the above, do the following from one of the Pascal Workstation System clients:

1. Boot your SRM/UX-capable version of Pascal, either from local mass storage (not from an SRM) or from flexible discs purchased as a separate product i.e., they are not supplied on the SRM/UX tape.
2. If you booted from local mass storage, and the DATA_COMM, IOMPX, LANDVR, and SRM modules were present in your INITLIB file, then skip steps 3 and 4 and go to step 5.
3. Execute (in order) the files DATA_COMM, LAN, and SRM. The modules IOMPX and LANDVR are present in LAN, the module SRM is present in SRM, and the module DATA_COMM is present in DATA_COMM.
4. Re-run the TABLE program that first executed when you booted.
5. Execute the FILER and request a volume listing (press V).

   The volume label corresponding to the root of volume 8 as specified in /etc/srmdconf should appear at unit #5 and the /WORKSTATIONS/SYSTEM (or /WORKSTATIONS/SYSTEMnnnnnn, where nnnnnn is the last 6 hex digits of the client's LAN Link Level Address) directory should appear at unit #45. If this does not happen contact your SRM/UX System Administrator for help.

   With the correct server volumes on line you are ready to copy the SRM/UX-capable Pascal Workstation System from either local mass storage or flexible discs to the server's SRM/UX file system.

6. If you booted from local mass storage, you already have the entire Pascal system in a particular directory on this mass storage device. Use the FILER to copy the entire contents of this directory to the /WORKSTATIONS/SYSTEM directory (relative to the root of volume 8 as specified in /etc/srmdconf) on the server.

If /WORKSTATIONS/SYSTEMnnnnnn is the directory corresponding to the on line volume at unit #45, you should prefix unit #45 to the volume /WORKSTATIONS/SYSTEM instead, since the purpose of this procedure is to build a generic system from which other systems can be constructed.

Execute the FILER, and prefix to #45:../SYSTEM.

If you booted from flexible discs, you should copy the Pascal Workstation System to the /WORKSTATIONS/SYSTEM directory. Copy the system one floppy at a time.

Note

Since you have a Series 300 workstation (booting over the LAN is not possible with a Series 200), you want the INITLIB file from the BOOT2: flexible disc. However, it is a good idea to make copies in the SYSTEM directory of a correct INITLIB for both the Series 200 computers (copy INITLIB from BOOT: to the SYSTEM directory and call it INIT_200) and the Series 300 computers (copy INITLIB from BOOT2: and call it INIT_300). Individual users may then use the correct INITLIB.

Make sure the modules DATA_COMM and SRM are present in INIT_200 and that the modules DATA_COMM, IOMPX, LANDVR, and SRM are present in INIT_300. Use the LIBRARIAN to add them if necessary (see chapter 7 for details on how to do so).

---

7. Copy the SYSTEM_P file to the /SYSTEMS directory (again, relative to the root of volume 8 as specified in /etc/srmdconf).

Verifying the Pascal Workstation System Booting Environment. *—LAN connection*

Client Tasks

At this point a minimal, bootable system is in place. To verify this :

1. Cycle power on your workstation (or, if the Debugger is loaded, enter the Debugger with CTRL-RESET and enter the sb (lower case) command).
2. Hit the space bar, to request attended boot mode, after the Boot ROM indicates that the keyboard has been found. (You must do this before the

Boot ROM begins searching for system boot files or it will boot the *first* system boot file it finds, that is, it will enter the unattended boot mode.)

3. The Boot ROM should communicate with /etc/rbootd on the server and find SYSTEM_P as a bootable system.
4. Enter the appropriate number(s) and letter (e.g. 1P) to try booting from your newly-created boot environment on the server.

   If everything is set up correctly, the files SYSTEM_P, INITLIB, STARTUP, and TABLE will all be loaded and the AUTOSTART file present in /WORKSTATIONS/SYSTEM will be streamed. The volume present at unit #5 after booting should correspond to the root of volume 8 as specified in /etc/srmdconf and the volume at unit #45 should correspond to /WORKSTATIONS/SYSTEM or /WORKSTATIONS/SYSTEMnnnnnn.

Booting the First Pascal Workstation System Client Over SRM

This procedure is separated into tasks on the server and tasks on the client. Do the server tasks first.

Server Tasks. From the server workstation, do the following :

1. Bring up the server according to the procedures in chapter 4 of this manual. Ensure that :
   a. the /etc/srmdconf entries are correctly made (use /etc/srmdconfcheck to verify the correctness of /etc/srmdconf)
   b. /etc/srmd (the server process) is running
   c. the directory /WORKSTATIONS/SYSTEM (relative to the root of volume 8 as specified in /etc/srmdconf) has been created
   d. the directory /SYSTEMS (relative to the root of volume 8 as specified in /etc/srmdconf) has been created

Client Tasks. After the SRM/UX System Administrator has done the above, do the following from one of the Pascal Workstation System clients:

1. Boot your SRM/UX-capable version of Pascal from either local mass storage (not SRM) or from flexible discs.

Note During this process your client workstation should not be attached to an SRM server or the Pascal Client SRM driver will get confused.

2. If you booted from local mass storage and the DATA_COMM and SRM modules were present in your INITLIB file, skip steps 3 and 4 and go to step 5.
3. Execute (in order) the files DATA_COMM and SRM.
4. Re-run the TABLE program that executed when you booted.
5. Execute the FILER and request a volume listing (press V).

   The volume label corresponding to the root of volume 8, as specified in /etc/srmdconf, should appear at unit #5 and the /WORKSTATIONS/SYSTEM or /WORKSTATIONS/SYSTEMnn directory should appear at unit #45. If this does not happen contact your SRM/UX System Administrator for help.

   With the correct server volumes on-line you are ready to copy the SRM/UX-capable Pascal Workstation System from either local mass storage or flexible discs to the server's SRM/UX file system.
6. If you booted from local mass storage, you already have the entire Pascal Workstation System in a particular directory on this mass storage device. Use the FILER to copy the entire contents of this directory to the /WORKSTATIONS/SYSTEM directory (relative to the root of volume 8 as specified in /etc/srmdconf) on the server. If the volume at unit #45 corresponds to /WORKSTATIONS/SYSTEMnn, enter the FILER and prefix to `#45:../SYSTEM`.

   If you booted from flexible discs, you will have to copy the Pascal Workstation System to the /WORKSTATIONS/SYSTEM directory. Copy the system one flexible disc at a time.

Note Copy the INITLIB file on the BOOT: flexible disc to the INIT_200 file and copy the INITLIB file on the BOOT2: flexible disc to the INIT_300 file. These are the correct INITLIB files for the Series 200 and Series 300 workstations, respectively.

Make sure that INIT_200 and INIT_300 each contain the modules DATA_COMM and SRM. Use the LIBRARIAN to add them if necessary.

If the workstation you are using to establish the booting environment on the server is a Series 200 computer, then "Dup-Link" INIT_200 to INITLIB (in the /WORKSTATIONS/SYSTEM directory). If it is a Series 300 computer, then "Dup-Link" INIT_300 to INITLIB in this directory.

7. Copy the SYSTEM_P file to the /SYSTEMS directory (again, relative to the root of Volume 8 which is specified in /etc/srmdconf).

Verifying the Pascal Workstation System Booting Environment. *—SRM connection*

Client Tasks

At this point a minimal, bootable system is in place. To verify this :

1. Cycle power on your workstation (or, if the Debugger is loaded, enter the Debugger with CTRL-RESET and enter the sb (lower case) command).
2. Hit the space bar, to request the attended boot mode, after the Boot ROM indicates the keyboard has been found. (You must do this before the Boot ROM begins searching for system boot files or it will boot the first system boot file it finds, that is, it will enter the unattended boot mode.)
3. The Boot ROM should should find SYSTEM_P as a bootable system on the server.
4. Enter the appropriate number(s) and letter (e.g. 1P) to try booting from your newly-created boot environment on the server.

   If everything is set up correctly, the files SYSTEM_P, INITLIB, STARTUP, and TABLE will all be loaded and the AUTOSTART file present in /WORKSTATIONS/SYSTEM will be streamed. The volume present at unit #5 after booting should correspond to the root of volume 8 as specified in /etc/srmdconf and the volume at unit #45 should correspond to /WORKSTATIONS/SYSTEM.

Adding Clients to the Booting Environment

Once you have installed the booting environment with one client of each type (HP BASIC/WS and Pascal Workstation System) on the server, you will need to complete the system by adding the rest of the clients.

General Recommendations

Organizational Hints

- The SRM/UX installation process has installed the directory `/srmuxroot/RMBBINS` on your SRM/UX file system. This directory contains all of the HP BASIC/WS binaries that have been modified since version 6.0 (the modified binaries are SRM, LAN, and SCSI). We recommend copying all of the rest of the HP BASIC/WS (version 6.0) binaries from your local flexible discs into this directory on the server's file system. Any of the binaries that are not shipped on the SRM/UX product tape can be used with SRM/UX from the HP BASIC/WS flexible discs.
- We recommend that you set up some kind of client user tree under the Volume 8 root directory that will provide a set of working directories for the users after they boot (e.g., /srmuxroot/users). For example, if you want your system to be organized according to client workstation names, you could set up client name directories; for example, `/srmuxroot/users/design1`, `/srmuxroot/users/test23`, etc. If you want your system to be organized according to client user names, you could set up user name directories; for example, `/srmuxroot/users/joe_smith`, `/srmuxroot/users/mary_jones`, etc.

Using Autostart Files

You can create autostart files for both HP BASIC/WS and Pascal Workstation System clients. One of the main functions of autostart files is to dynamically load a custom system on a client workstation. That is, in HP BASIC/WS you can boot a minimal system boot file (for example, SYSTEM_BA6) and then either manually perform the LOAD BIN and STORE SYSTEM commands to create a custom system or write an autostart program that will dynamically and automatically perform the LOAD BIN commands to create a custom system. Every time you boot the client from then on, the autostart file will load the custom system.

You can create both general and node-specific autostart files. Some of the clients may use the same set of binaries as other clients. In that case you can write a general autostart file that performs the LOAD BIN commands that create the custom HP BASIC/WS system.

Many client workstations are used for unique functions and, therefore, have special requirements that can be handled by creating unique, node-specific autostart files. The node-specific autostart file can perform the LOAD BIN commands that create the custom HP BASIC/WS system for each client's unique needs.

After the HP BASIC/WS system takes control from the Boot ROM, it follows a search path for autostart files in the following sequence: first, it locates its node-specific autostart file in the /srmuxroot/SYSTEMS directory under Volume 8. If the node-specific file is not found, then it looks for the general autostart file, /AUTOST under the Volume 8 root. Then, if no autostart files are found, the HP BASIC/WS system remains in the state that the system boot file established, and it turns the screen over to the user.

Note If you are booting over the LAN connection, you can use node-specific autostart files ONLY if you boot a system that contains the SRM and LAN binaries.

The advantage of autostart files to load the binaries is that they are quite small and, therefore, take up much less disc space than customized system boot files. The disadvantage of autostart files is the time required during boot-up and system-configuration to load all of the required binaries.

See your HP BASIC/WS and Pascal Workstation System manuals for details on how to write autostart files.

The Procedures For Adding Clients

Edit /etc/srmdconf

The SRM/UX System Administrator is responsible for editing the /etc/srmdconf configuration file, therefore all of the instructions below are addressed to the Administrator.

- You must give each client in the SRM/UX system access to Volume 8 (the SRM/UX root directory) if the client will boot from the server. To do this, enter the Volume 8 volume name (SRMUXROOT as shipped on the tape) into each client's volume list in the LAN-CLIENTS or SRM-CLIENTS table.
- Each interface in each client workstation must have its own line in either the LAN-CLIENTS table or in the SRM-CLIENTS table.
- Fill in the entries in all of the applicable fields in srmdconf so that the SRM/UX file system you planned in chapters 2 and 4 of this manual will be correctly utilized by the clients.

See chapter 4 of this manual for the complete discussion on editing /etc/srmdconf.

Edit /etc/boottab For LAN Clients

The SRM/UX System Administrator is responsible for editing the /etc/boottab file, therefore all of the instructions below are addressed to the Administrator.

You must ensure that the `paws-srm` line (and possibly the `basic-srm` line) in /etc/boottab contains the proper entries. You should add all of the system boot file names that you and the client users have built (for both HP BASIC/WS and Pascal Workstation System), and that will be booted from the server, to the fourth field in the `paws-srm` line.

For the Pascal Workstation System clients:

- you should add all of the custom INITLIB, STARTUP, and TABLE program names to the fifth field in the `paws-srm` line; full path names are not recommended.

See the discussion in:

- "The Model On the LAN Connection" section earlier in this chapter for details on using the recommended method for editing the /etc/boottab file
- "The Procedures" section earlier in this chapter, to follow the procedures for modifying /etc/boottab for the first clients. Then you can extend the procedures to fit all of the clients

Autostart Files

HP BASIC/WS on the SRM Connection. To create a *general* autostart file, /AUTOST under the Volume 8 root, which contains a program that executes the LOAD BIN commands that will create the most applicable system for your client users, follow this procedure:

- boot an HP BASIC/WS system boot file (e.g., SYSTEM_BA6, SYSB6_INSTALL, etc.), in attended mode
- load the SRM and DCOMM binaries (if they are not in the system boot file). Execute:
  - □ `LOAD BIN "/RMBBINS/SRM"`
  - □ `LOAD BIN "/RMBBINS/DCOMM"`
- load the EDIT binary (if it's not in the boot file) with:

  ```
  LOAD BIN "/RMBBINS/EDIT"
  ```
- write the program that loads the desired binaries and store it in a file named /AUTOST and run the file to verify proper operation
- execute a SYSBOOT or cycle the client workstation's power
- again, boot an HP BASIC/WS system boot file, in attended mode
- since no node-specific autostart-program file is found, the system boot program looks for and finds your Volume 8 /AUTOST file and the program therein dynamically loads the extended system
- if no autostart-program files are found, the client is left with the configuration established by the system boot file

To create a *node-specific* autostart file, for example /SYSTEMS/AUTOST42, under Volume 8 root, which contains a program that executes the LOAD BIN commands that will create the custom system for the client at node address 42, follow this procedure:

- boot an HP BASIC/WS system boot file (e.g., SYSTEM_BA6, SYSB6_INSTALL, etc.), in attended mode
- load the SRM and DCOMM binaries (if they are not in the system boot file). Execute:
  - □ `LOAD BIN "/RMBBINS/SRM"`
  - □ `LOAD BIN "/RMBBINS/DCOMM"`
- load the EDIT binary (if it's not in the boot file) with:

  ```
  LOAD BIN "/RMBBINS/EDIT"
  ```
- write the program that loads the desired binaries and store it in a file named /SYSTEMS/AUTOST42 and run it to verify proper operation
- execute a SYSBOOT or cycle the client workstation's power
- again, boot the HP BASIC/WS system boot file, in attended mode
- if the system boot program finds the node-specific autostart file, /AUTOST42, the program therein dynamically loads the extended system
- if the node-specific autostart-program file is not found, the system searches for a general autostart file /AUTOST. If the general file is not found the client is left with the configuration established by the system boot file.

HP BASIC/WS on the LAN Connection. Although the recommended method of booting HP BASIC/WS over LAN is to boot a custom system boot file that contains at least the SRM and LAN binaries, you could use a different method.

You could boot a minimal system boot file (e.g., SYSTEM_BA6), execute: LOAD BIN "/RMBBINS/SRM", LOAD BIN "/RMBBINS/LAN", LOAD BIN "/RMBBINS/EDIT", and write a general autostart file (/AUTOST) which—next time you boot this client—will load the SRM and LAN binaries and then query the LAN card for its emulated node address. It then proceeds to load the correct node-specific autostart file, which will leave your HP BASIC/WS system configured as you planned.

The general autostart program that will perform the above procedure is listed below:

```
10    ! RE-STORE "/AUTOST"
11    !  Suggested generic AUTOST program for deriving an
12    !  emulated node address of a LAN interface and loading the
13    !  appropriate node-specific autostart file
14    DIM Str$[80]
15    INTEGER Node
17    LOAD BIN "/RMBBINS/SRM"  !  MUST LOAD THIS ONE FIRST
19    LOAD BIN "/RMBBINS/LAN"
21    Str$=SYSTEM$("MSI")
29    STATUS VAL(Str$[9;2]),6;Node
31    ON ERROR GOTO Errortrap
35    LOAD "/SYSTEMS/AUTOST"&VAL$(Node)
39    ! if LOAD succeeds, will not reach here
41 Errortrap: !
42    OFF ERROR
43    IF ERRN<>56 THEN PRINT ERRM$
46    END
```

Pascal Workstation System Clients. The details of adding Pascal clients to the booting environment are provided in chapter 7 of this manual and are not repeated here.

Placing Autostart and System Boot Files in the File System

For HP BASIC/WS Clients. Ensure that all of the system boot files are in the /SYSTEMS directory under Volume 8 (Volume 8 is specified in the server's /etc/srmdconf file).

If there is one, ensure that the general autostart file, AUTOST, is under the Volume 8 root directory.

If there are any, ensure that the node-specific autostart files are in the /SYSTEMS directory under Volume 8.

For Pascal Workstation System Clients. Ensure that the system boot file, SYSTEM_P, is in the /SYSTEMS directory under Volume 8 (Volume 8 is specified in the server's /etc/srmdconf file).

If there is one, ensure that the general autostart file, AUTOSTART, is under the /WORKSTATIONS/SYSTEM directory under Volume 8.

If there are any, ensure that the node-specific autostart files (also called AUTOSTART) are under the node-specific directories /WORKSTATION/SYSTEMnn (for SRM) and /WORKSTATION/SYSTEMnnnnnn (for LAN), under Volume 8 (where nn and nnnnnn are the SRM client node address and LAN emulated client node address, respectively).

Automatically Booting a Preferred System

After you have booted and verified the first client of each type, and have added all of the remaining clients to the booting environment, there is one more level of customization available for your SRM/UX client booting environment: you can configure the server to automatically boot either the HP BASIC/WS or Pascal Workstation System. That is, you can configure the system so that each client automatically boots the language system that you predetermine is the one the users need.

The procedures to install this capability do not invalidate anything you have done up to this point.

The mechanisms that provide this capability are different on the SRM connection and LAN connection, as shown below.

Note If a client computer is configured not only as an SRM/UX client, but also as an HP-UX Cluster client (cnode) as well, then the client will boot HP-UX as its preferred system, if you use the unattended mode. (That is, automatically booting either HP BASIC/WS or Pascal Workstation System will not work if the client computer is also configured as an HP-UX cnode.)

To boot the client as an SRM/UX client requires the attended boot mode in this case.

On the LAN Connection

The mechanism that provides automatic booting of a preferred system over the LAN connection is an HP-UX-style Context Dependent File (CDF) structure for system boot files.

Hewlett-Packard has developed a structure called a **Context Dependent File** (CDF). A CDF is a mechanism for allowing different Cluster clients (cnodes) to see different contents for a file which has the same name for all clients. See the "HP-UX Clusters" chapter of the *HP-UX System Administration Concepts* manual for a complete discussion of HP-UX Context Dependent Files.

To install the capability for clients to automatically boot node-specific systems, you (the SRM/UX System Administrator) will create the CDF in /SYSTEMS under Volume 8 and will make changes to /etc/boottab.

All Clients Automatically Booting the Same System

If your SRM/UX system needs require all client workstations to run the same system, then follow the procedures below.

This is the simple case. All you need to do is create the system boot file you want all the clients to boot—using the procedures detailed above—and then place the system boot file's file name FIRST in the fourth field of the `paws-srm` line in /etc/boottab.

For example:

- If you have all HP BASIC/WS clients and you want them all to automatically boot SYSTEM_BA6, then your /etc/boottab line would look like this:

```
paws-srm:HPS300: :SYSTEM_BA6,SYSTEM_P:/AUTOST,INITLIB,STARTUP,TABLE
```

- If you have all Pascal Workstation System clients and you want them all to automatically boot SYSTEM_P, then your /etc/boottab line would look like this:

```
paws-srm:HPS300: :SYSTEM_P,SYSTEM_BA6:/AUTOST,INITLIB,STARTUP,TABLE
```

Every time you reboot the clients, and let them run unattended, they will boot the correct system boot file.

Automatically Booting Node-Specific Systems

If your SRM/UX system needs require client workstations to boot different systems, then follow the procedures below.

Create the CDF:

You will create a CDF with the name AUTOBOOT (you can use any name). The CDF AUTOBOOT will contain the actual system boot files you created for the clients. The CDF can contain both HP BASIC/WS and Pascal Workstation System boot files.

For example, let's say that you had an SRM/UX system with two HP BASIC/WS clients and one Pascal Workstation System client, all on the LAN connection. The first BASIC client will boot the SYSTEM_BA6 boot file. The second BASIC client will boot the SYSTEM_BA6_ALL boot file. The Pascal client will boot the SYSTEM_P boot file.

Create the CDF by executing the following commands from the server's HP-UX shell:

```
cd /srmuxroot/SYSTEMS #if /srmuxroot is your Volume 8 Root Directory

mkdir AUTOBOOT

chmod +H AUTOBOOT #convert AUTOBOOT to a CDF

cd AUTOBOOT+

ln -s /srmuxroot/SYSTEMS/SYSTEM_BA6 default

ln -s /srmuxroot/SYSTEMS/SYSTEM_BA6_ALL 0x0800090056A5

ln -s /srmuxroot/SYSTEMS/SYSTEM_P  0x08000900B69D
```

This example shows the three different "automatic" boot files. Notice how the system boot files are symbolically linked to the client's LAN interface as specified by its link level address. (See the ln(1) man page for details on symbolic links.) The link level address of the SRM/UX client is used as the "context" for the CDF, where "context" is an ASCII string made up of the following attributes: cnode name, fpoint hardware type, processor types, cnode type, and default.

HP BASIC/WS client 0x0800090056A5 will automatically boot the SYSTEM_BA6_ALL boot file. The Pascal Workstation System client 0x08000900B69D will automatically boot the SYSTEM_P boot file. All other HP BASIC/WS clients (in this case there is only one more BASIC client) will boot the default boot file, specified here as SYSTEM_BA6.

Note The form of the link level address given in the `ln` statement must match the format given in these examples exactly; that is, the leading 0x must be LOWER CASE (0X won't work), and other alpha characters in the link level address must be UPPER CASE (b69d won't work, B69D will). Each name must be exactly 14 characters long.

If you have problems with the CDF structure, check these things first.

You can list the structure of the example CDF as follows:

- *If you execute*:

  ```
  ll -d /srmuxroot/SYSTEMS/AUTOBOOT+
  ```

  The output might be:

  ```
  drwsr-xr-x   2 jones srmux  1024 Oct 12 13:50 /srmuxroot/SYSTEMS/AUTOBOOT+
  ```

- *If you execute*:

  ```
  ll /srmuxroot/SYSTEMS/AUTOBOOT+
  ```

  The output might be:

```
total 8
lrwxr-xr-x    1 jones   srmux    32 Oct 23 10:55 0x0800090056A5 ->/srmuxroot/SYSTEMS/SYSTEM_BA6
lrwxr-xr-x    1 jones   srmux    26 Oct 23 20:26 0x08000900B69D ->/srmuxroot/SYSTEMS/SYSTEM_P
lrwxr-xr-x    1 jones   srmux    28 Oct 12 13:50 default ->/srmuxroot/SYSTEMS/SYSTEM_BA6_ALL
```

Changes to /etc/boottab:

Once you have created the AUTOBOOT CDF, you simply add its name to the `paws-srm` line—as the first system boot file name in the fourth field—in the /etc/boottab file. (Again, the name AUTOBOOT is arbitrary; you can use whatever name you wish.)

Your `paws-srm` line in /etc/boottab might look like this (if you haven't changed it from above):

```
paws-srm:HPS300: :SYSTEM_BA6,SYSTEM_BA6_ALL,SYSTEM_P:
/AUTOST,INITLIB,STARTUP,TABLE
```

The printed output appears here on multiple lines, but there is only 1 line in the actual /etc/boottab file

To install the capability to automatically boot either HP BASIC/WS or Pascal Workstation System, install the name of the CDF into the first position in the fourth field of the paws-srm line; for example:

```
paws-srm:HPS300: :AUTOBOOT,SYSTEM_BA6,SYSTEM_BA6_ALL,SYSTEM_P:
/AUTOST,INITLIB,STARTUP,TABLE
```

The printed output appears here on multiple lines, but there is only 1 line in the actual /etc/boottab file

Note If the SRM/UX clients on the LAN connection will ALWAYS boot automatically, then you can leave SYSTEM_BA6, SYSTEM_BA6_ALL, and SYSTEM_P out of the paws-srm line.

However, if you need to boot some clients manually (i.e., with the attended boot mode), then leave these file names in the line so they will be displayed in the Boot ROM menu on the client's screen.

On the SRM Connection

All Clients Automatically Booting the Same System

If your SRM/UX system needs require all client workstations to run the same system, then follow the procedures below.

This is the simple case. All you need to do is create the system boot file you want all the clients to boot—using the procedures detailed above—and then place the system boot file FIRST in the /SYSTEMS directory under Volume 8 using the procedures detailed in the "General Information on Booting Over the SRM Connection" section above.

Automatically Booting Node-Specific Systems

If your SRM/UX system needs require client workstations to boot different systems, then follow the procedures below.

The mechanism that provides automatic booting of a preferred system over the SRM connection is the Loader Utility. The Loader Utility is the same one that was used with the SRM product.

The Loader Utility consists of four types of files: SYSTEM_LD, CONFIGER, CONFIG_LD, and CONFIG_LDnn. The SRM/UX installation process installs the SYSTEM_LD and CONFIGER files on your SRM/UX file system, in the /SYSTEMS directory under Volume 8 (/srmuxroot) as specified in the server's /etc/srmdconf file. You then create the CONFIG_LD and CONFIG_LDnn files with the CONFIGER program.

All Series 300 computers have Boot ROMs that are compatible with the Loader Utility. Series 200 computers must have a 3.0 or later Boot ROM (but not revision 3.0L).

How the Loader Utility Works. The Loader is an operating system that performs its function and then deletes itself, turning over control to the selected operating system (HP BASIC/WS or Pascal Workstation System).

The Loader Utility system boot file (SYSTEM_LD) must be the first system boot file in the /SYSTEMS directory under Volume 8. See the "General Information on Booting Over the SRM Connection" section above for instructions on how to make a system boot file the first file in a directory. The HP BASIC/WS and Pascal Workstation System boot files must also be in the /SYSTEMS directory under Volume 8 for automatic booting to work.

Creating a Default CONFIG_LD File. To create a default CONFIG_LD file on the SRM/UX server's file system, do the following from the client workstation:

1. MSI to the /SYSTEMS directory. Execute, for example:

   ```
   MSI "/SYSTEMS:REMOTE 23, 0;LABEL SRMUXROOT"
   ```

2. Load the CONFIGER program into memory. Execute:

   ```
   LOAD "CONFIGER"
   ```

3. Start the CONFIGER program. Press the Run key.
4. When the CONFIGER program displays the prompt:

   ```
   CONFIG FILE NAME
   ```

   Then:

Enter the required name for the default configuration file you are about to create. Type:

```
CONFIG_LD
```

5. When CONFIGER displays the prompt:

   ```
   ?
   ```

   Then:

   Enter the name of the Pascal Workstation System or HP BASIC/WS system boot file that you want to have load automatically for any workstation without its own configuration file. That is, enter the name of the system boot file that the majority of the clients will boot as a default.

   For example:

   ```
   SYSTEM_BA6
   ```

6. After you enter the system boot file above, and the CONFIGER program prompts again with:

   ```
   ?
   ```

   Then:

   Type (Return), to end the CONFIGER program.

   The ? prompt disappears and the CONFIGER program creates the default configuration file, CONFIG_LD, in /SYSTEMS.

Creating a Node-Specific CONFIG_LDnn File. To create a node-specific CONFIG_LDnn file on the SRM/UX server's file system, do the following from the client workstation:

1. MSI to the /SYSTEMS directory. Execute, for example:

   ```
   MSI "/SYSTEMS:REMOTE 23, 0;LABEL SRMUXROOT"
   ```

2. Load the CONFIGER program into memory. Execute:

   ```
   LOAD "CONFIGER"
   ```

3. Start the CONFIGER program. Press the (Run) key.
4. When the CONFIGER program displays the prompt:

```
CONFIG FILE NAME
```

Then:

Type `CONFIG_LDnn` Return, where nn is the SRM client's node address in the SRM/UX system.

For example, if the client's node address is 6, then type:

```
CONFIG_LD6 Return
```

5. When the CONFIGER program displays the prompt:

   ```
   ?
   ```

   Then:

   Type the name of the Pascal Workstation System or HP BASIC/WS system boot file that you want to boot automatically. For example:

   ```
   SYSTEM_P Return
   ```

6. When the CONFIGER program displays the prompt:

   ```
   ?
   ```

   Then:

   Type: Return, to end the CONFIGER program.

   The `?` prompt disappears and the CONFIGER program creates the node-specific configuration file, CONFIG_LDnn, in /SYSTEMS.

7. Repeat steps 3 through 6 as many times as is necessary to create a node-specific configuration file for every client that needs one.

6

Installing and Configuring HP BASIC/WS Clients

This chapter tells how to install the HP BASIC/WS system on any HP 9000 Series 200, or 300 computer so that it functions as a client in an SRM/UX system.

The SRM/UX server performs three major services for the clients: language-system booting, SRM/UX file system management, and client file-spooling management. Chapter 4 discusses these functions in detail from the server's perspective. This chapter looks at the three SRM/UX functions from the HP BASIC/WS client's perspective.

Before You Start

Before you begin the HP BASIC/WS system start-up:

- ensure all of the hardware components of your SRM/UX system are properly installed
- ensure that the SRM/UX System Administrator has properly installed, configured, and started all of the software components of the SRM/UX server
- ensure that the SRM/UX System Administrator has made the appropriate entries in the server's /etc/srmdconf file for your particular workstation. See chapter 4 of this manual for server configuration details and chapter 5 for information on client booting from the server.

What You Need for Start-up

- the media which holds the HP BASIC/WS system software
- the HP BASIC/WS system manual set, for more detailed information about the start-up options mentioned in this chapter

Booting the HP BASIC/WS Language System

For HP BASIC/WS 6.1, the SRM/UX System Administrator installs the system boot file and the LAN, SCSI, and SRM binaries on the server. The client user then should copy the rest of the binaries from the HP BASIC/WS 6.0 discs to the server. For later versions of HP BASIC/WS, the client user should copy the system boot file and all of the binaries to the server.

Once this is done, the client users can boot from the SRM/UX server rather than their local mass storage. This gives them the flexibility to boot all clients into the same system configuration, or into unique system configurations.

This chapter discusses booting HP BASIC/WS from local mass storage and creating a "custom" system boot file. Chapter 5 covers the full details of the HP BASIC/WS booting environment on the server.

From Your Workstation's Local Mass Storage

Note If you are installing HP BASIC/WS version 6.1, then skip this section.

For complete instructions on booting the BASIC system, refer to the manual: *Installing and Maintaining HP BASIC* for whichever version you are installing. There you will learn how to install the HP BASIC/WS system from the flexible discs shipped with the workstation.

After you install and boot the HP BASIC/WS system from the flexible discs, decide which language extension and driver binary files you need to create your "extended" or "customized" HP BASIC/WS system. Once you have created this system, store it in a directory (/SYSTEMS under Volume 8) on the SRM/UX file system. From then on, the client users can boot from the server.

Note If you want to boot this system over a LAN connection, then you should ask your SRM/UX System Administrator to add the name of your system boot file to the `paws-srm` line in the server's /etc/boottab file.

Creating and Storing an Extended HP BASIC/WS Language System

To understand the process of booting an HP BASIC/WS client from the server, read chapter 5. There you will see that the procedures for booting HP BASIC/WS from the server are different for SRM and LAN connections, but only from the server's side. From the client's side, the procedure is the same. This section tells you how to create "customized" or "extended" system boot files (i.e., an extended HP BASIC/WS language system) from the client workstation.

If you want your client to run under a custom HP BASIC/WS system, then you must create and store the custom system boot file from the client.

In order to create your custom system, you must boot the SYSTEM_BA6 system boot file (or another existing boot file), and then load the binaries you need from the /RMBBINS directory under Volume 8 on the server (assuming that you copied all of your HP BASIC/WS binaries from your local mass storage to /RMBBINS on the SRM/UX server). Finally, you will store the custom system in an appropriately named system boot file (e.g., SYS_CUSTOM1). See the procedures below.

To boot SYSTEM_BA6 from the server and then create another, custom boot file, do the following:

- Enter the attended-boot mode: repeatedly depress or hold down the (space) key after a SYSBOOT or after cycling the computer's power, until you get the Boot ROM menu of bootable system files to appear on the right-hand side of the screen.
- After you select the two or three character code for the system boot file you want (in this case SYSTEM_BA6), the HP BASIC/WS system comes on-line (approximately one minute later). You are placed in the Volume 8

root directory after you have booted. Typically that will be the **/srmuxroot** directory (unless your SRM/UX System Administrator has assigned it differently in the server's /etc/srmdconf VOLUME-TABLE).

- You then load the additional language extension and driver binaries you need using LOAD BIN commands, and execute the STORE SYSTEM command, which creates a new system boot file from which you can subsequently boot every time you, or any other HP BASIC/WS system user on the SRM/UX system, want to use this particular version of the customized HP BASIC/WS system.

  Be sure to execute the STORE SYSTEM command in the /SYSTEMS directory.

Note The binaries that are required for HP BASIC/WS to communicate with SRM/UX are SRM and DCOMM (on the SRM connection) or SRM and LAN (on the LAN connection). Before HP BASIC/WS can communicate with the SRM/UX server, these binaries must either be in the system you booted or you must load them (using the LOAD BIN command).

Note If you want to boot this system over a LAN connection, then you should ask your SRM/UX System Administrator to add the name of your system boot file to the **paws-srm** line in the server's /etc/boottab file.

Day-to-Day Booting From the Server

First, the SRM/UX System Administrator and HP BASIC/WS client users must establish the booting environment; that is, create the necessary system boot files and place them in /SYSTEMS under Volume 8 (as detailed above and in chapter 5). Then, the client users can boot any system boot file they wish any time they wish. To do so, follow the procedures below:

For Attended Booting:

1. Enter the attended-boot mode: repeatedly depress or hold down the (space) key after a SYSBOOT or after cycling the computer's power, until you get

the Boot ROM menu of bootable system files to appear on the right-hand side of the screen.

2. After selecting the two or three character code for the system boot file you want, the HP BASIC/WS system comes on-line (approximately one minute later). You are placed in the Volume 8 root directory after you have booted. Typically that will be the `/srmuxroot` directory (unless your SRM/UX System Administrator has assigned it differently in the server's /etc/srmdconf VOLUME-TABLE).

For Unattended Booting:

1. SYSBOOT or cycle the computer's power
2. Let the system run unattended if you are booting over the SRM connection. It will automatically select the first system boot file in the /SYSTEMS directory under Volume 8. If you are booting over the LAN connection, it will select the first system boot file in the `paws-srm` line in /etc/boottab (See your SRM/UX System Administrator for details).

   The HP BASIC/WS system comes on-line (approximately one minute later). You are placed in the Volume 8 root directory after you have booted. Typically that will be the /srmuxroot directory on the server's HP-UX file system (unless your SRM/UX System Administrator has assigned it differently in the server's /etc/srmdconf VOLUME-TABLE).

Using the SRM/UX File System From HP BASIC/WS

Permissions, Uids, and Gids

There is a complete discussion of file permissions in the *HP BASIC 6.1 Language Reference* (and later revisions) manual, in the Keyword Dictionary under the keywords: PERMIT, CHOWN, and CHGRP. Permissions in HP BASIC/WS under SRM/UX work just as described there, so the discussion will not be repeated here.

The HP-UX parameters uid (User ID) and gid (Group ID) may be new to the HP BASIC/WS user. The following will explain these two HP-UX parameters.

The uid together with the gid are used for all file permissions checks done by the server. The uid will also be used for jobs submitted to the HP-UX print spooler for this client.

In most cases, each client workstation will have one primary user; for example, Joe Smith will be the primary user of the client workstation with the `Client Name` of `joe_smith`. The HP-UX System Administrator or the SRM/UX System Administrator will assign Joe Smith a unique uid (e.g., uid=333) and a gid (e.g., gid=88). Now, when the SRM/UX System Administrator configures the server's /etc/srmdconf file, he or she will enter Joe Smith's uid and gid in the `Uid` and `Gid` columns for the `joe_smith` client workstation. At this point, then, all files that are created by Joe Smith, while using `joe_smith`, will be assigned uid=333 and gid=88 and all jobs that he submits to the print spooler will also use uid=333. However, be aware that any other user that uses client workstation `joe_smith` will have Joe Smith's uid and gid assigned to all of his or her files and print spooler jobs.

Compatibility Between File Types

For a complete discussion of compatibility between SRM/UX Files and HP-UX files and between SRM/UX Files and HP BASIC-UX files, see the "Data Storage and Retrieval" chapter of the *HP BASIC/WS 6.1 Programming Techniques, Volume 1* (and later revisions) manual.

Moving Around Between Volumes

You move around between volumes in the SRM/UX file system by using the volume name (as specified in the server's /etc/srmdconf file) in the LABEL field of the MSI command, for example:

```
MSI ":REMOTE 21, 5;LABEL NEWVOL"
```

```
MSI ":REMOTE 21, 5;LABEL SRMUXROOT"
```

etc.

You are not allowed to move above the root of the volume that you specify in the LABEL field of the MSI statement. If you did issue the command to go above the root of the volume—for example: `MSI "../../.."`, from /SYSTEMS—you would be left at `"/"`(the volume's root directory) from the HP BASIC/WS system's perspective, and at `/srmuxroot` from the server's

HP-UX perspective. Trying to move above a volume's root will cause no problems and no error messages will be issued; the attempt will simply be unsuccessful.

Pipes

SRM/UX has the added capability over SRM of using named pipes.

For a discussion on how to use pipes, see the "Directing Data Flow" chapter in the *HP BASIC 6.1 Interfacing Techniques* (and later revisions) manual.

Note There is a limit of 20 opens at a time on pipes. This could be 20 different pipes open, or 20 different clients opening one pipe.

Using the SRM/UX Spooling Environment From HP BASIC/WS

Spooling With Normal Priority

The SRM server allowed you to spool only a few file types: ASCII and HP-UX.

With SRM/UX you can spool any textual file type, i.e., ASCII, HP-UX, and BDAT files. You can also spool PROG files if BASIC/UX is installed and is executable on the SRM/UX server.

Before the HP BASIC/WS client user can do file spooling to any printers or plotters, the SRM/UX System Administrator must set up the SRM/UX spooling environment on the server—which also involves setting up the host workstation's HP-UX spooling environment. The Administrator will create spooling directories for the whole SRM/UX system, and will permit client workstations access to whichever spooling directories are appropriate in the server's /etc/srmdconf file. Ask your System Administrator about your client's spooling directories.

After the SRM/UX spooling environment is in place on the server, the client user can spool files in the following ways:

- you can ASSIGN TO a file in the spooling directory and then send output to this file
- you can COPY a file into the spooling directory
- you can set:
  - PRINTER IS
  - PRINTALL IS
  - PLOTTER IS
  - DUMP DEVICE IS

  to a file in the spooling directory, and then send output to this file.
- you can SAVE a program into the spooling directory
- you can STORE a program into the spooling directory if HP BASIC/UX is installed and is executable on the SRM/UX server

Once you close the file in the spooling directory, the SRM/UX server sends the file to the device specified in the /etc/srmdconf configuration file. See your SRM/UX System Administrator if you have questions about which spooling devices your client workstation is enabled to use.

If you need to abort the spooling job, consult your SRM/UX and/or HP-UX System Administrator—spooling jobs can be aborted only from the host computer's HP-UX system.

Priority Spooling

To spool files with priority over other spooling jobs, the SRM/UX System Administrator must set the permissions so that you can access the priority spooling directory.

If the Administrator has set the permissions appropriately, then you can spool files by doing any of the above operations to the PRIORITY subdirectory of the designated spooling directory.

Environment Files

If you want to pre-configure a spooling device, you may use an environment file. You specify the environment file by including its name in the spooling command string, i.e., *spoolfile.environmentfile.spoolfileextension.*

For example, if you want `file2.ASC` to be printed in the printer environment specified in the environment file `elite.asc`, then place `file2.elite.ASC` in the spool directory.

This spooling command string causes the environment file to be prefixed to the spooled file and the spooled file is printed or plotted as desired.

If You Have An HP-UX Account On The Server

If you have an HP-UX account on the server's host workstation, then you can perform some of the tasks that ordinarily call for SRM/UX System Administrator privileges.

Since you have an account on the HP-UX system, we assume you know your way around HP-UX well enough. Hence, no attempt is made to explain common HP-UX terminology or procedures.

There are three things you can do with regard to spooling if you have an HP-UX account on the server:

1. You can look at the server's /etc/srmdconf configuration file to discover the inner workings of the SRM/UX system spooling environment; for example, what are the spooling directories, what are the PRIORITY spooling directories, what directories can my client workstation access, etc.
2. You can abort spooling jobs.
   a. execute `lpstat -t`
   b. locate your spooling job's id by looking for the job with your user id (uid)
   c. execute `cancel` *spool_job_id*

   This procedure will remove your spool job from the spooler queue.
3. You can check the permissions on the PRIORITY spooling directories to see if you can use them.

7

Installing and Configuring Pascal Workstation Clients

This chapter tells how to install the Pascal Workstation System on any HP 9000 Series 200 or 300 computer to be a client on an SRM/UX system. It also describes how to establish contact with an SRM/UX server from the client workstation.

A major part of Pascal Workstation System start-up involves creating an appropriate directory structure on the SRM/UX file system. Much of this is done automatically by the SRM/UX installation process when you load the SRM/UX tape, but you must manipulate certain files to customize your SRM/UX Pascal Workstation System clients.

How you approach the client workstation start-up procedure depends on how tasks are divided between the SRM/UX System Administrator and the individual client workstation user, as well as the degree to which each client system will be customized. System customization involves modifying system boot files; modifying the TABLE and INITLIB files; and/or creating individual autostart files for each workstation.

Before You Start

Before you begin the Pascal Workstation system start-up:

- ensure all of the hardware components of your SRM/UX system are properly installed
- ensure that the SRM/UX system administrator has properly installed, configured, and started running all of the software components of the SRM/UX server

- ensure that the SRM/UX system administrator has made the appropriate entries in the server's /etc/srmdconf file for your particular workstation. See chapter 4 of this manual for server configuration details.

What You Need for Start-up

- the media which holds the Pascal Workstation system software
- the Pascal Workstation system manual set, for more detailed information about the start-up options mentioned in this chapter

Booting Pascal and Establishing Access to the SRM/UX System

For complete instructions on booting the Pascal Workstation system, refer to the manuals: *Pascal 3.X Workstation system Volume 1 and 2* for whichever version (for example: 3.2 replaces 3.X) you are installing. There you will learn how to install the Pascal Workstation system from the flexible discs shipped with the system.

The search rules for these files are NOT the same as for HP BASIC. The path searched first is /WORKSTATIONS/SYSTEMnnnnnn, where nnnnnn is the last 6 hex digits of the Pascal Workstation system client LAN card Link address.

Note Any alpha characters MUST be capitalized in the name of this directory; for example: /WORKSTATIONS/SYSTEM012ABC is correct, but /WORKSTATIONS/SYSTEM012abc is not.

If /WORKSTATIONS/SYSTEMnnnnnn does not exist, the path /WORKSTATIONS/SYSTEM is searched for the three extension boot files. If neither of these paths exist, or if the INITLIB file is not found, the boot will fail with an error: -10. If the STARTUP or TABLE files are not found, booting will be completed, but the Pascal Workstation system client system will be practically unusable. Both of these paths are relative to the root directory for volume 8, as described in the /etc/srmdconf section in chapter 4.

The names of the INITLIB, STARTUP and TABLE programs should follow the standard rules, based on the name of the SYSTEM_P file. For example, if the system file has the name SYSP33, the other files should be named INITP33, STARTP33, and TABLEP33.

Once booting is complete and TABLE has executed, rbootd is no longer used by the Pascal Workstation system client. Srmd on HP-UX and the SRM driver in the Pascal Workstation system INITLIB file take over. If *srmd*(1M) is not running or the proper modules are not in INITLIB, the client Pascal Workstation system will be unusable.

To establish access to the SRM/UX system from a running Pascal Workstation system, that did NOT boot from the SRM/UX server, do the following:

1. execute the required driver files: DATA_COMM and SRM (for the SRM connection) or DATA_COMM, LAN, and SRM (for the LAN connection)
2. manually reexecute the Pascal Workstation system's auto-configuration program: TABLE

Once the proper driver modules have been executed, the auto-configuration program can recognize the SRM/UX system.

Note The following instructions assume that the client workstation has its own flexible-disc drive.

Table 7-1. To Boot the Pascal Workstation system:

Step	Action	Result	Explanation or Note
1	Ensure that the workstation is off		
2	If the workstation has an attached disc drive, turn it on		
3	If the workstation is a Series 200, insert the disc labeled BOOT: into the drive with the lowest unit number. If the workstation is a Series 300, insert the disc labeled BOOT2: into the drive with the lowest unit number.		To determine which drive has the lowest unit number, refer to the disc drive's operator's guide
4	If the workstation has a dual flexible disc drive, insert the disc labeled SYSVOL: into the other available drive. (The SYSVOL: disc must be write-enabled)		

Table 7-1. To Boot the Pascal Workstation system: (continued)

Step	Action	Result	Explanation or Note

5	Turn on the workstation	The workstation loads the Pascal Workstation system (if booting from the server unattended; if you use attended booting, you should select the SYSTEM_P boot file from the Boot ROM menu) and then displays the prompt:
		`New system date?`
		If, instead, you get the prompt:
		`Please put SYSVOL in unit #3 and press the x key ...`
		then replace the boot disc with the disc labeled SYSVOL: and press the (x) key. (The SYSVOL: disc must be write-enabled)	

6	Enter the correct date and time by typing over the displayed date and time. Then press (Return)	The Pascal Workstation system displays the Main Command Level prompt line	Note that the time is recorded as 24-hour clock time (i.e., 1:00 p.m. is 13:00, 2:00 p.m. is 14:00, etc.)

Once you have booted the Pascal system, follow these steps to establish access to the SRM/UX system:

Table 7-2. To Establish Access to the SRM/UX system:

Step	Action	Result	Explanation or Note
1	If your Pascal Workstation system is on double-sided microdiskettes, replace the SYSVOL: disc with the disc labeled ACCESS:. Otherwise, replace the SYSVOL: disc with the disc labeled CONFIG:.		The ACCESS: or CONFIG: disc contains the driver modules required to access the SRM/UX system
2	Press (x)	The Pascal Workstation system displays: `Execute what file?`	To execute a file.
3	If you are running over the SRM connection, do step 3 (this step) and skip step 4. If you have double-sided microdiskettes, type (including the trailing period): `ACCESS:DATA_COMM.` (Return) Otherwise, type (including the trailing period): `CONFIG:DATA_COMM.` (Return)		To install the first required driver module, DATA_COMM. The required driver modules are programs that install themselves automatically when you execute them.

Table 7-2.
To Establish Access to the SRM/UX system: (continued)

Step	Action	Result	Explanation or Note
4	If you have double-sided microdiskettes, type (including the trailing period): `ACCESS:LAN.` Return		To install the required driver modules IOMPX and LANDVR.
5	Press x	The Pascal Workstation system displays: `Execute what file?`	To execute a file.
6	If you have double-sided microdiskettes, type (including the trailing period): `ACCESS:SRM.` Return Otherwise, type (including the trailing period): `CONFIG:SRM.` Return		To install the remaining required driver module, SRM.
7	If the workstation has only a single disc drive, replace the disc currently in the drive with the BOOT: or BOOT2: disc. Otherwise, ensure that the BOOT: or BOOT2: disc is still in one of the workstation's disc drives.		

Table 7-2.
To Establish Access to the SRM/UX system: (continued)

Step	Action	Result	Explanation or Note
8	Press (x)	The Pascal Workstation system displays: `Execute what file?`	To execute a file.
9	If the workstation is a Series 300, type (including the trailing period): `BOOT2:TABLE.` (Return) Otherwise, type (including the trailing period): `BOOT:TABLE.` (Return)		To manually execute the Pascal Workstation system's auto-configuration program, TABLE.
10	Replace the BOOT2: or BOOT: disc with the disc labeled ACCESS:.		
11	Press (F)	The Pascal Workstation system displays the Filer's command line.	To enter the Filer
12	Press (V)	The Filer displays a list of the volumes that the Pascal Workstation System recognizes. (See the following example display)	To see if the Pascal Workstation System recognizes the SRM/UX volume 8 at volume #5.
13	Continue with the next section on "Creating Directories on the SRM/UX System Disc."		

Example Pascal Filer Volumes Listing

```
Volumes on-line:
  1   CONSOLE:
  2   SYSTERM:
  3 # ACCESS:
  5 # PRIMARY:
  6   PRINTER:
Prefix is - PRIMARY:
```

Notice that in the listing for Volume 5 and in the last line, you should see either PRIMARY or the "Volume Name" that the SRM/UX system administrator has assigned to your client in the server's /etc/srmdconf file.

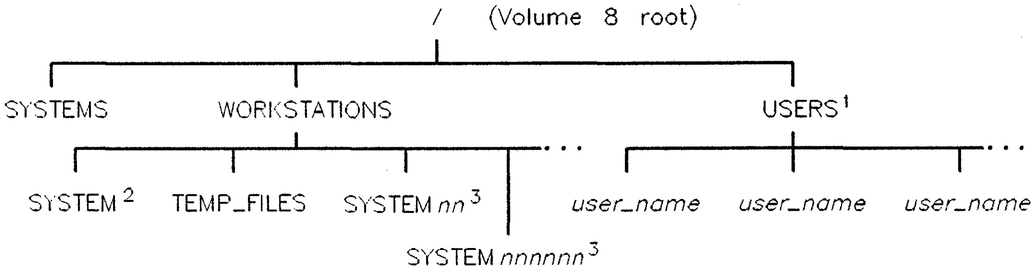
Table 7-3. Troubleshooting Hints

If:	Then:
the volume name of your SRM/UX volume 8 is not in the display	try reexecuting the DATA_COMM, LAN, SRM, and TABLE programs
the Filer's Volumes command still does not recognize volume #5	check to see whether the SRM/UX hardware is properly configured and installed. For instance, the (unmodified) TABLE program expects the select code on the workstation's SRM interface to be at its factory setting: 21. Refer to "special config" for other select codes and host nodes.
neither of the two suggestions above work	Refer to the troubleshooting appendix (Appendix B) of this manual and/or ask your SRM/UX System Administrator for help.

The following illustration shows a directory structure suitable to support Pascal Workstation Systems on an SRM/UX system:

This structure should be placed below the Volume 8 Root directory as defined in the SRM/UX server's /etc/srmdconf file. If this structure is created by the

Pascal Workstation system client user, it will get placed under the Volume 8 Root automatically.



[1] The USERS directory is optional but highly recommended.

[2] The SYSTEM directory is the default system directory, used by client workstations for which no dedicated system directory exists.

[3] *nn* and *nnnnnn* represent the two digit Node Address on SRM and the six hex character Link Level Address of the client workstation on LAN

Figure 7-1. SRM/UX Pascal Workstation System Directory Structure

Table 7-4. Create the Required Directory Structure on the Server's File System:

Step	Action	Result	Explanation or Note
1	Create the /WORKSTATIONS directory as follows:		
1a	Press M	The Filer displays: `Make file or directory? (F/D)`	To make a directory
1b	Press D	The Filer prompts: `Make what directory?`	To make a directory
1c	Type: `#5:/WORKSTATIONS` Return	The Filer prompts: `Directory is 'WORKSTATIONS' correct ? (Y/N)`	To specify the /WORKSTATIONS directory on volume #5 of the SRM/UX system disc
1d	Check the directory name and if it is correct, press Y If the directory name is wrong, press N and start over again with step 1a.	If you pressed Y the Filer creates the new directory and displays: `Directory WORKSTATIONS made` If you pressed N the Filer displays its command line and does *not* create the directory.	Y accepts the new directory. N rejects the new directory.

Table 7-4. Create the Required Directory Structure on the Server's File System: (continued)

Step	Action	Result	Explanation or Note
2	Make /WORKSTATIONS the default directory as follows:		To reduce the amount of typing involved in the next steps
2a	Press (P)	The Filer prompts: `Prefix to what directory?`	To specify a prefix directory path. The Filer prefixes every directory path with the current prefix (default) directory path.
2b	Type: `#5:/WORKSTATIONS` (Return)	The Filer displays: `Prefix is WORKSTATIONS:`	To specify the /WORKSTATIONS directory on volume #5 of the SRM/UX system disc

3	Within /WORKSTATIONS, the default directory, create the following directories (with names exactly as listed):
	`SYSTEM`
	`TEMP_FILES`
			See Step 1 of this table for a detailed example of how to create a directory.

4	Press (P) and type: /	The Filer displays the root directory's name	To return to the root directory

Table 7-4.
Create the Required Directory Structure on the Server's File System: (continued)

Step	Action	Result	Explanation or Note
5	Create the directory: `/USERS` Specify the directory name as you choose.		To create a /USERS directory on the SRM/UX file system. Within this directory, create a personal working directory for each workstation user.
6	Make the default directory (as specified in Step 5): `/USERS`		To reduce the amount of typing involved in the next step. See Step 2 for a detailed example of how to specify a prefix (default) directory.
7	Create a personal working directory for every workstation user. For example: john_public		Valid directory names are 1 to 14 characters long (1 to 16 characters long if the server administrator has converted to long file names—see chapter 4 for details), and may have upper-and lower case letters, digits, underscores (_), and periods (.).
8	Change to the default directory, i.e., the Volume 8 Root		

Table 7-4.
Create the Required Directory Structure on the Server's File System: (continued)

Step	Action	Result	Explanation or Note
9	Continue creating additional directories below the SRM/UX root, as needed, e.g., UTILS		To create an organized directory structure on the SRM/UX server's file system
10	Continue with the next section on "Placing Pascal system files in the SYSTEM Directory."		

Placing Pascal Workstation System Files in the SYSTEM Directory

To have workstations:

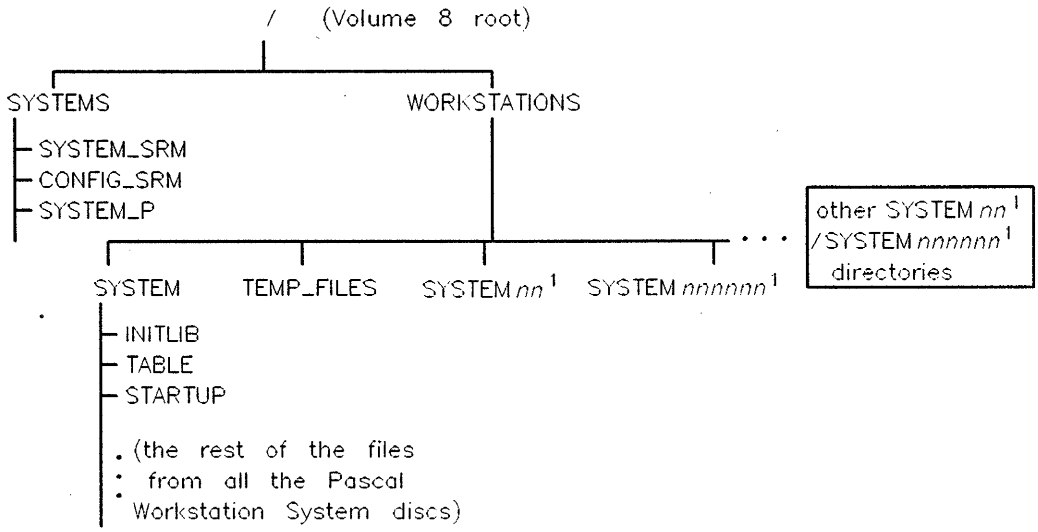
- automatically load the Pascal Workstation system from the SRM/UX system disc, and
- load or access any Pascal Workstation system subsystems, utility programs, libraries, etc. from the SRM/UX file system,

use one of the following methods:

Table 7-5. Place system files in /SYSTEMS

Method	Requirement(s)	Result
To boot in unattended mode, let each workstation's BootROM automatically boot the first system file in the /SYSTEMS directory.	■ The Pascal system boot file (SYSTEM_P) must be the *first* bootable system file[1] in /etc/boottab. ■ The rest of the Pascal system files must be in the /WORKSTATIONS/SYSTEM directory.	*All* workstations automatically boot the Pascal system.
To boot in attended mode, repeatedly depress or hold down the (space) key after a SYSBOOT or cycle power until you get the BootROM menu of bootable system files to appear on the right-hand side of the screen. Then enter the characters to select the SYSTEM_P system boot file.	The Pascal system boot file (SYSTEM_P) must be in the /WORKSTATIONS/SYSTEM directory, along with the rest of the Pascal system files.	If each workstation user selects the same SYSTEM_P file, then they will all be configured with the same Pascal system.

[1] A bootable system file is a system file whose name begins with SYS.



[1] *nn* and *nnnnnn* represent the two digit Node Address on SRM and the six hex character Link Level Address of the client workstation on LAN

Figure 7-2. Required Location of Pascal Workstation System Files

To place Pascal system files in the /WORKSTATIONS/SYSTEM directory on the SRM/UX file system from the client, follow these steps:

Table 7-6.
System Files in /WORKSTATIONS/SYSTEM on Server's Volume 8:

Step	Action	Result	Explanation or Note
1	Make /WORKSTATIONS/SYSTEM the default directory as follows:		To reduce the amount of typing involved in the next steps
1a	Press (P)	The Filer prompts: `Prefix to what directory ?`	To specify a prefix directory path. The Filer prefixes every directory path with the current prefix (default) directory path.
1b	Type: `/WORKSTATIONS/SYSTEM` (Return)	The Filer displays: `Prefix is SYSTEM:`	To specify the /WORKSTATIONS/SYSTEM directory on the SRM/UX volume 8
2	Replace the ACCESS: disc in the workstation's disc drive with the disc labeled BOOT:.		All of the files on the BOOT: disc are the same as those on the BOOT2: disc, *except* for the INITLIB file.

Table 7-6.
System Files in /WORKSTATIONS/SYSTEM on Server's Volume 8: (continued)

Step	Action	Result	Explanation or Note
3	Copy all of the BOOT: disc's files except SYSTEM_P and INITLIB to the default directory /WORKSTATIONS/SYSTEM on the SRM/UX volume 8 as follows:		Later you will be copying SYSTEM_P to the /SYSTEMS directory on the SRM/UX volume 8, and modifying the INITLIB files from the BOOT: and BOOT2: discs as you place them on the SRM/UX volume 8.
3a	Press (F)	The Filer prompts: `filecopy what file ?`	To copy one or more files
3b	Type: `BOOT:?,$` (Return)	The Filer lists the first file on the disc and asks if you want to copy it	The ? is a wildcard that tells the Filer to individually list each file and ask you if you want to copy it from the disc. The $ is a wildcard that tells the Filer to give the copy of the file the same name as the original. The comma separates the source file specification from the destination file specification.

Table 7-6. System Files in /WORKSTATIONS/SYSTEM on Server's Volume 8: (continued)

Step	Action	Result	Explanation or Note
3c	When the Filer displays the file names SYSTEM_P and INITLIB, press N For every other file, press Y.	The Filer reads and then writes each file you respond "yes" to.	To respond "yes" to every file-copy except SYSTEM_P and INITLIB.
4	Copy SYSTEM_P to the /SYSTEMS directory on the SRM/UX volume 8 as follows:		SYSTEM_P is the Pascal Workstation system boot file
4a	Press F	The Filer prompts: `filecopy what file ?`	To copy a file
4b	Type: `BOOT:SYSTEM_P, /SYSTEMS/$` Return	The Filer copies the file	The $ is a wildcard that tells the Filer to give the copy of the file the same name as the original. The comma separates the source file specification from the destination file specification.
5	Replace the disc in the workstation's disc drive with a Pascal Workstation system disc *other than* BOOT: and BOOT2:		

Table 7-6.
System Files in /WORKSTATIONS/SYSTEM on Server's Volume 8: (continued)

Step	Action	Result	Explanation or Note
6	Copy all of the files on the new disc to the default directory, /WORKSTATIONS/SYSTEM on the SRM/UX volume 8, as follows:		
6a	Press F	The Filer prompts: `filecopy what file ?`	To copy one or more files
6b	Type: *disc_name*`:=,$` Return where *disc_name* is the name of the new disc (including the colon after the name).	The Filer copies the files on the new disc one after the other.	The = is a wildcard that stands for any combination of characters and its use here causes the Filer to copy every· file on the new disc, without prompting. *Example* If you inserted the disc labeled ACCESS: into the workstation's disc drive, you would type: `ACCESS:=,$` Return

Table 7-6. System Files in /WORKSTATIONS/SYSTEM on Server's Volume 8: (continued)

Step	Action	Result	Explanation or Note
7	Repeat steps 5 and 6 for every remaining Pascal Workstation system disc (for example, SYSVOL: , CMP: , ASM: ,etc.		To copy all of the Pascal Workstation system files to the /WORKSTATIONS/SYSTEM directory on the SRM/UX volume 8
8	(P) /WORKSTATIONS		To prefix back to /WORKSTATIONS

For convenience, designate the /WORKSTATIONS/SYSTEM directory on the SRM/UX file system as your workstation's system volume. This allows you to access the SRM/UX system's copies of the Pascal Workstation system's subsystems by pressing keys, such as (E) for Editor, (F) for Filer, etc. To do this, follow the steps in the next table.

Table 7-7. To Designate Your Workstation's system Volume:

Step	Action	Result	Explanation or Note
1	Press Q	The Pascal Workstation system displays the Main Command Level prompt line	To exit from (or quit) the Filer
2	Press X	The Pascal Workstation system prompts: `Execute what file ?`	To execute a program
3	Type (including the trailing period): `TABLE.` Return	The TABLE auto-configuration program recognizes the SRM/UX server's /WORKSTATIONS/SYSTEM directory on volume 8 as volume #45	To reexecute the TABLE auto-configuration program
4	Press W Press S	The Pascal Workstation system prompts: `What new system unit number ?`	To designate a new system volume
5	Type: `#45:` Return	The Pascal Workstation system makes volume #45 (the /WORKSTATIONS/SYSTEM directory) the system volume	

Table 7-7.
To Designate Your Workstation's system Volume: (continued)

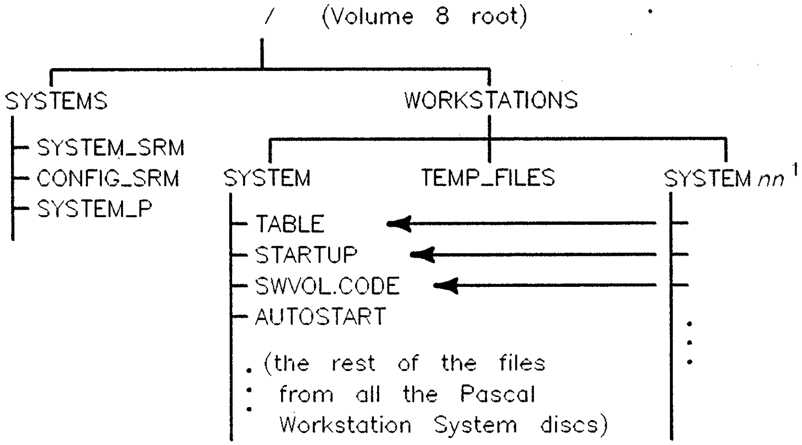
Step	Action	Result	Explanation or Note
6	Press Q	The What command terminates	To quit the What command. Continue on with the next section on "Giving Each Workstation Access to system files."

```
Assembler Compiler Editor Filer Librarian
library system volume Default volume Quit

ASSEMBLER:  SYSTEM:ASSEMBLER
COMPILER:   SYSTEMCMP:COMPILER
EDITOR:     SYSTEM:EDITOR
FILER:      SYSTEM:FILER
LIBRARIAN:  SYSTEM:LIBRARIAN

* system volume:  SYSTEM:
: Default volume: WORKSTATIONS:
```

Giving Each Workstation Access to System Files

To make the Pascal Workstation system files available in every workstation's own system directory, create links from those directories to the Pascal Workstation system files.

The illustration below shows the SRM/UX file system's directory structure with one possible set of links. The links are from one client workstation's system directory to the server's Pascal Workstation system files.



[1] *nn* represents the two digit Node Address of the Client workstation on the SRM Connection.

Figure 7-3.
Access to Pascal Workstation System Files Through Links

Note The two character client Node Address, that is specified for clients on the LAN connection in the server's /etc/srmdconf LAN-CLIENTS table, is meaningless to Pascal. Pascal Workstation system recognizes only the six-hex-digit Link Level Address.

Table 7-8. Create Links to the Pascal System Files

Step	Action	Result	Explanation or Note
1	Press (F)	The Pascal Workstation system displays the Filer prompt line	To enter the Filer
1a	Press (P)	The Filer prompts: `Prefix to what directory ?`	To prefix to a directory
1b	Type: `/WORKSTATIONS`	The Filer displays: `Prefix is WORKSTATIONS:`	To specify /WORKSTATIONS as the default directory
1c	Within /WORKSTATIONS, create a dedicated system directory for every workstation on your SRM/UX system (See the following examples).		To dedicate a system directory to a workstation, append the workstation's two (SRM card) or six character (LAN interface) node address to the directory name SYSTEM. (If a workstation's node address is only a single digit, precede it with a 0 (zero) to make it two digits long.) Every workstation's node address is either readable from an SRM card or is the last 6 digits of the workstation's LAN address.

Examples:

If a client workstation's node address is 8 (running over the SRM connection), create the directory:

```
SYSTEM08
```

If a client workstation's Link Level Address is 0x80009012345 then its client Node Address is 012345 (running over the LAN connection), so create the directory:

```
SYSTEM012345
```

Table 7-8.

Step	Action	Result	Explanation or Note
1d	Press D	The Filer prompts: `Duplicate or Move? (D/M)`	To create a duplicate link to a file
2	Press D	The Filer prompts: `Dup_link what file?`	To tell the Filer that you want to duplicate, not move, a file
3	Type: `SYSTEM/?,SYSTEMnn/$` Return OR `SYSTEM/?,SYSTEMnnnnnn/$` Return where nn is the client workstation's two-digit node address, and nnnnnn is the client workstation's six-hex-digit emulated node address.	The Filer lists the first file in the directory, /WORKSTATIONS/SYSTEM, and asks if you want to create a duplicate link to it	Remember to use a leading 0 (zero) with single-digit node address. The ? is a wildcard that tells the Filer to display one file name at a time and allow you to select only those files you want to create a link to.

Table 7-8. (continued)

Step	Action	Result	Explanation or Note
4	Press N when the Filer displays the file name AUTOSTART. Press Y when every other file name appears.	As the Filer creates a duplicate link to each file you respond "yes" to, the Filer displays each file name.	To respond "yes" to every duplicate link except for AUTOSTART.
5	Repeat steps 1 through 5 as many times as necessary to create a set of links from every workstation's own system directory to the Pascal Workstation system files in the /WORKSTATIONS/SYSTEM directory.	The Pascal Workstation system files are then accessible from every workstation's own system directory, and can be modified independently in each dedicated directory if so desired	
6	Press Q	The Pascal Workstation system displays its Main Command Level prompt line.	To exit from (quit) the Filer
7	Continue with the next section on "Adding Required Driver Modules to the Initialization Library."		

Adding Required Driver Modules to the Initialization Library

Upon bootup, the Pascal Workstation System automatically loads the driver modules required to access the SRM/UX system, if you include them in the system's initialization library (INITLIB file). The driver modules that provide access to the SRM/UX system are: DATA_COMM, IOMPX, LANDVR, and SRM. LANDVR and IOMPX are not needed if you are running over the SRM connection only. Both IOMPX and LANDVR are contained in the file "LAN" on the ACCESS: disc. The Pascal Workstation System's Librarian makes it easy for you to add these modules to those already contained in the system's original INITLIB file.

The INITLIB file on the Pascal Workstation System's BOOT: disc (for Series 200 computers), differs slightly from the INITLIB file on the BOOT2: disc (for Series 300 computers). To each of these two INITLIB files, add the DATA_COMM and SRM driver modules (if on SRM) or DATA_COMM, IOMPX, LANDVR, and SRM (if on LAN), as well as any other drivers you would normally want (e.g., GPIO), and create two new initialization libraries: one for client workstations that are Series 200 computers and one for client workstations that are Series 300 computers. If you will be running over the LAN connection, add the LAN driver module to INITLIB as well.

Both new initialization libraries will go into the SYSTEM directory of the SRM/UX volume 8. So that you can easily distinguish between the two initialization libraries, name one INIT_200 and the other INIT_300. Of course, you are free to name the libraries as you wish; these names are not required by the Pascal Workstation system.

The following diagram illustrates the process described above:

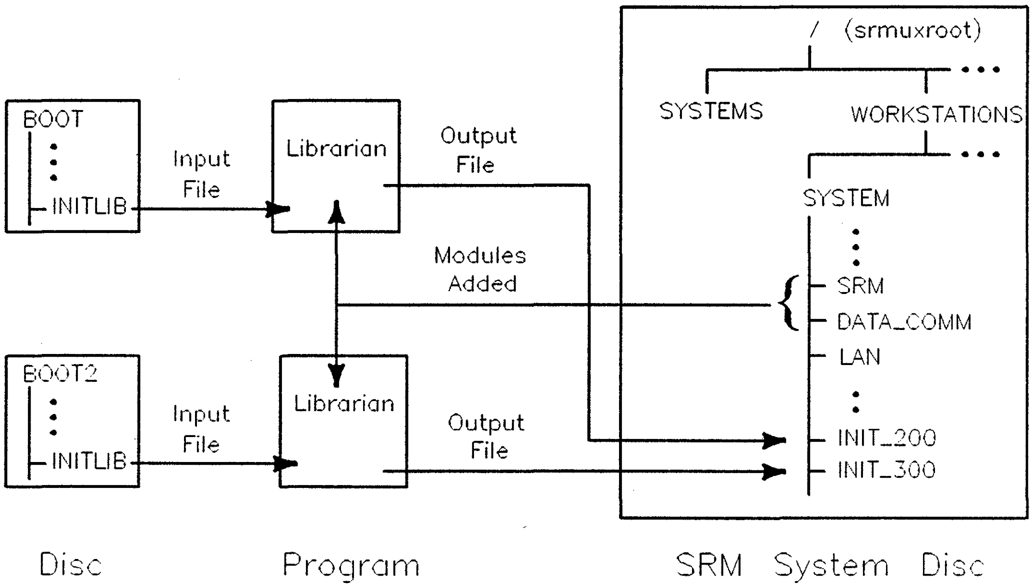


Figure 7-4. Adding SRM/UX Driver Modules to the Initialization Library

---

Note To create an initialization library for Series 200 workstations *AND* Series 300 workstations, you must perform the following procedure *TWICE* (once for each type of workstation).

Table 7-9.
Create a system Initialization Library for Series 200 *OR* Series 300 Workstations:

Step	Action	Result	Explanation or Note
1	Press L	The Librarian displays its prompt line and status	To load the Librarian. The Librarian should be loaded from the SRM/UX volume 8.
2	Specify the path to the new initialization library you will create, as follows:		
2a	Press O		To specify the name of the Output file
2b	Enter one of the file names shown in Table 6-10 below (INIT_200 or INIT_300)	The Librarian displays the message: `COPYING`	To name the output file after the workstations it is intended for. The output file will be created in default directory, /WORKSTATIONS, where all of the other Pascal Workstation system files are.

Table 7-10. INIT_200 and INIT_300

If you are creating an initialization library for:	then type (including the trailing period):
Series 200 workstations	INIT_200. Return
Series 300 workstations	INIT_300. Return

Table 7-10.

Step	Action	Result	Explanation or Note
3	If you are creating an initialization library for Series 200 workstations, insert the BOOT: disc into the local disc drive. If you are creating an initialization library for Series 300 workstations, insert the BOOT2: disc into the local disc drive.		BOOT: contains the INITLIB file for Series 200 and BOOT2: contains the INITLIB file for Series 300 workstations
4	Specify the name of the source INITLIB file that you will add modules to, as follows:		
4a	Press (I)		To specify the name of the input file
4b	Enter one of the file names shown in Table 6-11 below.	The Librarian shows INITLIB as the name of the input file and displays the line: `M input Module: KERNEL`	The Input file is the INITLIB on the BOOT: or BOOT2: disc.

Table 7-11. BOOT:INITLIB and BOOT2:INITLIB

If you are creating an initialization library for:	then type (including the trailing period):
Series 200 workstations	BOOT:INITLIB. (Return)
Series 300 workstations	BOOT2:INITLIB. (Return)

Table 7-10.

Step	Action	Result	Explanation or Note
5	Press (T)	After a few moments, the name of a new module (KBD) appears.	To transfer (copy) the module KERNEL to the output file

Note *Read the instructions for the next step before you proceed.* Each time you transfer a module to the output file, a new module name appears. You will continue to copy modules from INITLIB to the output file *UNTIL* the name of the module LAST appears. Do *NOT* transfer the module LAST at that time.

Table 7-10.

Step	Action	Result	Explanation or Note
6	Repeat step 5 *until* you see the line: `M input Module: LAST` near the bottom of the display. *Do not copy the module LAST yet*!	A new module name appears after you transfer the current module to the output file.	To transfer all of the INITLIB modules *except* LAST to the output file
7	Get the required SRM, LAN, and DATA_COMM drivers and include them in the output file, as follows:		The SRM, LAN, and DATA_COMM drivers allow the Pascal Workstation system to access an SRM/UX system.
7a	Press I	The Librarian clears the current input file name from the display	To change the input file.
7b	Type (including the trailing period): `/WORKSTATIONS /SYSTEM/DATA_COMM.` Return	The module name `DATA_COMM` appears near the bottom of the display	To specify the file containing the DATA_COMM driver. This file is in the default directory, /WORKSTATIONS/SYSTEM, on the SRM/UX volume 8 root.
7c	Press A		To tell the Librarian to transfer all modules from the input file, DATA_COMM, to the output file. (In this case, there is only one module in the input file.)

Table 7-10. (continued)

Step	Action	Result	Explanation or Note
7d	Press I	The Librarian clears the current input file from the display	To change the Input file
7e	If the clients using INITLIB will never access the server over the LAN connection, proceed to step 7i.		To avoid putting unnecessary modules in the INITLIB file.
7f	Type (including the trailing period): `/WORKSTATIONS /SYSTEM/LAN.` Return	The module name `IOMPX` appears near the bottom of the display.	To specify the file containing the LAN driver. This file is in the default directory, /WORKSTATIONS/SYSTEM on the SRM/UX volume 8.
7g	Press A		To tell the Librarian to transfer all modules from the input file, LAN, to the output file. (There are two modules in the input file: IOMPX and LANDVR.)
7h	Press I	The Librarian clears the current input file from the display	To change the Input file

Table 7-10. (continued)

Step	Action	Result	Explanation or Note
7i	Type (including the trailing period): `/WORKSTATIONS /SYSTEM/SRM.` Return	The module name `SRM` appears near the bottom of the display.	To specify the file containing the SRM driver. This file is in the default directory, /WORKSTATIONS/SYSTEM on the SRM/UX volume 8 root.
7j	Press A		To tell the Librarian to transfer all modules from the input file, SRM, to the output file. (There is only one module in the input file.)
8	Now transfer the module LAST to the output file, as follows:		
8a	Press I	The Librarian clears the current input file name from the display.	To change the Input file
8b	Enter one of the file names shown in Table 6-11 below.	The module name `KERNEL` appears near the bottom of the screen.	The INITLIB file on the BOOT: or BOOT2: disc contains the module LAST.

Table 7-11. BOOT: and BOOT2:—LAST

If you are creating an initialization library for:	then type (including the trailing period):
Series 200 workstations	BOOT:INITLIB. (Return)
Series 300 workstations	BOOT2:INITLIB. (Return)

Table 7-10.

Step	Action	Result	Explanation or Note
8c	Press (M)		To directly request a different module
8d	Type: `LAST` (Return)		To select module LAST
8e	Press (T)	Now all of the modules required to access SRM/UX system resources are in the new initialization library.	To transfer the module LAST to the output file.
9	Type (K)	The Librarian saves the output file INIT_200 or INIT_300 in the default directory, /WORKSTATIONS/SYSTEM, on the SRM/UX volume 8 root.	To "keep" the output file
10	Press (Q)	The Pascal Workstation system displays its Main Command Level prompt line.	
11	If you just created an initialization library for Series 200 workstations and you want to create one for Series 300 workstations, or the other way around, repeat steps 1 through 10. If you are finished creating the initialization library or libraries, continue with the next section on "Giving Each Workstation Access to the Correct Initialization Library."		

Giving Each Workstation Access to the Correct Initialization Library

For a workstation to boot the Pascal Workstation system from the SRM/UX server's file system, an initialization library must be available in the workstation's own system directory. If you have followed this procedure linearly from the beginning to this point, the client workstations don't yet contain an INITLIB in their system directories.

The /WORKSTATIONS/SYSTEM directory, however, contains the initialization libraries you just created for the client workstations. The files INIT_200 and INIT_300 contain modules required by Series 200 and Series 300 workstations, respectively.

To make the correct initialization library available in every workstation's own system directory, create links to INIT_200 and INIT_300. For example, the file INIT_300 might be called INITLIB (as required by the Pascal Workstation system) in a client workstation's own system directory.

To create a link from every client workstation's system directory to the correct initialization library, either INIT_200 or INIT_300, follow these steps:

Table 7-11. Create a Link to Either INIT_200 or INIT_300:

Step	Action	Result	Explanation or Note
1	Press (F)	The Filer displays its prompt line	To enter the Filer
2	Press (D)	The Filer prompts: `Duplicate or Move ? (D/M)`	To create a duplicate link to a file
3	Press (D)	The Filer prompts: `Dup_link what file?`	To tell the Filer that you want to duplicate, not move, a file.
4	On your hardware configuration worksheet(s) from chapter 3, look up the node address of a client workstation whose system directory does not yet have a link to the proper initialization library. Start with the first workstation listed on the worksheet		To obtain the value you need for the next step
5	Enter one of the responses shown in Table 6-12 below.	The Filer creates a link to INIT_200 (named INITLIB there) from the client workstation's own system directory.	Remember to use a leading 0 (zero) with single-digit node addresses

Table 7-12. Series 200 and 300 /SYSTEMnn/INITLIB

If the node address belongs to:	then type:
a Series 200 workstation	INIT_200,../SYSTEMnn/INITLIB (Return) where nn is the interface's two digit node address
a Series 300 workstation	INIT_300,../SYSTEMnn/INITLIB (Return) where nn is the interface's two or six digit node address

Table 7-18.

Step	Action	Result	Explanation or Note
6	Repeat steps 2 through 5 as many times as necessary to create a link from every client workstation's system directory to the correct initialization library.	This makes the correct initialization library available in every workstation's system directory	

If a workstation does not have its own system directory, then the necessary Pascal Workstation system files, including INITLIB, must be in the /WORKSTATIONS/SYSTEM directory on the server. At this point in the configuration process the /WORKSTATIONS/SYSTEM directory contains all of the system files, *except* INITLIB from the Pascal Workstation system discs.

The /WORKSTATIONS/SYSTEM directory, however, does contain the initialization libraries INIT_200 and INIT_300 with the driver modules required to access the SRM/UX system. The Pascal Workstation system can't find and load either INIT_200 or INIT_300 because it looks for the INITLIB file.

By creating a link from INIT_200 or INIT_300 to the file INITLIB, you make the initialization library available in the /WORKSTATIONS/SYSTEM directory.

Caution

INIT_200 may not work for Series 300 workstations, and INIT_300 will not work for Series 200 workstations. Any workstation without its own, dedicated system directory will try to use the INITLIB file in the /WORKSTATIONS/SYSTEM directory. Therefore, make the default INITLIB file whichever file (either INIT_200 or INIT_300) would successfully serve a *majority* of the client workstations on your SRM/UX system, or any new system you may attach before creating its own dedicated system directory.

Table 7-19. Create a Link from INIT_200 or INIT_300 to INITLIB:

Step	Action	Result	Explanation or Note
1	Press D	The Filer prompts: `Duplicate or Move ? (D/M)`	To create a duplicate link to a file
2	Press D	The Filer prompts: `Dup_link what file?`	To tell the Filer that you want to duplicate, not move, a file
3	If you expect to use INITLIB for Series 200 computers, then type: `INIT_200,INITLIB` Return If you expect to use INITLIB for Series 300 computers, then type: `INIT_300,INITLIB` Return	The Filer creates a link from INIT_200 or INIT_300 to the file INITLIB in the default directory /WORKSTATIONS/SYSTEM.	
4	Press Q	The Pascal Workstation system displays its Main Command Level prompt line	To exit from (quit) the Filer

After booting the Pascal Workstation System from the SRM/UX server, client workstation users will be able to access the contents of the SRM/UX volume(s) for which they are given access in the server's /etc/srmdconf file and for which TABLE has provided access. Client workstation users will not be allowed access to the SRM/UX server's HP-UX file system anywhere above the root directories of any of the volumes they have been given access to. However, the SRM/UX System Administrator can allow clients to have access to as many volumes as they need under the HP-UX root directory structure.

Unless you want to configure your SRM/UX system so that the Pascal Workstation system clients can access multiple volumes, you are done installing the Pascal clients at this point.

Giving Pascal Clients Access to Multiple SRM/UX Volumes

This section assumes that your SRM/UX System Administrator has made the necessary entries in the /etc/srmdconf file on the server to associate particular Volume numbers and Volume names with particular directories.

If your SRM/UX system has directories not contained in the Volume 8 subtree to which users need access, then you must modify the Pascal Workstation system's auto-configuration program, TABLE, and give clients access to the new TABLE program.

The TABLE auto-configuration program assigns a unit number to each mass storage device available to a workstation when it runs. The system directory and default on Volume 8 directory are the only SRM/UX directories that the unmodified TABLE program can recognize and assign a unit number to. To have TABLE recognize and assign unit numbers to other SRM/UX directories, you must edit and recompile TABLE's source program: CTABLE. This section describes how to perform such a configuration change.

Modifying the CTABLE Program

The Pascal Workstation System's CTABLE file already contains source code that assigns unit #5 to the SRM/UX volume 8 root and tries to assign unit #45 to the system directory from which the client booted.

Table 7-20. Allocate New Unit Numbers To Other SRM/UX Volumes

Step	Action	Result	Explanation or Note
1	Get the address number associated with each volume you want to automatically bring on-line. This information is available in the VOLUME-TABLE section of the file /etc/srmdconf in the server's file system. Your SRM/UX system administrator should provide these numbers to you.		
2	Record the select code of the card over which you are talking to the server, (the select code is probably 21)		
3	Return to the client workstation from which you began and Press (E) (for Editor)	The Editor displays the prompt: `file?`	To enter the Editor

Table 7-20.
Allocate New Unit Numbers To Other SRM/UX Volumes (continued)

Step	Action	Result	Explanation or Note
4	Type: *CTABLE Return	The Editor displays the source code	To edit the CTABLE source program in the default system directory on the SRM/UX file system.
5	Near the beginning of the source program, you will see the note shown below. Using the arrow keys, move the cursor to the beginning of the first blank line below the "box".		To prepare to insert a required compiler directive in the source program

```
(************************************************)
(*                                              *)
(*  Note: You will need to use the following    *)
(*  compiler directive if the 'INTERFACE'       *)
(*  file is not in your current LIBRARY.        *)
(*  Modify the volume name as necessary for     *)
(*  your configuration.                         *)
(*                                              *)
(*  $search  'CONFIG:INTERFACE. '$              *)
(*                                              *)
(************************************************)
```

Table 7-22.

Step	Action	Result	Explanation or Note
6	Press (I)	The Editor's command line disappears	To insert text
7	Type the compiler directive shown below		

Type (including the period after INTERFACE):

```
$search '#5:/WORKSTATIONS/SYSTEM/INTERFACE.'$
```

Table 7-22.

Step	Action	Result	Explanation or Note
8	Press (Select) or (EXECUTE)	The Editor's command line appears again	To accept the new line.
9	Press (J) and then (E)	The Editor displays the end of the source program	
10	Use the up arrow key to scroll upward through the program until you see the section of code shown below		This section of code assigns unit numbers to SRM/UX volumes. The statement **`tea_srm( 45,sc,ba,du);`** tries to assign unit #45 to the system volume from which the client booted.

```
{ duplicate entries for prefixing down the SRM }

with SRM_dav do
  begin
   {  tea_srm( 46,sc,ba,du);  {free}
   tea_srm( 45,sc,ba,du);  {for possible use as the system unit}
  end;  {with}
```

Table 7-23. sc, ba, and du

Variable	Description	Default Value
sc	The select code of the workstation's SRM or LAN interface. This should have been recorded in step 2 of this procedure.	21
ba	The node address of the server. Set this value to be what the SRM/UX system administrator has assigned and entered in the VOLUME-TABLE of the /etc/srmdconf file.	In a one server system, usually 0
du	The volume address of the server's SRM/UX directory. This was recorded in step 1.	usually 8

Table 7-22.

Step	Action	Result	Explanation or Note
11	Use the Editor's Delete, Insert, and eXchange functions to modify the section of CTABLE shown above as required for your SRM/UX system configuration. *Read the example below before you proceed; use it as a guide.*		Remember to press Select or EXECUTE to accept the changes from one function and redisplay the Editor's command line to select another function.

Example:

To illustrate the first set of changes required to give workstations access to multiple SRM/UX volumes, let us assume you have the following SRM/UX volumes available:

```
                        A server at node address 0 with:

CASE ONE:               a directory at volume address 8
CASE TWO:               a directory at volume address 9
CASE THREE:             a directory at volume address 10
                        The workstation's SRM interface cabled to
                        this server is at select code 21.

                        Another server at node address 1 with:

CASE FOUR:              a disc at volume address 8
CASE FIVE:              a disc at volume address 9
                        The workstation has a second SRM interface
                        at select code 22 cabled to this server.
```

CTABLE contains the program line for CASE ONE (ba is assigned the value 0 and du is assigned the value 8, if no other value for them is specified). To allow access to the volumes in CASE TWO through CASE FIVE, assign a new unit number to each disc, by modifying CTABLE as shown:

```
 tea_srm( 46, sc, ba, 9);    {change du to 9 for CASE TWO}
 tea_srm( 47, sc, ba, 10);   {Add this line for CASE THREE}
 tea_srm( 48, 22, 1, 8);     {Add this line for CASE FOUR}
 tea_srm( 49, 22, 1, 9);     {Add this line for CASE FIVE}

 tea_srm( 45, sc, ba, du);   { This line is for CASE ONE}
                                    {for possible use as the system unit}
          end;  {with}
```

Table 7-22.

Step	Action	Result	Explanation or Note
12	Use the down arrow key to scroll through the program until you see the section of code shown below.		This section of code recognizes the SRM/UX system (root) directory as unit #5. Unit #45 references the client workstation's own system directory.

```
{ prefix the primary and secondary SRM unit entries  }

if not unit_prefix_successful('#5:/') then
  {do nothing};  {tries to set up uvid for possible default unit assignment below}

{ if not unit_prefix_successful('#46:/?') then zap_assigned_unit(46);  {free}

if not unit_prefix_successful('#45:'+srmsysprefix+srmnode(unitable^[45].sc)) then
  if not unit_prefix_successful('#45:'+srmsysprefix) then
    zap_assigned_unit(45);
```

Table 7-22.

Step	Action	Result	Explanation or Note
13	Use the Editor's Delete, Insert, and eXchange functions to modify the section of CTABLE shown above as required for your SRM/UX system configuration. *Read the example below before you proceed; use it as a guide.*		Remember to press (Select) or (EXECUTE) to accept the changes from one function and redisplay the Editor's command line to select another function.

Example:

To illustrate the changes required to give workstations access to multiple SRM/UX volumes, the example SRM/UX system configuration shown above in step 11 continues here.

For the Pascal Workstation System to recognize additional SRM/UX volumes (besides the SRM/UX system volume) by the new unit numbers you assigned to them, modify CTABLE as follows:

```
{  prefix the primary and secondary SRM unit entries }

if not unit_prefix_successful('#5:/") then
  {do nothing};  {tries to set up uvid for possible default unit assignment below}

  ###Remove the { comment delimiter and the ? for CASE TWO, as shown below.

if not unit_prefix_successful('#46:/') then zap_assigned_unit(46); {free}

  ###Add these lines for CASE THREE, CASE FOUR, and CASE FIVE, respectively:

if not unit_prefix_successful('#47:/') then zap_assigned_unit(47);  {free}
if not unit_prefix_successful('#48:/') then zap_assigned_unit(48);  {free}
if not unit_prefix_successful('#49:/') then zap_assigned_unit(49);  {free}

if not unit_prefix_successful('#45:'+srmsysprefix+srmnode(unitable^[45].sc)) then
  If not unit_prefix_successful('#45:'+srmsysprefix) then
    zap_assigned_unit(45);
```

Table 7-22.

Step	Action	Result	Explanation or Note
14	From the Editor's command line, press Q	The Editor displays several exit options	To tell the Editor that you want to finish the editing session
15	Press W	The Editor prompts for the name of the new file	To write the edited CTABLE out to a new file, while keeping the original CTABLE file unchanged
16	Type: `*NEWCTABLE` Return	The Editor displays the message `Writing ...` and then prompts: `Exit from or Return to the Editor ?`	To save the edited CTABLE as NEWCTABLE in the default system directory on the server's SRM/UX file system
17	Press E	The Pascal Workstation system displays its Main Command Level prompt line	To exit from the Editor
18	Continue with the next section on "Compiling, Running, and Verifying the Modified CTABLE Program."		To produce a new auto-configuration program that can recognize all of the available directories on your SRM/UX system

Compiling, Running, and Verifying the Modified CTABLE Program

To compile, run, and verify your NEWCTABLE auto-configuration program, follow these steps:

Table 7-23. Compile, Run, and Verify your NEWCTABLE program

Step	Action	Result	Explanation or Note
1	Press C	The Compiler prompts: `Compile what text?`	To load the Compiler
2	Type: `*NEWCTABLE` Return	The Compiler prompts: `Printer Listing?`	To specify the modified CTABLE program you just created and saved to the default system directory
3	Press N		For no listing

Table 7-23.
Compile, Run, and Verify your NEWCTABLE program (continued)

Step	Action	Result	Explanation or Note
4	Press (Return)	The Compiler compiles the source program and displays status messages as the compilation progresses and then finishes	To accept the displayed output file name of NEWCTABLE.CODE.
5	If your did not get any compilation errors, press (R)		To run the NEWCTABLE.CODE program
6	Press (F)	The Pascal system displays the Filer's command line	To enter the Filer
7	Press (V)	The Filer displays a list of the volumes that the workstation recognizes. The listing should show the new unit numbers (for example, 46,47,48, and so on) you assigned to your additional SRM/UX volumes	To verify that the new auto-configuration program recognizes all of your SRM/UX volumes
8	Press (Q)	The Pascal Workstation system displays its Main Command Level prompt line	To exit from the Filer
9	Continue with the next section on "Giving Each Workstation Access to the New TABLE Program."		To make the new auto-configuration program available to all of the workstations on your SRM/UX system

Giving Each Workstation Access to the New TABLE Program

Your newly created auto-configuration program, NEWCTABLE.CODE, and the old, unmodified auto-configuration program, TABLE, are both in the /WORKSTATIONS/SYSTEMnnnnnn directory (where nnnnnn is your client's six hex-digit node address over the LAN connection) or the /WORKSTATIONS/SYSTEMnn directory (where nn is your client's two-digit node address on the SRM connection). Also, at this point in the configuration process, every client workstation's own system directory contains a link to the old TABLE program.

For workstations to boot the Pascal Workstation system from the SRM/UX server's file system, the new auto-configuration program NEWCTABLE.CODE:

- must be called TABLE,
- must be accessible from every client workstation's own system directory

You can accomplish both of these things by overwriting the old TABLE auto-configuration program with the new one. Overwriting changes the contents of a file while preserving its original name and its links.

To overwrite the old auto-configuration program, TABLE, with the new auto-configuration program, NEWCTABLE.CODE, follow these steps:

Table 7-24. Overwrite TABLE with NEWCTABLE.CODE

Step	Action	Result	Explanation or Note
1	Press (F)	The Pascal Workstation system displays the Filer's command line	To enter the Filer
2	Press (F)	The Filer prompts: `filecopy what file ?`	To attempt to copy a file (or perform a filecopy)

3	Type:
	`NEWCTABLE.CODE,TABLE`
	(Return)
		The Filer displays:
		`Reading ...`
		and then:
		`SYSTEM:TABLE exists ... Remove/Overwrite/Neither? (R/O/N)`
			To copy the file NEWCTABLE.CODE to TABLE in the default directory, /WORKSTATIONS/SYSTEM

4	Press (O)	The Filer overwrites the file TABLE with the contents of NEWCTABLE.CODE	To overwrite the file
5	Press (Q)	The Pascal Workstation system displays its Main Command Level prompt line	To exit from the Filer

By booting the Pascal system from the SRM/UX server's file system, client workstation users get automatic access to all of the available volumes in the SRM/UX file system. At this point you have given workstation users complete access to SRM/UX system resources.

There are a few more things you may want to do to make booting-up the Pascal Workstation system more convenient for users. See the next section on "System Customization Options" for details.

System Customization Options

By customizing client-workstation's bootup schemes, you provide users with fast access to the particular SRM/UX volumes and system resources they use most often. For example, at bootup time the Pascal Workstation system can automatically:

- make the user's personal SRM/UX directory the default directory, or make whichever SRM/UX directory the user accesses most frequently the default directory
- load subsystems (FILER, EDITOR, COMPILER, etc.) and other frequently-run programs from an SRM/UX system into a workstation's internal memory so that they run quickly when invoked
- assign unit numbers to a set of frequently-accessed directories on an SRM/UX disc, so that the user need only specify short unit numbers rather than long paths to access those directories

This section explains how to provide workstation users with fast, easy access to essential SRM/UX resources each time they boot their Pascal Workstation system.

Bootup-Time Default SRM/UX Directory

If a workstation user works primarily with the files in a certain SRM/UX directory, there are two ways to configure the Pascal Workstation system to boot-up so that directory is the default directory:

- create an autostart file for the workstation
- modify the workstation's auto-configuration program: *TABLE

Creating an autostart file is the faster, easier way of making this type of configuration change, but it will make boot-up take a few seconds longer than if you modify the TABLE program. But, unless you have other reasons to modify the TABLE program, the slight savings of boot-up time probably doesn't warrant going to the trouble.

Faster Program Execution

To achieve faster program execution, configure the Pascal Workstation system to automatically "P-load" subsystems (FILER, EDITOR, COMPILER, etc.) and/or other frequently-run programs. To "P-load" means to permanently transfer a copy of the subsystem or program from the SRM/UX disc to the client workstation's internal memory. Once the subsystem or program is "P-loaded" in a workstation's internal memory, it runs immediately whenever the user invokes it.

To make this kind of configuration change, create an autostart file for the client workstation. See the instructions later in this section to create autostart files.

Note Be sure that the combined size of the subsystems and/or programs to be loaded does not exceed the capacity of the client workstation's memory.

SRM/UX Directory Access by Unit Number, Rather Than by Path

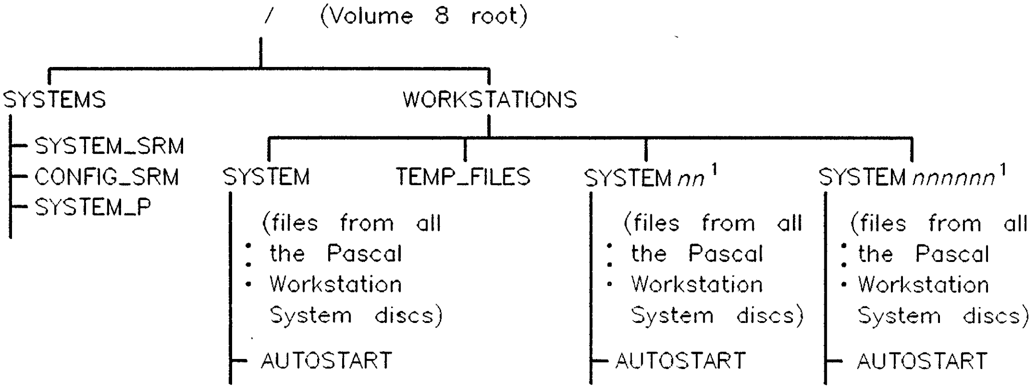
If a workstation user often accesses a number of directories on the same SRM/UX file system, configure the Pascal Workstation system to assign unit numbers to those directories. Then, to access one of those directories, the user need specify only the assigned unit number, instead of the (often lengthy) path to the directory.

To make this type of configuration change, modify the workstation's auto-configuration program: TABLE. Also, your SRM/UX system administrator must make the proper entries in the server's /etc/srmdconf file to allow access to these directories (see chapter 4).

Creating an Autostart File for Each Client Workstation

Using the Pascal Workstation system Editor, create an autostart file for each client workstation on your SRM/UX system. The autostart file must be in the client's own system directory.

The illustration below shows the correct placement of a client workstation's autostart file within the SRM/UX server's volume 8 directory structure:


/ (Volume 8 root)
SYSTEMS
WORKSTATIONS
SYSTEM_SRM
CONFIG_SRM
SYSTEM_P
SYSTEM
TEMP_FILES
SYSTEM*nn*[1]
SYSTEM*nnnnnn*[1]
(files from all the Pascal Workstation System discs)
AUTOSTART
(files from all the Pascal Workstation System discs)
AUTOSTART
(files from all the Pascal Workstation System discs)
AUTOSTART

[1] *nn* and *nnnnnn* represent the two digit Node Address on SRM and the six hex character Link Level Address of the client workstation on LAN

Figure 7-5. Required Location of Pascal Workstation system Autostart file

The Pascal Workstation system treats the contents of an autostart file as if they came directly from a workstation's keyboard. Therefore, an autostart file may contain keystrokes that:

- invoke or exit from Pascal Workstation system subsystems
- initiate or terminate a command
- are a response to a system prompt

Immediately after a workstation boots the Pascal Workstation system, the system executes the contents of the workstation's autostart file (if one exists).

To create an autostart file for each client workstation on your SRM/UX system, follow these steps:

Table 7-25. Create an Autostart file for Each client

Step	Action	Result	Explanation or Note
1	From Main Command Level, press (E)	The Editor prompts: `file?`	To start the Editor
2	Press (Return)	The Editor displays its prompt line and clears the screen	This tells the Editor that you are going to create a new file
3	Press (I)	The Editor's command line disappears	To insert text

Note

Read the rest of this procedure without performing the tasks, to gain an understanding of the procedure flow. Examples for steps 4 through 9 show how to create a custom autostart file for a workstation user named Susan. Susan wants the Pascal Editor and Filer automatically "P-loaded" into her workstation's memory, to allow faster access to those subsystems. She wants the default directory at boot-up time to be her private directory on the SRM/UX file system. Finally, she wants the default directory on another available SRM/UX directory (unit #46:) to be /MARKETING/REPORTS.

Table 7-32.

Step	Action	Result	Explanation or Note
4	Type in the contents of a particular client workstation's autostart file, pressing (Return) to start each line.		

Example

To satisfy Susan's configuration needs, create this autostart file:

```
Line 1:
Line 2:     [7]
Line 3:     P#45:EDITOR.
Line 4:     P#45:FILER.
Line 5:     FP#5:/USERS/SUSAN
Line 6:     U#46:/MARKETING/REPORTS
Line 7:     Q
```

Explanation

- *Line 1* is a blank line, representing the keystroke (Return) in response to the system's date prompt (the first prompt of the boot-up sequence). The Pascal Workstation system gets the correct date from the SRM/UX server, so there is no need to set the date here.
- *Line 2* has an optional 7 entered which sets the time zone for Mountain Standard Time, in response to the system's time prompt. The Pascal Workstation system gets the correct time from the SRM/UX server Computer, so it is not necessary to set the time here.
- *Line 3*, `P` invokes the Permanent (P-load) command from Main Command Level. `#45:EDITOR.` represents the response required to load the Editor from the SRM/UX server's volume 8. Remember to include the trailing period after the subsystem name.
- *Line 4*, `P` invokes the Permanent command from Main Command Level. `#45:FILER.` represents the response required to load the Filer from the SRM/UX server's volume 8. Remember to include the trailing period after the subsystem name.

- *Line 5*, F invokes the Filer from Main Command Level. P invokes the Filer's Prefix command, and #5:/USERS/SUSAN represents the response required to set the default directory to Susan's personal SRM/UX directory.
- *Line 6*, U invokes the Filer's Unit directory command. U#46:/MARKETING/REPORTS represents the response required to set the default directory on unit #46: to REPORTS.
- *Line 7* has a Q to Quit (exit from) the Filer and return to Main Command Level.

Table 7-32.

Step	Action	Result	Explanation or Note
5	Press (Select)	The Editor redisplays its command line	To accept what you typed in
6	Press (Q)	The Editor displays several exit options	To tell the Editor that you want to finish the editing session
7	On your hardware configuration worksheet(s) from chapter 3, look up the node address of the client workstation whose autostart file you are creating.		
8	Press (W)	The Editor prompts for the name of the new file	To write the autostart instructions out to a new file
9	Enter the path shown below.		To store the client workstation's autostart file in the workstation's own system directory. You *must* name the file *AUTOSTART.

Type (including the trailing period):

```
#5:/WORKSTATIONS/SYSTEMnn/AUTOSTART. (Return)
```

where nn is the workstation's two (SRM) or six (LAN) digit node address. (Use a leading 0 (zero) for single-digit node addresses.)

Table 7-33. Store the Client's Autostart File

Example Entry:	Result:
If the node address of Susan's workstation were 4, you would type `#5:/WORKSTATIONS/SYSTEM04 /AUTOSTART.` (Return)	This stores Susan's autostart file in her workstation's own system directory.

Table 7-32.

Step	Action	Result	Explanation or Note
10	Press (E)	The Pascal Workstation system displays its Main Command Level prompt line	To exit from the Editor
11	Repeat steps 1 through 10 as many times as necessary to create a unique autostart file for every client workstation. *Modify the examples in steps 4 and 9 as needed* to suit your SRM/UX user's needs and to accurately reflect the client workstation's node addresses on your SRM/UX system.		It may be easiest to do these steps from each client.

If you do not intend to modify the auto-configuration program for each workstation, you are finished with the start-up procedure for the client Pascal Workstation systems.

Modifying Each Client Workstation's Auto-Configuration Program

Each workstation's auto-configuration program, TABLE, has a source program called CTABLE. The program, CTABLE, is available in every workstation's own system directory.

You may have already modified the copy of CTABLE that is in the /WORKSTATIONS/SYSTEM directory, to give client workstations access to multiple to multiple SRM/UX directories. If so, the source program for each workstation's TABLE program is the modified version of CTABLE, which you named NEWCTABLE, and is now located in the /WORKSTATIONS/SYSTEM directory.

To customize each workstation's TABLE program, you must:

- edit the proper source program
- compile the edited source program
- make the new auto-configuration program available as TABLE in the client workstation's own system directory

The earlier section of this chapter called: "Giving Workstations Access to Multiple SRM/UX Directories" has detailed instructions for customizing the TABLE program. Therefore, this section contains only an overview of how to customize each client-workstation's auto-configuration program. The differences are noted here, in detail, where the instructions differ from those in that section.

To customize a client workstation's auto-configuration program, TABLE, follow these steps:

Table 7-33. Customize a client's TABLE Program

Step	Action

1	*If* you have already modified the client workstation's TABLE program to allow access to multiple SRM/UX directories, edit the source program:
	`#5:/WORKSTATIONS/SYSTEM/NEWCTABLE.TEXT`
	Otherwise, edit the source program:
	`#5:/WORKSTATIONS/SYSTEMnn/CTABLE.TEXT`
	where nn stands for the workstation's two (SRM) or six (LAN) digit node address.

2	If you have *not* already modified the workstation's TABLE program to allow access to multiple SRM/UX directories, add the following compiler directive to the beginning of the TABLE program:
	`$search '#5:/WORKSTATIONS/SYSTEM/INTERFACE. '$`
	This directive belongs on the blank line just above the statement:
	`program {self-configuring} ctable;`

Note

Read the rest of this procedure without performing the tasks, to gain an understanding of the procedure flow. Steps 3 through 6 show how to create a customized TABLE program for a Pascal workstation user named Susan. Susan wants to access one of the SRM/UX system's spooler directories with a unit number (#47:), rather than a path (/PRINTER). She wants the default directory at boot-up time to be her private directory on the SRM/UX file system. Finally, she wants the default directory on another available SRM/UX directory (unit #46:) to be /MARKETING/REPORTS.

Table 7-37.

Step	Action
3	Locate and modify the appropriate section(s) of code to make the Pascal Workstation system configuration most convenient for the user. *Use the following example as a guide.*

Example:

To make the changes Susan requests, change parts of her auto-configuration source program, as follows:

```
{ duplicate entries for prefixing down the SRM }

(*********************************************************************)
(* NOTE: Additional duplicate SRM entries may be assigned here, then   *)
(*  prefixed down below after assigning the temp_unitable. However,    *)
(*  for correct behavior in assigning the system unit, specifically    *)
(*  if booting off the SRM/UX, unit #45 must be assigned AFTER all     *)
(*  the other SRM/UX units have been assigned!                         *)
(*********************************************************************)

with SRM_dav do
 begin
   #  changes made previously to give access to another SRM/UX directory
   tea_srm( 46, sc, ba, 1);  {free}

   #  Add this line to assign an additional unit number to the SRM/UX system,
   #  (also assigned unit #45:)
   tea_srm( 47, sc, ba, du);

   tea_srm( 45, sc, ba, du); {for possible use as the system unit}
  end; {with}


   .
   .
   .
{ prefix the primary and secondary SRM/UX unit entries }

   # Auto-configuration program makes the directory referenced by unit #5:
   # the default directory. Add the correct path name to Susan's private
   # SRM/UX directory.

if not unit_prefix_successful('#5:/USERS/SUSAN')  then
 {do nothing};{tries to set up uvid for possible default unit assignment below}
```

```
   # Add the path name to access-time default directory for the other
   # available SRM/UX directory.

if not unit_prefix_successful('#46:/MARKETING/REPORTS')
        then zap_assigned_unit(46);
        {free}

   # Add the next line to allow the Pascal Workstation system to
   # recognize the additional unit number you assigned to
   # the SRM/UX volume 8 directory.

   # Also add the path name to of the directory which this unit
   # number is to reference.

if not unit_prefix_successful('#47:/PRINTER')
   then zap_assigned_unit(47);

if not unit_prefix_successful('#45:'+srmsysprefix+srmnode(unitable^[45].sc))
   then if not unit_prefix_successful('#45:'+srmsysprefix)
     then zap_assigned_unit(45);
```

Table 7-37.

Step	Action
4	Choose a name other than CTABLE for the edited source program and write it out to the client workstation's own system directory. Then exit the Editor. *Example:* If you decided to call Susan's edited source program CTABLE.TEXT, and the node address of her workstation was 9, write her source program to the file: `#5:/WORKSTATIONS/SYSTEM09/CUSTOMCTABLE.TEXT`
5	Compile, run, and verify the new auto-configuration program. (The compiled program may also be in the workstation's own system directory.) *Example:* If you compiled Susan's auto-configuration source program, CUSTOMCTABLE.TEXT, the compiled version would be: `#5:/WORKSTATIONS/SYSTEM09/CUSTOMCTABLE.CODE`

6	Copy the compiled auto_configuration program to the file name TABLE in the client workstation's own system directory. (You must *remove*—not overwrite—the workstation's original TABLE program at the Filer prompt, since this TABLE program is unique to the user.)
	Example:
	To specify the filecopy for Susan's customized auto- configuration program, type:
	`#5:/WORKSTATIONS/SYSTEM09/CUSTOMCTABLE.CODE,`
	`#5:/WORKSTATIONS/SYSTEM09/TABLE`

7	Repeat steps 1 through 6 as many times as necessary to customize every workstation's auto-configuration program: TABLE. *Modify the examples in steps 2 through 5 as needed* to suit your SRM/UX user's needs.

You are now finished with the start-up procedure for the Pascal Workstation systems on the SRM/UX system. *Every* client workstation with a BootROM version 3.0 or later (but not 3.0L) will have the ability to automatically boot the Pascal Workstation system from the SRM/UX server's file system. After the client workstations boot-up, they will customize their own configurations according to their resident auto_configuration programs and/or autostart files.

8

Migrating Applications and Data from SRM to SRM/UX Using srmdrestore(1)

The *srmdrestore*(1) command can be used to read SRM backup tapes onto the HP-UX file system on your SRM/UX server. It is the easiest way to move large numbers of files from SRM to SRM/UX.

You should use *srmdrestore*(1) AFTER your SRM/UX server is installed and configured, so that the emulated SRM volumes you will be restoring to will already be set up.

There are two restrictions you must be aware of before using *srmdrestore*(1):

- *Srmdrestore*(1) can only read SRM backup tapes from SRM version 3.0 or later. Backup tapes from previous SRM versions have a completely different format.
- Duplinked files are not allowed to appear on different SRM/UX file system devices; that is, *srmdrestore*(1) will not restore duplinked files from your SRM backup tape if you are trying to put them on separate disc drives. However, if you are restoring duplinked files to an emulated SRM volume in SRM/UX that does not itself contain any local or NFS mount points, then *srmdrestore*(1) will successfully restore those duplinked files, even if the emulated SRM volume's `Root Directory` is NFS mounted on your SRM/UX server's file system.

To bring your SRM files over to SRM/UX, first make a backup tape of the SRM file system. Install that tape in your SRM/UX server's tape drive. Do a "catalog" on the tape to see what files it contains (if you are not sure) using the `-t` option to *srmdrestore*(1). Finally, restore the files you want on your SRM/UX system by executing the SRM/UX command *srmdrestore*(1) with the appropriate options (see the table below).

The srmdrestore Command Options

Table 8-1. Options to the srmdrestore Command

Option	What it does ...
-d	Create directories as needed. If a directory, which appears as a component of the path name of the file to be restored, does not exist: create that directory. For example: Srmdrestore, upon restoring the file named "./srmuxroot/RMB/FILE.TEXT", will create the directories "./srmuxroot" and "./srmuxroot/RMB", if they do not already exist.
-m	Retain the original modification dates and times on the files. Normally, the current date and time will be used when the file is restored. If this option is specified, the date the file was last modified (as determined from the tape) will be restored, instead. This option has no effect on directories as they are restored.
-v	Print additional information. (Verbose mode) Srmdrestore prints the full path name to stdout *after* each file is restored. If combined with the -t option, -v causes the catalog to print additional information. If you abort srmdrestore (for example: with a CTRL-C), then all of the file names that have been printed have been completely restored. This fact may help you to know the state of the restore process, i.e., which files have been restored and which haven't.
-u	Run in unconditional mode. Normally, srmdrestore will only overwrite files on the disc which are older than the ones on the tape. If the -u option is specified, restored files will replace existing files on the disc, regardless of which files are newer.

Table 8-1. Options to the srmdrestore Command (continued)

Option	What it does ...
-t	Catalog the tape. Do not restore any files. Just print a catalog of the tape. Print the catalog in the identical format that SRM uses. When combined, options -t and -v print the address of the file on the tape, flags indicating information about the file, and the tape of a target duplink.
-M	Set permission mask. (umask) Normally, a restored file will have its permissions set to 666 (octal) if it's a regular file and 777 (octal) if it's a directory (as modified by the current umask). This option allows you to set the restored file's umask to some non-default value. The "execute" bit will always be cleared for non-directory files being restored. For example: "srmdrestore -M 0700" will force all directories to be restored with permission mode 0700 and all regular files to be restored with permission mode 0600.

Note The examples in this chapter use /dev/rct as the "generic" device file for the cartridge tape device on the SRM/UX server. Check with your HP-UX System Administrator before executing *srmdrestore*(1) if you don't know the name of the appropriate device file on your SRM/UX server.

Cataloging a Tape

You can generate a catalog of files that are on a tape with the command:

```
srmdrestore -t /dev/rct
```

The srmdrestore output would look something like this:

```
Volume Backed Up:    "DISC1           "
Backup Media Number: 1
Backup File:         "dave            "
Backup Date:          7 Aug 90  11:58:36 AM
Backup Type:         Copy
Backup Log File:     "BKP080790115836 "

Tree SRM              Creation  Last Mod  Size
Lvl  Pathname         Date      Date      (bytes)
---- ---------------- --------- --------- --------
  0  .                18 Apr 89 26 Jul 90      ---
  1  USERS             8 Nov 84 16 Jul 90      ---
  2  BASIC6           12 Sep 88 12 Jul 90      ---
  3  B22              16 May 90 17 May 90      ---
  4  AM               16 May 90 17 May 90      ---
  5  AMD.TEXT         26 Apr 90  7 Jun 90     9216
  5  AMDVR.TEXT       26 Apr 90  7 Jun 90    27648
  5  AMEX.TEXT        26 Apr 90  7 Jun 90     5120
  5  AMHDR.CODE       17 May 90  7 Jun 90     3072
```

You could place this catalog in a file named, for example: `tape1.cat` by executing:

```
srmdrestore -t /dev/rct >tape1.cat
```

Restoring Your Files

Files will be restored to the directory you are in when you execute *srmdrestore*(1).

Temporary Files

TM# Files

When restoring files, *srmdrestore*(1) will not overwrite a file on disc until the entire file has been read from the tape. *Srmdrestore*(1) reads each file into a temporary file with the same name as the target file, but with the prefix "TM#". Thus, if you abort a restore in the middle, the only effect is to leave a temporary file; the file being restored is not created. Once the file has been completely read from the tape, the temporary file is renamed to the target file name.

Duplink Files

When files that have links to other files are restored, *srmdrestore*(1) creates a directory that contains temporary files (temporary files are symbolic links to duplink files). This directory is created in the current directory with the name `BKmmddyyhhmmss`, where mmddyyhhmmss is the month, day, year, hour, minute and second of the backup. To find out what exactly this name is, look at the "Backup Log File" field of the tape catalog (generated when you run *srmdrestore*(1) with the -t option). This directory contains duplinked files with names of the form: `DL#nnnn` where `nnnn` is a decimal integer (e.g., 7324). This directory and its contents are removed when *srmdrestore*(1) reaches the end of the last tape being restored.

For example, for the tape listed in the previous catalog, a duplink could be named: `BK080790115836/DL#7324` and the temporary file for `AMD.TEXT` would be named: `USERS/BASIC6/B22/AM/TM#AMD.TEXT`.

Multiple Tape Restores

The *srmdrestore*(1) command is designed to allow some flexibility in how you restore files from multiple backup tapes.

In the simplest case, *srmdrestore*(1) will issue the prompt:

```
srmdrestore: type next tape device file name (Return=file_name) >
```

when it encounters the end of each tape (except the last tape). When you get the above prompt, all you need to do is insert a new tape and hit Return after the tape has completed loading.

If you have multiple tape drives, you may type the name of the device file for another tape unit in response to the prompt.

If an error occurs, or the command is aborted at this point, all you have to do to continue is to correct the error and then reexecute the *srmdrestore*(1) command.

You can restore tapes out of sequence, with the following restrictions:

- A file that is duplinked to another file on an earlier tape will not be restored, if the earlier tape has not been restored. *Srmdrestore*(1) issues a warning (to stderr) for each such duplinked file.

  To determine the files that will be affected, use the "-tv" option to catalog the tape. Files which have the letter "`l`" in the "`Flags`" field are duplinked files. The number following the "`l`" indicates the number of the tape that contains the first instance of the duplinked file. If that tape number has already been restored, or is the current tape, then the duplinked file will be restored.

- A file that crosses a tape boundary will not be restored if the previous tape has not been restored. *Srmdrestore*(1) issues a warning (to stderr) for each file that crosses a tape boundary.

  You can determine if a file crosses a tape boundary by examining the tape catalog (generated when you run *srmdrestore*(1) with the -t option). The files that cross tape boundaries will always be either the first or last file on the tape, and will be indicated by a * following the date field of that file's entry.

- All other files will be restored, even if the tapes are inserted out of sequence. This *srmdrestore*(1) feature allows you to recover selective files from the middle of a multiple backup-tape-set.

Error Logging

All errors in the restore process will be logged to `stderr`. Redirect these messages to a file if you wish to examine them later.

To redirect the error messages to a file from sh (or ksh) execute:

```
srmdrestore -dvm /dev/rct >restored.files 2>notrestored.files
```

This command sequence will log all errors to the file named `notrestored.files` and will log all successfully restored file names to the file named `restored.files`.

Any tape change requests will be logged only to your terminal, not to stderr.

Pattern Matching

To selectively restore individual files, specify their full path names in the *srmdrestore*(1) command line, exactly as they would appear in the catalog listing. You can specify wildcards to match multiple file names from the tapes. These wildcards function just like the wildcards in the HP-UX "sh" shell do. (See regexp(7) in the *HP-UX Reference* manual for details.)

Restore Examples

Example 1

Execute the command:

```
srmdrestore -dvm /dev/rct './USERS/BASIC6/B22/AM/AMEX.TEXT' './USERS/BASIC6/B22/AM/AMD.TEXT'
```

This command will restore two files. The single quote characters are used to prevent the HP-UX shell from incorrectly interpreting characters in the file name.

Example 2

Execute the command:

```
srmdrestore -dvm /dev/rct '*.TEXT'
```

This command will restore all of the files (in any directory) that have names ending in ".TEXT".

Example 3

If you cataloged a tape by executing the command:

```
srmdrestore -t /dev/rct
```

and got this output:

```
Volume Backed Up:    "DISC1           "
Backup Media Number: 1
Backup File:         "dave            "
Backup Date:          7 Aug 90  11:58:36 AM
Backup Type:         Copy
Backup Log File:     "BKP080790115836 "

Tree SRM              Creation  Last Mod  Size
Lvl  Pathname         Date      Date      (bytes)
---- ---------------- --------- --------- --------
  0  .                18 Apr 89 26 Jul 90      ---
  1  USERS             8 Nov 84 16 Jul 90      ---
  2  BASIC6           12 Sep 88 12 Jul 90      ---
  3  B22              16 May 90 17 May 90      ---
  4  AM               16 May 90 17 May 90      ---
  5  AMD.TEXT         26 Apr 90  7 Jun 90     9216
  5  AMDVR.TEXT       26 Apr 90  7 Jun 90    27648
  5  AMEX.TEXT        26 Apr 90  7 Jun 90     5120
  5  AMHDR.CODE       17 May 90  7 Jun 90     3072
```

and you wanted to restore this tape to the directory `/srmuxroot`, you would execute the commands:

```
cd /srmuxroot
srmdrestore -dvm /dev/rct >restored.filelist
```

The path name of each file restored is printed to the file `restored.filelist` after it has been restored.

After you executed the above srmdrestore command, the first file restored would be:

```
    ./USERS/BASIC6/B22/AM/AMD.TEXT
```

and it would have the modification date and time it had on the tape.

Srmdrestore(1) would create the directories: `./USERS`, `./USERS/BASIC6`, `./USERS/BASIC6/B22`, and `./USERS/BASIC6/B22/AM` before restoring the file `AMD.TEXT`, if the directories didn't already exist.

9

SRM/UX System Examples

In all of the following examples, and any time you set up your real SRM/UX system, we recommend setting up a simple, single-server, single-client system first (as shown in Example 1 here). Then after you have set up the simple system, you can proceed to bring up the entire SRM/UX system more easily and successfully.

Example 1—A Simple SRM/UX System

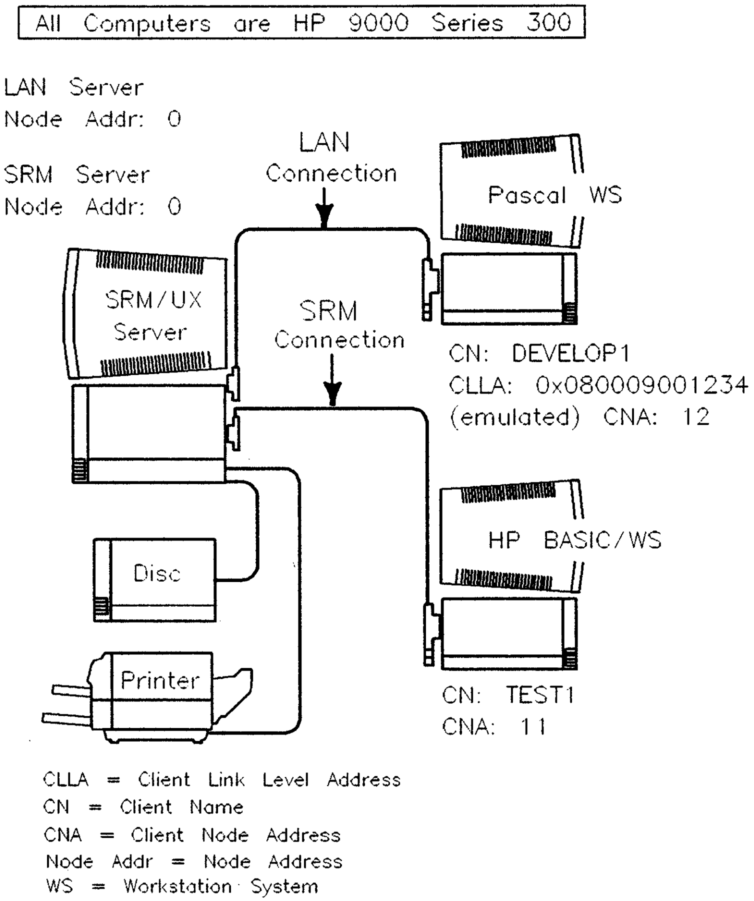


Figure 9-1. Example 1—A Simple SRM/UX System

To establish this simple SRM/UX system, follow these step-by-step instructions:

1. Set up the system hardware by following the instructions in chapter 3 of this manual, and the instructions in each of the component's manuals.
2. Photocopy the "SRM/UX Client Descriptions" worksheet , the "SRM/UX LAN Worksheet", and the "SRM/UX SRM Worksheet" in chapter 2.

3. Fill out the `SRM/UX Client Descriptions` worksheet as follows:

Table 9-1. Example 1—SRM/UX Client Descriptions

SERVER NAME: HUBO			
Client Name	Comments	Connection Type	Volume List
DEVELOP1	Susan H. ext. 501 Bldg. C hp9000s370	LAN	SRMUXROOT
TEST1	Prod. line Post 5. Joe hp9000s340	SRM	SRMUXROOT

4. Fill out the "SRM/UX SRM Worksheet" as follows:

Table 9-2. Example 1—SRM/UX SRM Worksheet

SERVER NAME: HUBO						
Client Name	Server Node Address[1]	Server's SRM Device File	Client Node Address	Uid	Gid	Umask
TEST1	0	/dev/srm	11	105	9	022

[1] This field is not entered in the /etc/srmdconf SRM-CLIENTS table.

5. Fill out the "SRM/UX LAN Worksheet" as follows:

Table 9-3. Example 1—SRM/UX LAN Worksheet

SERVER NAME: HUBO							
Client Name	Client Link Level Address	Client Internet Address	Client Node Address[1]	Server Node Address	Uid	Gid	Umask
DEVELOP1	0x080009001234	15.99.99.22	12	0	101	9	022

[1] The Client Node Address and Server Node Address are emulated on the LAN connection

6. Ensure that the HP-UX System Administrator has HP-UX installed, configured, and running on the server computer.
7. Install the SRM/UX server software by following the instructions in chapter 4 (and chapter 5 for installing the ability for clients to boot from the server) of this manual.
8. Configure the server's HP-UX environment for SRM/UX by following the instructions in chapter 4 of this manual. This involves reconfiguring the HP-UX kernel and creating an SRM device file.

9. Edit your server's /etc/srmdconf file to appear as follows:

```
# Long lines in this example will wrap in this printed output.
VOLUME-TABLE

# Volume Descriptions
#
# Volume         |Volume|     |     |Temp     |Root
# Name           |Number| Uid | Gid |Directory|Directory
#----------------+------+-----+-----+---------+----------------------------------
SRMUXROOT        :   8  :     :     :tmp      :/srmuxroot

LAN-CLIENTS

# LAN Client Descriptions
#
# Client       |Client         |Client|Server|          |     |     |     |
# Link Level   |Internet       |Node  |Node  |Client    |     |     |     |Volume
# Address      |Address        |Addr  |Addr  |Name      | Uid | Gid |Umask|List
#--------------+---------------+------+------+----------+-----+-----+-----+--------
0x080009001234 :  15.99.99.22  :  12  :   0  :DEVELOP1  : 101 :  9  : 022 :SRMUXROOT
# Susan's hp9000s370

SRM-CLIENTS

# SRM Client Descriptions
#
#                |Client|          |     |     |     |
# Server's SRM   |Node  |Client    |     |     |     |Volume
# Device File    |Addr  |Name      | Uid | Gid |Umask|List
#----------------+------+----------+-----+-----+-----+-----------------------------
/dev/srm         :  11  : TEST1    : 105 :  9  : 022 : SRMUXROOT #Joe's hp9000s340

SPOOL-ENVIRONMENT

# The following environment variables are pre-initialized for spooling
# processes.
#
#       SHELL                   - defaults to /bin/sh, unless overridden
#       TZ                      - inherited from the OS, if it is set there,
#                                 otherwise defaults to MST7MDT
#       SRMSPOOLFILENAME        - defaults to the full pathname to the file
#                                 being spooled (e.g. /srmuxroot/PRINTER/xyzzy)
#       PATH                    - defaults to /bin:/usr/bin. Recommend that
#                                 it be set to include paths needed for all
#                                 commands used in Spool Commands
#----------------------------------------------------------------------------------
PATH=/bin:/usr/bin:             # srmdlpfilter is in /usr/bin

SPOOL-TABLE

# Spooling Commands
#
# Spool                 |        |Spool
# Directory             |Priority|Command
#----------------------+--------+------------------------------------------------
/srmuxroot/PRINTER      :    2   : srmdlpfilter | lp $PRIORITY
```

10. Run *srmdconfcheck*(1M) to validate the above srmdconf file.

11. Bring up the server daemon, srmd, by executing:

```
/etc/srmd
```

See the "Starting the srmd Process on the Server" section in chapter 4 for detailed instructions.

12. Bring up your HP BASIC/WS client on the SRM connection by following the instructions on booting the first client in chapter 5 of this manual.
13. Bring up your Pascal Workstation System client on the LAN connection by following the instructions on booting the first client in chapter 5 of this manual.

You are now done bringing up the simple SRM/UX system. You verified the client booting environment in the steps above. Next, move around in Volume 8 of the SRM/UX file system from both clients to verify correct operation. And, finally, try spooling some files to test the SRM/UX spooling environment.

Example 2—Multiple Clients on SRM and LAN Connections

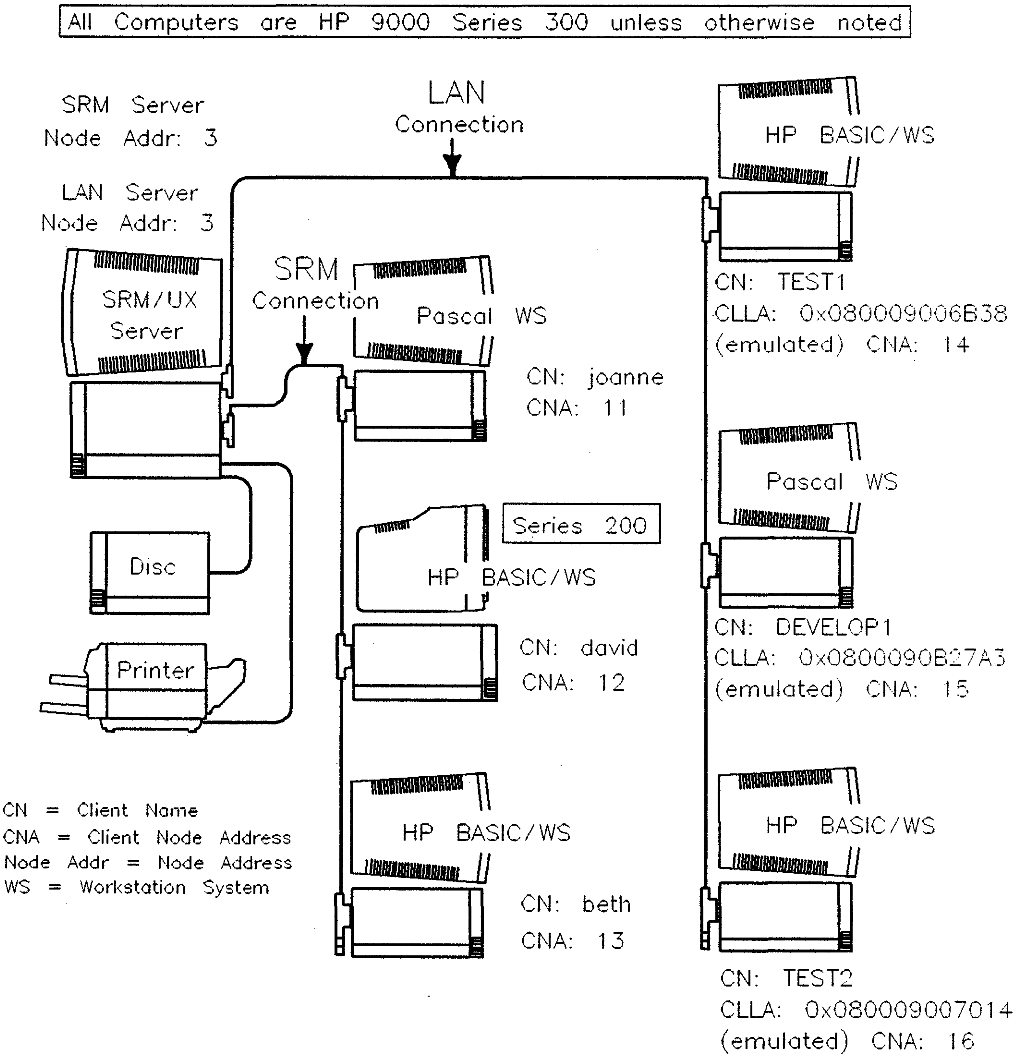


Figure 9-2. Example 2—Multiple Clients on SRM and LAN

To establish this SRM/UX system, follow these step-by-step instructions:

1. Set up the system hardware by following the instructions in chapter 3 of this manual, and the instructions in each of the component's manuals.
2. Photocopy the "SRM/UX Client Descriptions" worksheet, the "SRM/UX LAN Worksheet", and the "SRM/UX SRM Worksheet" in chapter 2.
3. Fill out the `SRM/UX Client Descriptions` worksheet as follows:

Table 9-4. Example 2—SRM/UX Client Descriptions

SERVER NAME: HUB3			
Client Name	**Comments**	**Connection Type**	**Volume List**
joanne	Joanne F. hp9000s340	SRM	SRMUXROOT, HPUXROOT, JOANNE
david	David B. hp9000s226	SRM	SRMUXROOT, DAVID
beth	Beth R. hp9000s370	SRM	SRMUXROOT, BETH
TEST1	John B hp9000s370	LAN	SRMUXROOT, JOHNB
DEVELOP1	Robert B hp9000s340	LAN	SRMUXROOT, HPUXROOT, ROBERTB
TEST2	Joe C. hp9000s370	LAN	SRMUXROOT, JOEC

4. Fill out the "SRM/UX SRM Worksheet" as follows:

Table 9-5. Example 2—SRM/UX SRM Worksheet

SERVER NAME: HUB3						
Client Name	Server Node Address[1]	Server's SRM Device File	Client Node Address	Uid	Gid	Umask
joanne	3	/dev/srm	11	107	9	022
david	3	/dev/srm	12	109	9	022
beth	3	/dev/srm	13	111	9	022

[1] This field is not entered in the /etc/srmdconf SRM-CLIENTS table.

5. Fill out the "SRM/UX LAN Worksheet" as follows:

Table 9-6. Example 2—SRM/UX LAN Worksheet

SERVER NAME: HUB3							
Client Name	Client Link Level Address	Client Internet Address	Client Node Address[1]	Server Node Address	Uid	Gid	Umask
TEST1	0x080009006B38	15.99.99.28	14	3	102	9	022
DEVELOP1	0x0800090B27A3	15.99.99.30	15	3	101	9	022
TEST2	0x080009007014	15.99.99.32	16	3	105	9	022

[1] The Client Node Address and Server Node Address are emulated on the LAN connection

6. Ensure that the HP-UX System Administrator has HP-UX installed, configured, and running on the server computer.
7. Install the SRM/UX server software by following the instructions in chapter 4 (and chapter 5 for installing the ability for clients to boot from the server) of this manual.
8. Configure the server's HP-UX environment for SRM/UX by following the instructions in chapter 4 of this manual. This involves reconfiguring the HP-UX kernel and creating an SRM device file.

9. Edit your server's /etc/srmdconf file to appear as follows:

```
# Long lines in this example will wrap in this printed output.
VOLUME-TABLE

# Volume Descriptions
#
# Volume         |Volume|     |     |Temp     |Root
# Name           |Number| Uid | Gid |Directory|Directory
#----------------+------+-----+-----+---------+---------------------------------
SRMUXROOT        :   8  :     :     : tmp     :/srmuxroot
HPUXROOT         :   9  :     :     : tmp     : /
JOHNB            :  10  :     :     : tmp     : /users/johnb
ROBERTB          :  10  :     :     : tmp     : /users/robertb
JOEC             :  10  :     :     : tmp     : /users/joec
JOANNE           :  10  :     :     : tmp     : /users/joanne
DAVID            :  10  :     :     : tmp     : /users/david
BETH             :  10  :     :     : tmp     : /users/beth

LAN-CLIENTS

# LAN Client Descriptions
#
# Client       |Client         |Client|Server|          |     |     |     |
# Link Level   |Internet       |Node  |Node  |Client    |     |     |     |Volume
# Address      |Address        |Addr  |Addr  |Name      | Uid | Gid |Umask|List
#--------------+---------------+------+------+----------+-----+-----+-----+--------
0x080009006B38 :  15.99.99.28  :  14  :   3  :  TEST1   : 102 :   9 : 022 :SRMUXROOT,
JOHNB  # John B's hp9000s370
0x0800090B27A3 :  15.99.99.30  :  15  :   3  : DEVELOP1 : 101 :   9 : 022 :SRMUXROOT,
HPUXROOT,ROBERTB  # Robert B's hp9000s340
0x080009007014 :  15.99.99.32  :  16  :   3  :  TEST2   : 105 :   9 : 022 :SRMUXROOT,
JOEC  # Joe C's hp9000s370

SRM-CLIENTS

# SRM Client Descriptions
#
#                |Client|          |     |     |     |
# Server's SRM   |Node  |Client    |     |     |     |Volume
# Device File    |Addr  |Name      | Uid | Gid |Umask|List
#----------------+------+----------+-----+-----+-----+-----------------------------
/dev/srm         :  11  : joanne   : 107 :  9  : 022 :SRMUXROOT,HPUXROOT,JOANNE
# Joanne F's hp9000s340
/dev/srm         :  12  : david    : 109 :  9  : 022 :SRMUXROOT,DAVID # David B's hp9000s226
/dev/srm         :  13  : beth     : 111 :  9  : 022 :SRMUXROOT,BETH # Beth R's
 hp9000s370
```

```
SPOOL-ENVIRONMENT

# The following environment variables are pre-initialized for spooling
# processes.
#
#       SHELL                   - defaults to /bin/sh, unless overridden
#       TZ                      - inherited from the OS, if it is set there,
#                                 otherwise defaults to MST7MDT
#       SRMSPOOLFILENAME        - defaults to the full pathname to the file
#                                 being spooled (e.g. /srmuxroot/PRINTER/xyzzy)
#       PATH                    - defaults to /bin:/usr/bin. Recommend that
#                                 it be set to include paths needed for all
#                                 commands used in Spool Commands.
#---------------------------------------------------------------------------------
PATH=/bin:/usr/bin:             # srmdlpfilter is in /usr/bin

SPOOL-TABLE

# Spooling Commands
#
# Spool                |        |Spool
# Directory            |Priority|Command
#----------------------+--------+--------------------------------------------------
/srmuxroot/PRINTER     :    2   : srmdlpfilter | lp $PRIORITY
```

10. Run *srmdconfcheck*(1M) to validate the above srmdconf file.

11. Create the Volume 10 directories that you have entered into the VOLUME-TABLE above, i.e., those for JOHNB, ROBERTB, JOEC, JOANNE, DAVID, and BETH.

Note We recommend that you create a `tmp` directory in the `Root Directory` of each Volume that appears in the VOLUME-TABLE of the /etc/srmdconf file below (e.g., for Volume JOHNB, there would be a /users/johnb/tmp directory).

You may instead choose to put all `Temp Directories` in a common location, such as /tmp, as long as each `Temp Directory` is under the same HP-UX mount point as the `Root Directory` of its Volume.

---

12. Bring up the server daemon, srmd, by executing:

    `/etc/srmd`

    See the "Starting the srmd Process on the Server" section in chapter 4 for detailed instructions.

13. Bring up the first HP BASIC/WS client on the SRM connection by following the instructions on booting the first client in chapter 5 of this manual.
14. Bring up the first Pascal Workstation System client on the SRM connection by following the instructions on booting the first client in chapter 5 of this manual.
15. Bring up the first HP BASIC/WS client on the LAN connection by following the instructions on booting the first client in chapter 5 of this manual.
16. Bring up the first Pascal Workstation System client on the LAN connection by following the instructions on booting the first client in chapter 5 of this manual.
17. Bring up all of the remaining clients on both the SRM and LAN connection by following the instructions in the "Adding Clients to the Booting Environment" section of chapter 5 of this manual.

You are now done bringing up the Example 2 SRM/UX system. You verified the client booting environment in the steps above. Next, move around in various volumes of the SRM/UX file system from all of the clients to verify correct operation. And, finally, try spooling some files from each client to test the SRM/UX spooling environment.

Example 3—A Multiple Server, Multiple Client, Multiple Connections System

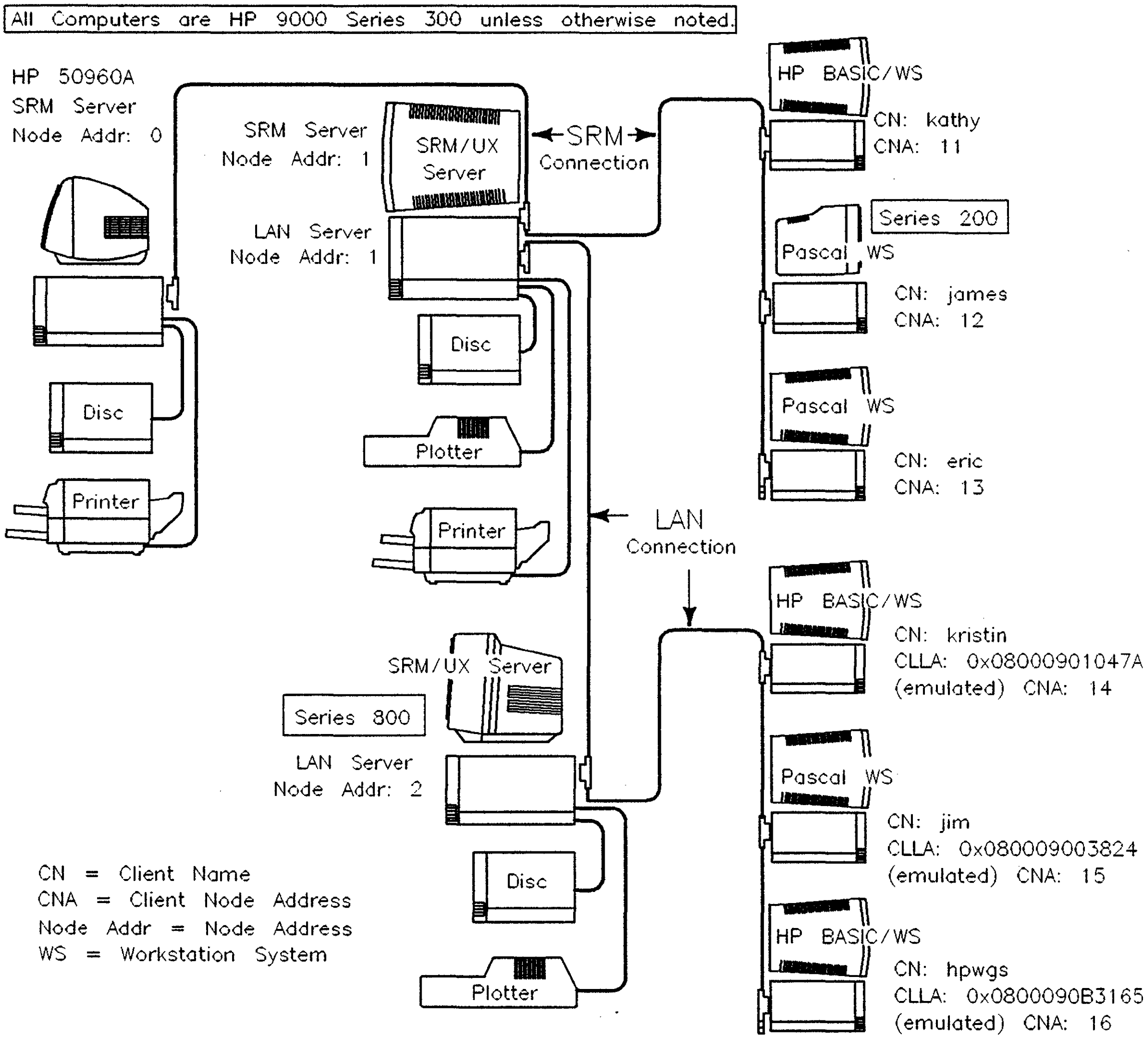


Figure 9-3. Example 3—Multiple Servers, Multiple Clients on SRM and LAN

To establish this SRM/UX system, follow the step-by-step instructions provided in the sections below. Each section contains the procedures for configuring one of the servers. Even though the SRM/UX system contains multiple servers, you can bring up the system in discrete subsystem units. There is nothing special that needs to be done to have all of the subsystems work together as a complete SRM/UX system, except for the hardware interconnections.

Configure the SRM Server

The SRM server is used in this example to show that you can use your SRM server with SRM/UX servers in an overall system. The SRM server should already have been set up by following the instructions in the *Shared Resource Management System Manager's Guide*, *Shared Resource Management Hardware Installation Manual*, and *Shared Resource Management Software Installation Manual*.

Configure the Series 300 Server

To establish this SRM/UX system, follow these step-by-step instructions:

1. Set up the system hardware by following the instructions in chapter 3 of this manual, and the instructions in each of the component's manuals.
2. Photocopy the "SRM/UX Client Descriptions" worksheet, the "SRM/UX LAN Worksheet", and the "SRM/UX SRM Worksheet" in chapter 2.

3. Fill out the SRM/UX Client Descriptions worksheet for the Series 300 server (at Server Node Address 1) as follows:

Table 9-7.
Example 3—SRM/UX Client Descriptions (Series 300 server)

SERVER NAME: HUB1			
Client Name	**Comments**	**Connection Type**	**Volume List**
kathy	BASIC/WS Kathy W. s340 ext.533	SRM	SRMUXROOT, KATHY
james	PAWS, James's s236 ext.229	SRM	SRMUXROOT, JAMES
eric	PAWS, Eric M's s310 Bldg. 3 ext.113	SRM	SRMUXROOT, HPUXROOT, ERIC
kristin	BASIC/WS Kristin W. s370 ext.191	LAN	SRMUXROOT, KRISTIN
jim	PAWS Jim H's s340 ext.212	LAN	SRMUXROOT, JIM
hpwgs	BASIC/WS Fred G. Bldg.2	LAN	SRMUXROOT, HPUXROOT, HPWGS

4. Fill out the "SRM/UX SRM Worksheet" for the Series 300 server (at Server Node Address 1) as follows:

Table 9-8.
Example 3—SRM/UX SRM Worksheet (Series 300 server)

SERVER NAME: HUB1						
Client Name	**Server Node Address[1]**	**Server's SRM Device File**	**Client Node Address**	**Uid**	**Gid**	**Umask**
kathy	1	/dev/srm	11	119	9	022
james	1	/dev/srm	12	121	9	022
eric	1	/dev/srm	13	123	9	022

[1] This field is not entered in the /etc/srmdconf SRM-CLIENTS table.

5. Fill out the "SRM/UX LAN Worksheet" for the Series 300 server (at `Server Node Address` 1) as follows:

Table 9-9. Example 3—SRM/UX LAN Worksheet (Series 300 server)

SERVER NAME: HUB1							
Client Name	**Client Link Level Address**	**Client Internet Address**	**Client Node Address**[1]	**Server Node Address**	**Uid**	**Gid**	**Umask**
kristin	0x08000901047A	15.99.99.33	14	1	111	9	022
jim	0x080009003824	15.99.99.35	15	1	113	9	022
hpwgs	0x0800090B3165	15.99.99.37	16	1	115	9	022

[1] The Client Node Address and Server Node Address are emulated on the LAN connection

6. Ensure that the HP-UX System Administrator has HP-UX installed, configured, and running on the Series 300 server computer.
7. Install the SRM/UX server software on the Series 300 server by following the instructions in chapter 4 (and chapter 5 for installing the ability for clients to boot from the server) of this manual.
8. Configure the Series 300 server's HP-UX environment for SRM/UX by following the instructions in chapter 4 of this manual. This involves reconfiguring thė HP-UX kernel and creating an SRM device file.

9. Edit your Series 300 server's /etc/srmdconf file to appear as follows:

```
# Long lines in this example will wrap in this printed output.
VOLUME-TABLE

# Volume Descriptions
#
# Volume         |Volume|     |     |Temp     |Root
# Name           |Number| Uid | Gid |Directory|Directory
#----------------+------+-----+-----+---------+---------------------------------
SRMUXROOT        :   8  :     :     :  tmp    : /srmuxroot
HPUXROOT         :   9  :     :     :  tmp    : /
KATHY            :  10  :     :     :  tmp    : /users/kathy
JAMES            :  10  :     :     :  tmp    : /users/james
ERIC             :  10  :     :     :  tmp    : /users/eric
KRISTIN          :  10  :     :     :  tmp    : /users/kristin
JIM              :  10  :     :     :  tmp    : /users/jim
HPWGS            :  10  :     :     :  tmp    : /users/hpwgs

LAN-CLIENTS

# LAN Client Descriptions
#
# Client       |Client         |Client|Server|          |     |     |     |
# Link Level   |Internet       |Node  |Node  |Client    |     |     |     |Volume
# Address      |Address        |Addr  |Addr  |Name      | Uid | Gid |Umask|List
#--------------+---------------+------+------+----------+-----+-----+-----+--------
0x08000901047A : 15.99.99.33   :  14  :   1  :  kristin : 111 :   9 : 022 :SRMUXROOT,
KRISTIN  # Kristin W's hp9000s370
0x080009003824 : 15.99.99.35   :  15  :   1  :  jim     : 113 :   9 : 022 :SRMUXROOT,
JIM      # Jim's  hp9000s340
0x0800090B3165 : 15.99.99.37   :  16  :   1  :  hpwgs   : 115 :   9 : 022 :SRMUXROOT,
HPUXROOT, HPWGS   # Fred's hp9000s370

SRM-CLIENTS

# SRM Client Descriptions
#
#                |Client|          |     |     |     |
# Server's SRM   |Node  |Client    |     |     |     |Volume
# Device File    |Addr  |Name      | Uid | Gid |Umask|List
#----------------+------+----------+-----+-----+-----+-----------------------------
/dev/srm         :  11  : kathy    : 119 :  9  : 022 : SRMUXROOT,KATHY # Kathy W's s340
/dev/srm         :  12  : james    : 121 :  9  : 022 : SRMUXROOT,JAMES # James's  s236
/dev/srm         :  13  : eric     : 123 :  9  : 022 : SRMUXROOT,HPUXROOT,ERIC # Eric M's s310, Bldg. 3
```

```
SPOOL-ENVIRONMENT

# The following environment variables are pre-initialized for spooling
# processes.
#
#       SHELL                   - defaults to /bin/sh, unless overridden
#       TZ                      - inherited from the OS, if it is set there,
#                                 otherwise defaults to MST7MDT
#       SRMSPOOLFILENAME        - defaults to the full pathname to the file
#                                 being spooled (e.g. /srmuxroot/PRINTER/xyzzy)
#       PATH                    - defaults to /bin:/usr/bin. Recommend that
#                                 it be set to include paths needed for all
#                                 commands used in Spool Commands.
#----------------------------------------------------------------------------
PATH=/bin:/usr/bin:             # srmdlpfilter is in /usr/bin

SPOOL-TABLE

# Spooling Commands
#
# Spool                |        |Spool
# Directory            |Priority|Command
#----------------------+--------+--------------------------------------------
/srmuxroot/PRINTER     :    2   : srmdlpfilter | lp $PRIORITY
```

10. Run *srmdconfcheck*(1M) to validate the above srmdconf file.

11. Create the Volume 10 directories that you have entered into the VOLUME-TABLE above, i.e., those for KATHY, JAMES, ERIC, KRISTIN, JIM, HPWGS.

Note

We recommend that you create a `tmp` directory in the `Root Directory` of each Volume that appears in the VOLUME-TABLE of the /etc/srmdconf file below (e.g., for Volume KATHY, there would be a /users/kathy/tmp directory).

You may instead choose to put all `Temp Directories` in a common location, such as /tmp, as long as each `Temp Directory` is under the same HP-UX mount point as the `Root Directory` of its Volume.

---

12. Bring up the server daemon, *srmd*(1M), by executing:

    `/etc/srmd`

    See the "Starting the srmd Process on the Server" section in chapter 4 for detailed instructions.

13. Bring up the HP BASIC/WS client on the SRM connection by following the instructions on booting the first client in chapter 5 of this manual.
14. Bring up the first Pascal Workstation System client on the SRM connection by following the instructions on booting the first client in chapter 5 of this manual.
15. Bring up the first HP BASIC/WS client on the LAN connection by following the instructions on booting the first client in chapter 5 of this manual.
16. Bring up the first Pascal Workstation System client on the LAN connection by following the instructions on booting the first client in chapter 5 of this manual.
17. Bring up all of the remaining clients on both the SRM and LAN connection by following the instructions in the "Adding Clients to the Booting Environment" section of chapter 5 of this manual.

You are now done bringing up the Series 300 server's subsystem. You verified the client booting environment in the steps above. Next, move around in various volumes of the SRM/UX file system from all of the clients to verify correct operation. And, finally, try spooling some files from each client to test the SRM/UX spooling environment.

Note Pascal users must modify their TABLE programs in order to access more than one server, or more than one volume per server. See "Giving Pascal Clients Access to Multiple SRM/UX Volumes" in Chapter 7 for details.

Configure the Series 800 Server

1. Set up the system hardware by following the instructions in chapter 3 of this manual, and the instructions in each of the component's manuals.

Note The Series 800 server can not have SRM clients.

2. Photocopy the "SRM/UX Client Descriptions" worksheet and the "SRM/UX LAN Worksheet" in chapter 2.

3. Fill out the `SRM/UX Client Descriptions` worksheet as follows:

Table 9-10.
Example 3—SRM/UX Client Descriptions (Series 800 server)

SERVER NAME: SERV800			
Client Name	**Comments**	**Connection Type**	**Volume List**
kristin	BASIC/WS Kristin W. s370 ext.191	LAN	SRMUXROOT, KRISTIN
jim	PAWS Jim H's s340 ext.212	LAN	SRMUXROOT, JIM
hpwgs	BASIC/WS Fred G. Bldg.2	LAN	SRMUXROOT, HPUXROOT, HPWGS

4. Fill out the "SRM/UX LAN Worksheet" for the Series 800 server (at `Server Node Address` 2) as follows:

Table 9-11.
Example 2—SRM/UX LAN Worksheet (Series 800 server)

SERVER NAME: SERV800							
Client Name	Client Link Level Address	Client Internet Address	Client Node Address[1]	Server Node Address	Uid	Gid	Umask
kristin	0x08000901047A	15.99.99.33	14	2	111	9	022
jim	0x080009003824	15.99.99.35	15	2	113	9	022
hpwgs	0x0800090B3165	15.99.99.37	16	2	115	9	022

[1] The Client Node Address and Server Node Address are emulated on the LAN connection

5. Ensure that the HP-UX System Administrator has HP-UX installed, configured, and running on the Series 800 server computer.

6. Install the SRM/UX server software on the Series 800 server by following the instructions in chapter 4 (and chapter 5 for installing the ability for clients to boot from the server) of this manual.

7. Configure the Series 800 server's HP-UX environment for SRM/UX by following the instructions in chapter 4 of this manual. This involves reconfiguring the HP-UX kernel.

8. Edit your Series 800 server's /etc/srmdconf file to appear as follows:

```
# Long lines in this example will wrap in this printed output.
VOLUME-TABLE

# Volume Descriptions
#
# Volume         |Volume|     |     |Temp     |Root
# Name           |Number| Uid | Gid |Directory|Directory
#----------------+------+-----+-----+---------+---------------------------------
SRMUXROOT        :   8  :     :     :  tmp    : /srmuxroot
HPUXROOT         :   9  :     :     :  tmp    : /
KRISTIN          :  10  :     :     :  tmp    : /users/kristin
JIM              :  10  :     :     :  tmp    : /users/jim
HPWGS            :  10  :     :     :  tmp    : /users/hpwgs

LAN-CLIENTS

# LAN Client Descriptions
#
# Client       |Client          |Client|Server|          |     |     |     |
# Link Level   |Internet        |Node  |Node  |Client    |     |     |     |Volume
# Address      |Address         |Addr  |Addr  |Name      | Uid | Gid |Umask|List
#--------------+---------------+------+------+----------+-----+-----+-----+---------
0x08000901047A :  15.99.99.33  :  14  :   2  :  kristin : 111 :   9 : 022 :SRMUXROOT,
KRISTIN  # Kristin W's hp9000s370
0x080009003824 :  15.99.99.35  :  15  :   2  :  jim     : 113 :   9 : 022 :SRMUXROOT,
JIM      # Jim's  hp9000s340
0x0800090B3165 :  15.99.99.37  :  16  :   2  :  hpwgs   : 115 :   9 : 022 :SRMUXROOT,
HPUXROOT, HPWGS   # Fred's hp9000s370

SRM-CLIENTS

# SRM Client Descriptions
#
#                |Client|          |     |     |     |
# Server's SRM   |Node  |Client    |     |     |     |Volume
# Device File    |Addr  |Name      | Uid | Gid |Umask|List
#----------------+------+----------+-----+-----+-----+------------------------------
```

```
SPOOL-ENVIRONMENT

# The following environment variables are pre-initialized for spooling
# processes.
#
#       SHELL                   - defaults to /bin/sh, unless overridden
#       TZ                      - inherited from the OS, if it is set there,
#                                 otherwise defaults to MST7MDT
#       SRMSPOOLFILENAME        - defaults to the full pathname to the file
#                                 being spooled (e.g. /srmuxroot/PRINTER/xyzzy)
#       PATH                    - defaults to /bin:/usr/bin. Recommend that
#                                 it be set to include paths needed for all
#                                 commands used in Spool Commands.
#-------------------------------------------------------------------------------
PATH=/bin:/usr/bin:             # srmdlpfilter is in /usr/bin

SPOOL-TABLE

# Spooling Commands
#
# Spool               |        |Spool
# Directory           |Priority|Command
#---------------------+--------+-----------------------------------------------
/srmuxroot/PRINTER    :   2    : srmdlpfilter | lp $PRIORITY
```

9. Run *srmdconfcheck*(1M) to validate the above srmdconf file.

10. Create the Volume 10 directories that you have entered into the VOLUME-TABLE above, i.e., those for KRISTIN, JIM, and HPWGS.

Note We recommend that you create a `tmp` directory in the `Root Directory` of each Volume that appears in the VOLUME-TABLE of the /etc/srmdconf file below (e.g., for Volume KATHY, there would be a /users/kathy/tmp directory).

You may instead choose to put all `Temp Directories` in a common location, such as /tmp, as long as each `Temp Directory` is under the same HP-UX mount point as the `Root Directory` of its Volume.

11. Bring up the server daemon, *srmd*(1M), by executing:

    `/etc/srmd`

    See the "Starting the srmd Process on the Server" section in chapter 4 for detailed instructions.

12. Bring up the first HP BASIC/WS client on the LAN connection by following the instructions on booting the first client in chapter 5 of this manual.
13. Bring up the Pascal Workstation System client on the LAN connection by following the instructions on booting the first client in chapter 5 of this manual.
14. Bring up all of the remaining clients on the LAN connection by following the instructions in the "Adding Clients to the Booting Environment" section of chapter 5 of this manual.

You are now done bringing up the Series 800 server's subsystem. You verified the client booting environment in the steps above. Next, move around in various volumes of the SRM/UX file system from all of the clients to verify correct operation. And, finally, try spooling some files from each client to test the SRM/UX spooling environment.

Note Pascal users must modify their TABLE programs in order to access more than one server, or more than one volume per server. See "Giving Pascal Clients Access to Multiple SRM/UX Volumes" in Chapter 7 for details.

The Full SRM/UX System

If you have followed the procedures in each of the server sections above, then you should now have a completely configured SRM/UX system. Verify the operation of all the clients in terms of client booting, moving around the SRM/UX file system, and using the SRM/UX spooling environment.

If you have problems, ensure that you have followed the above instructions carefully. If you have, then see Appendix B for help in troubleshooting your system.

Recommendations For Managing Complex Systems

Clients may access multiple servers on SRM and/or LAN connections. Following the rules for setting unique node addresses on all SRM cards will ensure correct operation of SRM/UX on the SRM connection. The following rules and recommendations will ensure correct operation of SRM/UX on the LAN connection:

- Each client, as defined by its `Client Link Level Address`, MUST have the SAME `Client Internet Address` in /etc/srmdconf on ALL of its servers.
- Each client, as defined by its `Client Link Level Address`, MUST be given a UNIQUE `Server Node Address` in /etc/srmdconf on EACH of its servers. This will be easier to manage if each server uses the same `Server Node Address` for all of its clients.
- We recommend that each client, as defined by its `Client Link Level Address`, be given the SAME `Client Node Address` in /etc/srmdconf on ALL of its servers.

Client computers may use multiple SRM or LAN interfaces. The following rules and recommendations will ensure correct operation of SRM/UX:

- If a client has multiple LAN interfaces, then each interface MUST be assigned a UNIQUE `Client Internet Address` in /etc/srmdconf, even if they are on the same subnet.
- A client may use multiple interfaces to access the same server, to provide hardware redundancy. However, we recommend that a client NOT use multiple interfaces to access the same server SIMULTANEOUSLY, because the server will not be able to tell that it is serving a single client instead of multiple clients. Operations such as file locking may cause deadlocks in this situation.

Protecting Data

As the SRM/UX System Administrator, you may wish to prevent other HP-UX programs from modifying files maintained by SRM/UX. SRM/UX file locking (for example, the HP BASIC/WS "LOCK" command) does not provide this protection. It only protects files from other SRM/UX clients.

You could use HP-UX security mechanisms to protect a tree of the file system, and at the same time allow clients to share data freely or protect themselves from each other. For example, you could ensure that only SRM/UX clients can modify files kept under /srmuxroot, by making sure all clients share the same Group ID (Gid). Specify a Gid of 9 for this purpose. Set the Gid field of volume 8 in /etc/srmdconf to 9, and start up *srmd(1M)*, or execute *srmdreconfig(1M)*, if *srmd(1M)* is already running. Then, all clients will belong to group 9 when accessing volume 8. If you choose a gid other than 9 for this scheme, you will need to change the group ID of all shared directories and files to that gid to allow client access to the shared directories and files.

When you installed SRM/UX on the server, all files and directories defaulted to `Uid`=1 (bin) and `Gid`=9 (ws).

Make sure that /srmuxroot, and all of the directories and files that you wish clients to be able to modify, belong to group 9.

On HP BASIC/WS clients, you can use the CAT command to display the group of files, and the CHGRP command to change the group if the client owns the file. From HP-UX on the server, the *ll*(1) and *chgrp*(1) commands can accomplish the same tasks.

Set the permissions of all directories and files to 775. This allows clients to access them through group permissions, while preventing users other than root from modifying them.

Client users may set the permissions on specific files to prevent group sharing if they wish.

A

Porting BASIC and Pascal Application Programs from SRM to SRM/UX

Porting BASIC Programs

Differences Between SRM and SRM/UX

SRM/UX differs from the SRM system in the following ways:

- SRM/UX supports HFS-style permissions (using CHGRP, CHOWN, and PERMIT) to control access to files and directories.
- SRM/UX does not support use of the PROTECT statement to control access to files and directories.
- SRM/UX ignores SRM passwords.
- CAT listings of SRM/UX directories differ from SRM listings in that they reflect the change to HFS-style permissions.
- SRM/UX can be used with local area networks (LAN connections), in addition to SRM connections.
- SRM/UX allows the use of named pipes.

The sections that follow describe the above differences as they apply to porting applications that run on SRM so they will run on SRM/UX.

Use of the MSVS

Users should address the SRM/UX server's file system in the same way as they address the SRM. They should provide the Select Code, Server Node Address, and Volume Name in the form, :REMOTE *sc*,*sna*;LABEL *volname*, for example:

```
:REMOTE 21,0;LABEL DISC1
```

which is the same as a standard SRM msvs. The Server Node Address and Volume Name are determined by the entries in the LAN-CLIENTS table in the /etc/srmdconf file made by the SRM/UX System Administrator for LAN clients. For SRM clients, the Server Node Address is determined by the switches on the server's SRM interface card. The Volume Name is determined by the entries in the SRM-CLIENTS table in the server's /etc/srmdconf file.

Because the msvs does not distinguish between an actual SRM server file system and the SRM/UX server file system, the user can execute the following program segment to find out which type of file system he or she is accessing:

```
10  DIM A$(0:2)[80]
20  MSI ":REMOTE 21,0; LABEL DISC1"   ! or whatever the msvs
30  CAT TO A$(*)
40  PRINT A$(2)
```

The result: `FORMAT: SDF` will indicate a true SRM file system, while: `FORMAT: SRM-UX` will indicate an SRM/UX file system. (This applies for both LAN and SRM cards.)

Passwords and File Access Control

The main difference between SRM and SRM/UX from the HP BASIC/WS clients' perspectives is how the server controls file access. That is:

- SRM uses a password-per-file scheme, with each password controlling up to 6 different properties.
- SRM supports multiple passwords per file or directory, each with different property control sets. The SRM file access control properties are unrelated to the HP-UX read/write/execute file protection properties; that is, SRM passwords and the HP-UX file protection scheme are not compatible. (See the *HP-UX System Administration Concepts* manual in the "File system" chapter for a complete discussion of HP-UX file protection.)
- The HP-UX file protection properties are the ones that must be observed by SRM/UX to protect the non-SRM/UX HP-UX files and processes running on the server platform. The SRM/UX server ignores SRM passwords specified in path names. Therefore, the client will not have to modify any applications

if they presently use passwords in path names, with the exception of HP BASIC/WS application programs that use the PROTECT statement.

Different CAT Statement Format

The output of the CAT statement from SRM/UX is different from SRM and reflects the HFS-like permissions instead of the SRM passwords. It is a combination of the SRM and HFS formats (see the examples below). The SRM/UX server hides files with names longer than 16 characters from the HP BASIC clients.

The output of CAT TO A$(*) has the standard SRM format, just as LIF and HFS do in the current release of HP BASIC. (The PUB ACC field of SRM output will show an 'M' if the user owns the file. It will show R and/or W depending on whether or not the public read and write permission bits are set. These rules are the same as the current local HFS rules for CAT TO A$(*).) If the user specifies the EXTEND keyword on CAT TO A$(*), then the output will be formatted with SRM/UX formatting rules instead. (This is analogous to the way HFS and LIF CAT output is done on CAT TO A$(*).) The string array must be declared with at least 80 characters per element; this is the same size requirement as the SRM CAT TO A$(*).

Example SRM/UX CAT Output—80 Column Display

```
         1         2         3         4         5         6         7         8
12345678901234567890123456789012345678901234567890123456789012345678901234567890

:REMOTE 21,0
LABEL:  BOOT
FORMAT: SRM-UX
AVAILABLE SPACE: 123456789
                  FILE   NUMBER  REC     MODIFIED                           OPEN
FILE NAME        TYPE   RECORDS  LEN  DATE       TIME   PERMS   OWNER GROUP STAT
================ ===== ========  ==== =============== ========= ===== ===== ====
SYSTEMS          DIR       1024     1  1-Mar-90 16:56 RWXRWXRWX     2     9
WORKSTATIONS     DIR       1024     1  7-Feb-90 12:30 RWXRWXRWX     2     9
EDITTEST.TEXT    TEXT         8   256 12-Dec-89 15:20 RW-R--R--   175    54
AUTOST           PROG         2   256  5-Jan-90 15:07 RW-RW-RW-   175    54
PRINTER          DIR       1024     1  7-Feb-90 12:30 RWXRWXRWX     2     9
PTEST            ASCII        1   256  2-Jan-90 10:51 RW-RW-RW-    17     9 LOCK
PTESTCAT         HP-UX      984     1  2-Mar-90 15:12 RW-RW-R--   175    54 OPEN
```

Example SRM/UX CAT Output—50 Column Display

```
         1         2         3         4         5
12345678901234567890123456789012345678901234567890

:REMOTE 21,0
LABEL:  BOOT
FORMAT: SRM-UX
AVAILABLE SPACE: 123456789
                  FILE   NUMBER  REC            OP
FILE NAME         TYPE  RECORDS  LEN    PERMS   ST
================ ===== ======== ===== ========= ==
SYSTEMS          DIR       1024     1 RWXRWXRWX
WORKSTATIONS     DIR       1024     1 RWXRWXRWX
EDITTEST.TEXT    TEXT         8   256 RW-R--R--
AUTOST           PROG         2   256 RW-RW-RW-
PRINTER          DIR       1024     1 RWXRWXRWX
PTEST            ASCII        1   256 RW-RW-RW- LO
PTESTCAT         HP-UX      984     1 RW-RW-R-- OP
```

LAN Interface Status/Control Registers

SRM/UX status registers have been implemented to correspond as closely as possible with the SRM interface card status registers, except that there are additional registers for which there is no SRM analog. In addition, new LAN control registers have been implemented.

```
Status
Register          Meaning
    0       Card Identification
               21 if the Remote Control switch (R) is set to 0
              149 if the switch is set to 1

    1       Interface Interrupts
                1 = interrupts enabled
                0 = interrupts disabled

    2       Not implemented  (interface busy on SRM)

    3       Not implemented  (interface Firmware ID on SRM)

    5       Data Availability
                0 = receiver buffer empty
```

This will always be the result, because the performance of the LAN card is so high. Hence, the BASIC user will never perceive any other situation.

6 Node Address

When the SRM binary is loaded, a "pseudo" node number is provided by the SRM/UX server (from /etc/srmdconf). This 2-digit number is used for a node-specific autostart file (e.g., /SYSTEMS/AUTOSTxx). If the SRM binary is not loaded, STATUS of this register will return 0.

7 CRC Errors (and Frame Errors)

Cyclic redundancy check errors plus frame errors detected since powerup or reset

8 Number of Buffer Overflows

in receive buffers since powerup or reset

9 Card State

0 = card not powered up
1 = card buffers incorrectly defined
2 = data/address ports not responding
3 = card stopped
4 = normal operation
5 = driver detected a non-recoverable hardware error; card is stopped

11 Amount of available Space in Transmit Buffer

12 Number of Transmission Retries

since powerup or reset

13 Current scale factor for connection establishment timeouts (default value is 1; default reset at powerup and SCRATCH A);

	legal range of values is 1 to 32 inclusive
14	Current scale factor for normal operations timeouts (default value is 1; default reset at powerup and SCRATCH A); legal range of values is 1 to 32, inclusive
21-26	Link Level Address 6 Bytes Status registers 21 through 26 each contain one byte of the Link Level Address. The first byte is in register 21, the second byte is in register 22, etc.

Control Register	Meaning
18	Initialize driver statistics to 0 Ordinarily done at power up
35	Set Default Configuration Causes driver to set all of its pseudo registers to power up defaults; hardware is not reconfigured Link address is set from interface card RAM MODE is set to 0 Multicast mask is set to all 0's Receive buffer size is set to 32 Number of receive buffers is set to 320 Number of transmit buffers is set to 4 LAN stats are set to zero

Porting Pascal Programs

The Filer with SRM/UX

The Filer commands available to the Pascal Workstation System user are a mixture of SRM and HFS commands. For SRM/UX volumes, the Filer displays both file locking and HFS permissions information when the extended listing is requested.

For a normal listing, the only difference visible to the user is where the Filer displays: `Directory type =`. For SRM volumes, as an example, `Directory type = SRM 21,0,8` is displayed. This shows that the select code is 21, the server node is 0, and the disc volume is 8. For HFS volumes, `Directory type = HFS 777 17 9` will be displayed, indicating file permissions 777 and `uid = 17`, `gid = 9`. For SRM/UX volumes when booted over LAN, `Directory type = SRM/UX 21,127,8` will be displayed, i.e. the emulated SRM information is displayed.

For an extended listing with an SRM/UX volume, the field `..directory info ...` has been enhanced to show both the SRM- like file locking status for the file, and also the HFS permissions associated with the file. For example, SRM volumes may display:

```
..directory info...
MRWSPC CLOSED
```

Also, the words SHARED, EXCLUSIVE and CORRUPT can be used to replace CLOSED in the directory info description. Of course, CLOSED, SHARED and EXCLUSIVE represent the file locking attributes held by the file.

MWRSPC describes access rights to the file.

HFS volumes display:

```
...directory info...
d777m   17u    9g
```

If the file is a regular file, d will be blank. The 777 entry describes the file permissions, 17 and 9 the user and group id's associated with the file. See the Pascal Workstation System manuals for more details on HFS.

For SRM/UX volumes, the directory information field will contain a mixture of these two formats. The HFS information will remain in place, and also an abbreviation will indicate the current file locking status with 'CL' for closed, etc. :

```
...directory info.....
d777m   17u    9g CL
```

Here 'CL' could also be 'EX', 'SH' or 'CO'.

Passwords and File Access Control

The main difference between SRM and SRM/UX from the Pascal Workstation System client's perspective is how the server controls file access. That is:

- SRM uses a password-per-file scheme, with each password controlling up to 6 different properties.
- SRM supports multiple passwords per file or directory, each with different property control sets. The SRM file access control properties are unrelated to the HP-UX read/write/execute file protection properties; that is, SRM passwords and the HP-UX file protection scheme are not compatible. (See the *HP-UX System Administration Concepts* manual in the "File system" chapter for a complete discussion of HP-UX file protection.)
- The HP-UX file protection properties are the ones that must be observed by SRM/UX to protect the non-SRM/UX HP-UX files and processes running on the server platform. The SRM/UX server ignores SRM passwords specified in path names.

New Commands

The Filer's HFS command has been enhanced to work with SRM/UX units as well as HFS units (note that for the workstation, HFS means a local hard disk and not the SRM/UX file system which is an HFS disk from the server's perspective). With the HFS command, for files to which the workstation has access, the FILER may be used to modify the `Uid` and `Gid` fields, as well as the file mode which describes access rights for owner, group, and other.

In addition, the TABLE program has been modified to allow the LAN interface to replace the SRM interface card when making a `tea_srm` call. This means that users may modify the TABLE program for SRM/UX units in the same way as they always have for SRM units, whether they are running SRM/UX over the SRM connection or the LAN connection.

B

Troubleshooting Your SRM/UX System

You will need an HP-UX account on the host computer that is acting as the SRM/UX server to use most of the techniques and procedures in this chapter. If the HP-UX System Administrator gives a client user an HP-UX account on the SRM/UX server, then he or she can use the troubleshooting techniques in this chapter. Otherwise, the SRM/UX System Administrator will have to do the troubleshooting. In most cases, you do not need to have super-user privileges to perform these procedures.

The problems below are presented from the client user's perspective; that is, they are presented as you would see them from a client workstation as you were trying to use SRM/UX.

General Information

SRM/UX uses several log files to provide helpful information for tracking down problems.

- The server process, *srmd*(1M), logs messages to /usr/adm/srmd.log
- Spooling processes started by *srmd*(1M) log messages to /usr/lib/srmux/srmdspool.log
- The HP-UX diskless-boot-daemon, *rbootd*(1M) (used for booting clients over the LAN connection), logs messages to /usr/adm/rbootd.log, (unless the HP-UX System Administrator has specified a different log file with the -L option on the invocation of rbootd in /etc/rc).

Note The SRM Access Utilities (e.g. *srmcp*(1)) will not correctly move files between SRM and SRM/UX, unless the files are of type UX. The reason is that other file types (such as PROG, BDAT, or TEXT types) need to have a 512 byte header block, called a "WSHeader", stored with them on SRM/UX. The "WSHeader" contains information that is included in directories on SRM, but cannot be included in HP-UX directories on the SRM/UX server. The SRM Access Utilities cannot create or understand "WSHeaders".

Client Booting Problems

SRM Connections

The Symptom: The Boot ROM menu doesn't display any files from the SRM/UX server.

The possible problems:

1. Is *srmd*(1M) running on the server?
2. Is this client configured correctly on server?
3. Are there any SYSTEM files in /SYSTEMS on Volume 8 (usually /srmuxroot/SYSTEMS)?
4. Are there any hardware problems ?

The solutions:

1 and 2. To check the first two items, i.e., whether *srmd*(1M) is running on the server and whether this client is correctly configured on the server, proceed as follows:

- Use *srmdstat*(1) on the server. For example, if the client SRM card is at node address 20, and the device file for the SRM card on the server is /dev/srm, you would execute (from the server):

  ```
  /etc/srmdstat -s /dev/srm 20
  ```

- Or, if you know that your client workstation is supposed to have a client name of ABCD you could execute:

```
/etc/srmdstat -n ABCD
```

After you execute *srmdstat*(1):

- If *srmd*(1M) is not running, you will see a message on the server's screen similar to this:

  ```
  srmdstat: SRM/UX server not executing
  ```

  See your SRM/UX System Administrator to get *srmd*(1M) started. (See the "Starting the srmd Process on the Server" section of Chapter 4.)

- If *srmd*(1M) is running, but your workstation is not configured as a client, you will see an *srmdstat*(1) message similar to this:

  ```
  SRM client not found
  ```

  OR this:

  ```
  client name not found
  ```

  Your SRM/UX System Administrator will need to add a client definition for your workstation to /etc/srmdconf, or possibly fix a definition that is already there. He or she can use *srmdconfcheck*(1M) and *srmdreconfig*(1M) to do this without having to shut down the entire server (which would impact other SRM/UX clients).

- If your workstation is configured correctly, you will see an *srmdstat*(1) message similar to this:

  ```
  ABCD on /dev/srm at node 20 not active
  ```

  OR this:

  ```
  ABCD on /dev/srm at node 20 has 0 open files
  ```

3. To check if there are any SYSTEM files in /srmuxroot/SYSTEMS:

- From the server execute:

  ```
  ll /srmuxroot/SYSTEMS
  ```

  to see if the correct system boot files are present in the directory.

- From an HP BASIC/WS client:

If another SRM/UX client is able to boot successfully from your server, then execute `CAT "/SYSTEMS"` from the other client and you should see the expected files with "type" SYSTM in the CAT output.

- From a Pascal Workstation System client:

  If another SRM/UX client is able to boot successfully from your server, then do an Extended list in the Filer of the /SYSTEMS directory from the other client and you should see the expected files with "type" SYSTM in the Filer output.

4. If there are any hardware problems:

- Hardware problems are mostly beyond the scope of this manual. However, if your client workstation has an HP 50961 Resource Management Coax Adaptor, make sure that the switches on the SRM card itself, and on the Coax Adaptor are set to the same Node Address. (It is easy to mistakenly set one of the switches to the 1's complement of the other.) See chapter 3 for instructions on setting up the hardware for an SRM/UX system.

LAN Connections

Symptom 1: The Boot ROM menu doesn't display any files from the SRM/UX server.

The possible problems:

1. Does the client have the correct Boot ROM for booting from LAN?
2. Is *rbootd*(1M) running on the server?
3. Is this client configured correctly on the server?
4. Is /etc/boottab configured correctly?
5. Are there any SYSTEM files in /SYSTEMS on Volume 8 (usually /srmuxroot/SYSTEMS)?

The solutions:

1. Check the version of your client's Boot ROM:

- The Boot ROM menu will show what version of the Boot ROM is installed in your client. You must have Revision B or later to boot from LAN. All Series 300 client Boot ROMs can be upgraded to Revision B, if necessary, in order to boot from LAN. However, Series 200 computers cannot support booting from LAN.

2. Check to see if *rbootd*(1M) is running on the server:

- Execute the following HP-UX command from the server:

  ```
  ps -ef | grep rbootd | grep -v grep
  ```

  If the command doesn't produce any output, then *rbootd*(1M) is not executing. See your HP-UX System Administrator to get *rbootd*(1M) running.

3. Check that this client is configured correctly on the server:

- Check that the Link Level Address of your workstation's LAN Interface (displayed by most Boot ROMs at powerup) matches the `Client Link Level Address` declared for this client in the LAN-CLIENTS table in the server's /etc/srmdconf file.

- Check that your workstation's `Volume List` in the LAN-CLIENTS table in /etc/srmdconf includes the `Volume Name` for Volume 8 as it is defined in the VOLUME-TABLE in /etc/srmdconf. (If there is a mismatch in the labels, then *srmdconfcheck*(1M) will report an error, or an error message will be in /usr/adm/srmd.log from when *srmd*(1M) was last started up or reconfigured.)

4. Check to see that /etc/boottab is configured correctly:

- Check that correct file names for your system boot files appear in the paws-srm (or basic-srm) line(s) in /etc/boottab. (See chapter 5 for details on configuring /etc/boottab.)

5. To check if there are any SYSTEM files in /srmuxroot/SYSTEMS:

- From the server execute:

  ```
  ll /srmuxroot/SYSTEMS
  ```

  to see if the correct system boot files are present in the directory.

- From an HP BASIC/WS client:

  If another SRM/UX client is able to boot successfully from your server, then execute `CAT "/SYSTEMS"` from the other client and you should see the expected files with "type" SYSTM in the CAT output.

- From a Pascal Workstation System client:

If another SRM/UX client is able to boot successfully from your server, then do an Extended list in the Filer of the /SYSTEMS directory from the other client and you should see the expected files with "type" SYSTM in the Filer output.

Symptom 2: The Boot ROM boots a system, but the client workstation does not complete its initialization correctly.

The possible problems:

1. Is *srmd*(1M) running on server?
2. Is this client configured correctly on server?
3. Is some other computer using your workstation's Internet Address?
4. If booting a Pascal client, are the boot extension files present in the correct directory?
5. If booting a Pascal client, is the last thing displayed on the CRT the copyright message?
6. If booting a Pascal client, are Volumes #5 and #45 undefined?

The solutions:

1 and 2. To check the first two items, i.e., if *srmd*(1M) is running on the server and if this client is correctly configured on the server, proceed as follows:

- Use *srmdstat*(1) on the server. For example, if the `Client's Internet Address` is 15.99.99.10, you would execute:

  ```
  /etc/srmdstat -i 15.99.99.10
  ```

- Or, if you know that your client workstation is supposed to have a `Client Name` of XYZ you could execute:

  ```
  /etc/srmdstat -n XYZ
  ```

After you execute *srmdstat*(1M):

- If *srmd*(1M) is not running, you will see a message on the server's screen similar to this:

  ```
  srmdstat: SRM/UX server not executing
  ```

  See your SRM/UX System Administrator to get *srmd*(1M) started. (See the "Starting the srmd Process on the Server" section of Chapter 4.)

- If *srmd*(1M) is running, but your workstation is not configured as a client, you will see an *srmdstat*(1) message similar to this:

  ```
  LAN client not found
  ```

  OR this:

  ```
  client name not found
  ```

  Your SRM/UX System Administrator will need to add a client definition for your workstation to /etc/srmdconf, or possibly fix a definition that is already there. He or she can use *srmdconfcheck*(1M) and *srmdreconfig*(1M) to do this without having to shut down the entire server (which would impact other SRM/UX clients).

- If your workstation is configured correctly, you will see an *srmdstat*(1) message similar to this:

  ```
  XYZ at 15.99.99.10 not active
  ```

  OR this:

  ```
  XYZ at 15.99.99.10 has 0 open files
  ```

3. Check to see if some other computer is using your workstation's Internet (IP) Address:

- Check /etc/hosts on your server to see if some other computer is also trying to use your workstation's IP address. It is recommended that you add SRM/UX clients to the server's /etc/hosts file to help avoid this problem. (See chapter 4 for details.)

- If you have multiple SRM/UX servers on your LAN connection (subnet), check the /etc/srmdconf files on all servers to see if a different SRM/UX client on one of the other servers is using your workstation's IP address.

- If another HP-UX computer is using your workstation's IP address, you can usually detect it with HP-UX by executing:

  ```
  telnet 15.99.99.10
  ```

  where `15.99.99.10` is an example IP address; you should use *your* client's IP address in the *telnet*(1) statement.

  If you get a `login` prompt from some HP-UX computer, then you have found the computer which is using your workstation's IP address.

If you find that your `Client Internet Address` is a duplicate of another SRM/UX client, then have your SRM/UX System Administrator(s) correct the situation. If your client's IP address is a duplicate of a computer that is not a member of the SRM/UX system, then you will have to get your site's HP-UX System Administrator(s) together with your SRM/UX System Administrator(s) to resolve the problem.

4. The INITLIB, STARTUP, and TABLE files must be present in either /WORKSTATIONS/SYSTEMnnnnnn or /WORKSTATIONS/SYSTEM. Have your SRM/UX System Administrator verify that these extension boot files are in one of these correct locations.

5. If the copyright message is the last thing displayed on your CRT, then use the techniques in the "Booting the First Pascal Workstation System Client Over LAN" section of chapter 5 to establish a connection to your SRM/UX server. Then:

- If you are using a Series 200 computer as the client workstation, make sure that your INITLIB file came from the BOOT: flexible disc (the INIT_200 version of INITLIB as discussed in chapter 7). Or if you are using a Series 300 computer as the client workstation, make sure that your INITLIB file came from the BOOT2: flexible disc (the INIT_300 version of INITLIB as discussed in chapter 7).

6. If Volumes #5 and #45 are undefined, then use the techniques in the "Booting the First Pascal Workstation System Client Over LAN" section of chapter 5 to establish a connection to your SRM/UX server. Then:

- Make sure modifications to TABLE do not disable SRM/UX access. Use the EDITOR to examine the source CTABLE.TEXT (or whatever you may have named it), and ensure that no changes you have made are affecting units #5 or #45. See the "Modifying the TABLE Program" section of the *Special Configurations* chapter of *Pascal Workstation System, Volume 2* for details on safely modifying CTABLE.TEXT. Follow those instructions for reinstalling TABLE.
- Make sure that the necessary drivers for SRM/UX on the LAN connection are in INITLIB. Use the Librarian to verify that the IOMPX, LANDVR, and SRM modules are in INITLIB.

File Serving Problems

Symptom 1: Accesses to an open file appear to hang.

The possible problem:

1. Does another SRM/UX client have the file locked?

The solution:

1. Check to see if another SRM/UX client has the file locked:

- Use *srmdstat*(1) to see which SRM/UX clients have the file open, and to check to see if any have the file locked. (See chapter 4 and/or the man page for a discussion of *srmdstat*(1))

  Execute (for example):

  ```
  srmdstat -f /srmuxroot/SHARED_DATA
  ```

  which would produce output that looks like this:

  ```
  /srmuxroot/SHARED_DATA open by:
             XYZ at 15.99.99.10 owns LOCK
             RST at 15.99.99.11
  ```

  In this example, client RST's attempts to read or write this file would hang until client XYZ released its LOCK.

  The SRM/UX System Administrator could force client XYZ to release the lock by using *srmdreset*(1M). (See the "Resetting SRM/UX Clients" section of chapter 4 and/or the *srmdreset*(1M) man page.)

Symptom 2: HP BASIC/WS clients get `ERROR 450 - (Volume not found)`

The possible problem:

1. Does the volume LABEL in the HP BASIC/WS statement match one of the `Volume Names` in the client's `Volume List` in the LAN-CLIENTS or SRM-CLIENTS tables in /etc/srmdconf? (It must.)

   The solutions for 1:
   If the BASIC/WS user misspelled the volume LABEL, reexecute the BASIC command with a correct volume LABEL.

Use *srmdconfcheck*(1M), or look at /usr/adm/srmd.log, to make sure that the volume was configured correctly in /etc/srmdconf. (See chapter 4 and/or the *srmdconfcheck*(1M) man page.)
See the SRM/UX System Administrator if you need to add an additional volume to your client's `Volume List`. The SRM/UX Administrator can edit /etc/srmdconf, and use *srmdconfcheck*(1M) and *srmdreconfig*(1M) to add a new volume without having to shut down and restart the SRM/UX server. (The details on how to add a volume to a client's `Volume List` appear in chapter 4.)

Symptom 3: HP BASIC/WS clients get `ERROR 51 - (File not currently assigned)` or Pascal clients get `error -10: unrecognized SRM error`.

The possible problems:

1. The SRM/UX Administrator stopped the server with *srmdshut*(1M)
2. The SRM/UX Administrator closed all of the client's files with an *srmdreset*(1M)
3. The client was rebooted or reinitialized.

Examine /usr/adm/srmd.log for more details. (See chapter 4 for details about *srmdshut*(1M) and *srmdreset*(1M), and chapter 5 for details about rebooting and reinitialization.)

The solutions:

1, 2, and 3. HP BASIC/WS clients are fairly tolerant of server restarts. You can often continue an interactive session by executing:

```
MSI SYSTEM$("MSI")
```

from the HP BASIC/WS client, although you will probably have to rerun programs from the beginning.

1, 2, and 3. For Pascal Workstation System clients to recover from server restarts, you should put the line `P *TABLE` in your AUTOSTART file. Then after *srmd*(1M) is restarted, the Pascal user can execute TABLE from the Command Interpreter. You will have to rerun programs from the beginning.

Symptom 4: HP BASIC/WS clients on the LAN connection sometimes get `ERROR 319 (Connection not established)` or `ERROR 353 (Data link failure)`

The possible problems:

1. Is the server heavily loaded? (Large numbers of clients and/or clients doing lots of file I/O to SRM/UX)
2. Is the SRM/UX server accessing NFS-mounted file systems for clients?

The solutions:

1. Check to see if the server is heavily loaded. (i.e., are there large numbers of clients and/or clients doing lots of file I/O to SRM/UX?)

- You can increase the timeout values for HP BASIC/WS by using CONTROL registers 13 and 14. The values of these registers, which can be up to 32, are used to increase the timeouts of the HP BASIC/WS client. The value is multiplied by the default timeout to generate the actual timeout used; so a value of 4 will increase the timeout by a factor of 4. (See the LAN Interface STATUS and CONTROL registers section of Appendix A of the *HP BASIC/WS Language Reference* manual for details (Version 6.1 or later).)

2. Check to see if the SRM/UX server is accessing NFS-mounted file systems for clients:

- For NFS-mounts to systems on the same LAN connection (subnet) as the SRM/UX server, no problems should occur. However, if the NFS-mount is accessed via a lower bandwidth connection, then the response time of the NFS-mount may become long enough to cause SRM/UX client-requests to time out.

  NFS-mounts which are not on the same LAN connection (subnet) should always use the `-o rsize=1024,wsize=1024` option on the HP-UX */etc/mount* command, which does the NFS-mount. (You can specify this in the /etc/checklist file or as an option to */etc/mount*; see *mount*(1M).)

  Either:

  □ Use CONTROL registers 13 and 14 as described above

  Or:

  □ Restructure the application to use an HP-UX process to transfer data to and from the SRM/UX server, so that SRM/UX clients can always access files which are local to the SRM/UX server.

Pascal-Specific File Serving Problems

Linking Temporary Files to Their Final Destinations

Many of the Pascal Workstation System programs (such as the Compiler and Librarian) will create temporary files, write to them, and then attempt to link them to their final name. This scenario is especially common when the final target file already exists, and will be replaced only if the operation, such as a compile, succeeds. If the final target file and the `Temp Directory` are on the same physical disk, then everything works correctly. Otherwise the link will fail because hard links are not allowed across physical disks. This can occur if the SRM/UX server has local mounted disks, or if it has NFS mounts. (The SRM will have analogous problems if the SRM's MOUNT command is used.)

The best solution to this problem is usually to configure the VOLUME-TABLE in /etc/srmdconf so that each physical disk, or NFS mount point, which is to be accessed by SRM/UX clients, has its own volume.

If the SRM/UX server has only 1 local disk, and no other local mounted disks or NFS mounts, then this problem will not occur.

Spooling Problems

General Information

- Spooling processes started by *srmd*(1M) will log messages to `/usr/lib/srmux/srmdspool.log`
- HP-UX spooling for *lp*(1) will log messages to /usr/spool/lp/log
- If you have spooling problems, you can check the output of *srmdconfcheck*(1M) to see if the SPOOL-TABLE section of /etc/srmdconf is configured correctly. (See chapter 4 for details)
- You can use *lpstat*(1) to get the current status of HP-UX spooling jobs. SRM/UX jobs will be spooled under the user id (Uid) that the file was opened with, which will be either the client `Uid` from the LAN-CLIENTS or SRM-CLIENTS Table of /etc/srmdconf, or the `Uid` from

the VOLUME-TABLE, if one was specified there. (See chapter 4 for details on the tables in /etc/srmdconf, and/or the *lpstat*(1) man page for details on checking the HP-UX spooling jobs status.)

- Files placed in the `Spool Directory` by processes other than *srmd*(1M) (e.g. BASIC/UX) will NOT be spooled by *srmd*(1M).

Specific Problems

Symptom 1: No output comes out on the printer or the plotter

The possible problems:

1. Is the HP-UX spooler up?
2. Are the spool files still in the `Spool Directory`?
3. Are the spool files in the ABORTED subdirectory of the `Spool Directory`, or in the ABORTED subdirectory of the PRIORITY subdirectory?
4. Are the `Spool Commands` from the SPOOL-TABLE in /etc/srmdconf working correctly?

The solutions:

1. Check to see if the HP-UX spooler is up:

- Use *lpstat*(1) to verify that the HP-UX spooler is running, and that the printer or plotter is accepting requests. (See your HP-UX System Administrator for help if *lpstat*(1) indicates a problem. Also, see chapter 4 of this manual for more details on the SRM/UX spooling environment.)

2. Check to see if the spool files are still in the `Spool Directory`:

If the spool files are still present in the `Spool Directory`, then the spool process may not have completed yet, or the client workstation may not have closed the file yet.

- To see if any spool processes are currently executing, from the server execute:

```
ps -ef | grep srmdlpfilter | grep -v grep
```

- For example, if /srmuxroot/PRINTER/JOB1 was still present, you could check to see if it is still opened by a client by executing from the server:

```
srmdstat -f /srmuxroot/PRINTER/JOB1
```

- □ If no SRM/UX client has the file open, the *srmdstat*(1) output will look like this:

  ```
  /srmserve/PRINTER/JOB1 not open
  ```

- □ If the file is still open, the output will look like this:

  ```
  /srmuxroot/PRINTER/JOB1 open by:
             XYZ at 15.99.99.10
  ```

Note Files are not spooled until the client closes the file.

3. Check to see if the spool files are in the ABORTED subdirectory of the `Spool Directory`, or in the ABORTED subdirectory of the PRIORITY subdirectory.

This could have happened because:

- The SRM/UX System Administrator stopped the server with *srmdshut*(1M)
- The SRM/UX System Administrator closed all of the client's files with *srmdreset*(1M)
- The client was rebooted or reinitialized

The server process will put messages in /usr/adm/srmd.log when it moves spool files to ABORTED subdirectories. (See chapter 4 for details about *srmdshut*(1M) and *srmdreset*(1M), and chapter 5 for details about rebooting and reinitialization.)

4. Check to see if the `Spool Commands` from the SPOOL-TABLE in /etc/srmdconf work as stand-alone commands:

- Suppose, for example, that your `Spool Command` is:

  ```
  srmdlpfilter | lp $PRIORITY
  ```

  - □ You could test the HP-UX spooling part of your `Spool Command` (if you wanted to simulate a normal priority spool job) by executing:

    ```
    cat <file> | lp
    ```

    where `<file>` is an ordinary HP-UX file.

- □ You could test the HP-UX spooling part of your Spool Command (if you wanted to simulate a PRIORITY spool job) by executing:

  ```
  cat <file> | lp -p2
  ```

  where <file> is an ordinary HP-UX file.

If no output appears on your printer or plotter after executing those two commands, then you may need to see your HP-UX System Administrator for assistance.

- If the normal and PRIORITY parts of your Spool Command work correctly, then you can test the entire command by executing:

  ```
  SRMSPOOLFILENAME=<file>; cat <file> | srmdlpfilter | lp
  ```

  where <file>'might be an ordinary HP-UX file, or it might be one of the special workstation types that your workstation was trying to spool.

Note When files are printed using *srmdlpfilter*(1) as a stand-alone command, they are NOT removed when the printing is completed. See chapter 4 and/or the man page for details on *srmdlpfilter*(1).

Other Problems

BASIC/UX and SRM/UX Conflict On SRM Cards

The SRM/UX server requires exclusive access to any SRM cards on the server computer that will be used to talk to SRM/UX clients. If BASIC/UX is being used on the same computer as the SRM/UX server, then BASIC/UX on that computer will not be able to use any SRM cards that *srmd*(1M) is using.

There is currently a defect that causes BASIC/UX to hang during bootup if there are SRM device files in /dev/rmb which have non-zero permissions, and which are currently in use by *srmd*(1M).

The work-around is to execute (from the server):

```
   chmod 0 /dev/rmb/srm*
```

before starting up BASIC/UX. (If you have multiple SRM cards in your server, and *srmd*(1M) is only using some of them, then BASIC/UX can still use the remaining SRM cards. In that case you would only set the permissions to 0 on the SRM device files in /dev/rmb that correspond to the SRM cards that *srmd*(1M)is using.)

Note This work-around will fail if you start up BASIC/UX as the super-user. If you need to run BASIC/UX as the super-user, you will have to remove all SRM device files from /dev/rmb that correspond to SRM cards that *srmd*(1M) uses. You will also have to do this after each *rmbconfig*(1).

C

SRM/UX Reference

SRM/UX Front Panel Interface Commands

Several new commands have been incorporated in SRM/UX which provide functionality similar to some of the SRM console commands.

Below is a list of the SRM/UX command "man-pages" that are also accessible on-line:

- srmd(1M)
- srmdconf(4)
- srmdconfcheck(1M)
- srmdlpfilter(1)
- srmdreconfig(1M)
- srmdreset(1M)
- srmdrestore(1)
- srmdshut(1M)
- srmdstat(1)

NAME

srmd - start the SRM/UX daemon

SYNOPSIS

srmd

DESCRIPTION

The SRM/UX daemon is a boot-server, file-server, and spooling-server for HP BASIC/WS and Pascal Workstation System clients.

The SRM/UX daemon is started by executing **/etc/srmd.** Usually this will be done in **/etc/rc** when the HP-UX system boots up.

Srmd logs errors, warnings, and general information to **/usr/adm/srmd.log** and spooling information to **/usr/lib/srmux/srmdspool.log.**

Only the superuser can execute *srmd.*

DIAGNOSTICS

Errors will occur under the following conditions:

If *srmd* is executed on a diskless cnode.

If another instance of *srmd* is already executing.

If the configuration file **/etc/srmdconf** does not exist, or contains fatal errors.

AUTHOR

Srmd was developed by HP.

FILES

/etc/srmdconf
/usr/adm/srmd.log
/usr/lib/srmux/srmdpipe
/usr/lib/srmux/srmd.lock
/usr/lib/srmux/srmdspool.log

SEE ALSO

srmdconf(4), srmdconfcheck(1M), srmdreconfig(1M), srmdreset(1M), srmdshut(1M), srmdstat(1).

NAME

srmdconf - SRM/UX server and client configuration file

SYNOPSIS

srmdconf

DESCRIPTION

The file **/etc/srmdconf** describes the SRM/UX file system, LAN clients, SRM clients and spooling environment. The file has five sections (VOLUME-TABLE, LAN-CLIENTS, SRM-CLIENTS, SPOOL-ENVIRONMENT and SPOOL-TABLE), each with its own format. The sections must be in the order listed above. A template *srmdconf* can be found in the **/etc/newconfig/srmux** directory. The file itself has the following format:

VOLUME-TABLE

VOLUME-TABLE entries (detailed below)

LAN-CLIENTS

LAN-CLIENTS table entries (detailed below)

SRM-CLIENTS

SRM-CLIENTS table entries (detailed below)

SPOOL-ENVIRONMENT

SPOOL-ENVIRONMENT table entries (detailed below)

SPOOL-TABLE

SPOOL-TABLE entries (detailed below)

Lines in *srmdconf* that start with "#" are comment lines.

A **VOLUME-TABLE** entry is described by a series of colon (:) separated fields, terminated by a newline character. The fields are:

Volume Name Emulated volume label of an SRM disk. The volume name can be up to 16 characters long.

Volume Number Emulated SRM disk volume number.

Uid User ID assigned by the *srmd* server process for files and directories created on this volume. Uid is an integer of up to five digits.

Gid Group ID assigned by the *srmd* server process for files and directories created on this volume. Gid is an integer of up to five digits.

Temp Directory Directory map for /WORKSTATIONS/TEMP_FILES.

Root Directory Path to SRM/UX root directory.

A LAN-CLIENT entry is described by a series of (:) separated fields, terminated by a newline character. The fields are:

Client Link Level Address
The ETHERNET address of the attached LAN card. This is a 12 character hexadecimal number.

Client Internet Address
Client's internet address.

Client Node Address
Client's emulated node address.

Server Node Address
Emulated node address of the server.

Client Name The name associated with this client. The client name can be up to 10 characters long.

Uid User ID assigned by the *srmd* server for files and directories created by this client. Uid is an integer of up to five digits.

Gid Group ID assigned by the *srmd* server for files and directories created by this client. Gid is an integer of up to five digits.

Umask HP-UX umask applied to new files and directories created by this client.

Volume List List of accessable volumes. Volume names must be found in the VOLUME-TABLE.

An SRM-CLIENT entry is described by a series of (:) separated fields, terminated by a newline character. The fields are:

Server's SRM Device File
Name of the SRM/UX server's SRM interface device file.

Client Node Address
Client's node address (set on the card's DIP switches)

Client Name The name associated with this client. The client name can be up to 10 characters long.

Uid User ID assigned by the *srmd* server for files and directories created by this client. Uid is an integer of up to five digits.

Gid Group ID assigned by the *srmd* server for files and directories created by this client. Gid is an integer of up to five digits.

Umask HP-UX umask applied to new files and directories created by this client.

Volume List List of accessable volumes. Volume names must be found in the VOLUME-TABLE.

A SPOOL-ENVIRONMENT entry assigns a value to an environment variable to be passed to the shell executing the spool command. The entry must contain an (=) sign to be valid. No other checking is done.

A SPOOL-TABLE entry is described by a series of (:) separated fields, terminated by a newline character. The fields are:

Spool Directory Absolute path to SRM/UX spooling directory.

Priority Priority assigned to jobs spooled to this spool device. Valid priorities are 1 through 7, inclusive. This field is optional.

Spool Command
Command to be executed by SHELL to accomplish spooling to this device.

Begin configuring SRM/UX with the template provided in **/etc/newconfig/srmux/srmdconf.** Then add volumes, LAN clients, SRM clients, spooling environment variables and spooling commands to your *srmdconf*. See chapter 4 of the *SRM/UX: System Administrator's and User's Guide* for details on how to build *srmdconf*.

Check the validity of your additions and/or changes to *srmdconf* with **/etc/srmdconfcheck.**

Finally, start the *srmd*(1M) server **/etc/srmd** or reconfigure the server using **/etc/srmdreconfig.**

DEPENDENCIES

Series 800

Entries in the SRM-CLIENTS section are not allowed. The series 800 does not support the SRM interface card.

AUTHOR

Srmdconf was developed by HP.

FILES

/etc/newconfig/srmux/srmdconf

SEE ALSO

rbootd(1M), srmd(1M), srmdconfcheck(1M), srmdreconfig(1M).

NAME

srmdconfcheck - validate a new SRM/UX configuration file

SYNOPSIS

srmdconfcheck [**-f** *file*]

DESCRIPTION

Srmdconfcheck validates a new SRM/UX configuration file. It processes **/etc/srmdconf**, unless a different file name is specified with the -f option. It reports all problems it finds in the configuration file on stdout.

The intended usage is that the SRM/UX System Administrator will edit **/etc/srmdconf** (or a copy in a different file name), validate it with *srmdconfcheck*, and then start up *srmd*(1M) with the new configuration file, or dynamically reconfigure *srmd*(1M) via *srmdreconfig*(1M).

Only the superuser can execute *srmdconfcheck*.

Options

-f *file* Check this *file* instead of **/etc/srmdconf**

RETURN VALUE

srmdconfcheck returns 0 if no errors are detected in the configuration file; otherwise it returns 1.

DIAGNOSTICS

Errors will occur under the following conditions:

If the configuration file to be checked does not exist, or cannot be opened for reading.

If *srmdconfcheck* detects an error in the specified configuration file. (See chapter 4 in the *SRM/UX: System Administrator's and User's Guide* for a complete description of the rules for SRM/UX configuration files.)

AUTHOR

Srmdconfcheck was developed by HP.

FILES

/etc/srmdconf

EXAMPLES

To validate your **/etc/srmdconf** file, execute the command:

```
/etc/srmdconfcheck
```

To validate a copy of the *srmdconf* file, execute the command:

```
/etc/srmdconfcheck -f srmdconf_copy
```

where *srmdconf_copy* is a copy of the *srmdconf* file.

SEE ALSO

srmd(1M), srmdconf(4), srmdreconfig(1M), srmdreset(1M), srmdshut(1M), srmdstat(1).

NAME

srmdlpfilter - translate a workstation file to an HP-UX file

SYNOPSIS

srmdlpfilter [**-rl**] [**-p** *"parms"*] [**-f** *file*]

DESCRIPTION

Srmdlpfilter provides file translation from HP BASIC/WS and Pascal/WS files to HP-UX ASCII files. The following types are supported: ASCII, BDAT, PROG, HP-UX, Pascal data and Pascal TEXT. The translated file is sent to standard output.

Options

-r Cause no translation and remove the workstation header

-l When used with the -r option prefixes the LIF file type and the LIF extension to the nontranslated workstation file. Useful when creating user-built filters for unsupported file types.

The -l option is ignored if the -r option is not specified.

-p *"parms"* Passes command line parameters to HP BASIC/UX when invoking HP BASIC/UX to translate HP BASIC PROG files.

-f *file* Specify the *file* to filter. Allows *srmdlpfilter* to be used as a standalone command.

EXAMPLES

Use *srmdlpfilter* in the SPOOL-TABLE section of **/etc/srmdconf** to send unfiltered binary output (e.g., dump graphics output from HP BASIC) to a laser printer:

```
srmdlpfilter -r | lp -oraw $PRIORITY
```

OR to send filtered output to a laser printer:

```
srmdlpfilter | lp $PRIORITY
```

Below is a simple use of *srmdlpfilter* to translate a workstation ASCII file to an HP-UX file:

```
cat /tmp/file.ASC | srmdlpfilter > /tmp/file.UX
```

To filter an unsupported file type:

```
srmdlpfilter -rl -f /usr/local/bin/myfile | myfilter
```

Srmdlpfilter will strip the workstation header and will prefix the LIF type and LIF extension fields so that the user-created filter *myfilter* can process a file type not supported by *srmdlpfilter.*

Filtering an HP BASIC/UX PROG file can be done by:

```
srmdlpfilter -p "-i -w3000k" -f ACKER
```

The above command attempts to start HP BASIC/UX with a workspace of 3000 kilobytes and run **/usr/lib/rmb/rmbclean** at startup time to filter the HP BASIC/UX PROG file. The filtered output will be sent to standard output. **Note, HP BASIC/UX must be installed on the SRM/UX server for** *srmdlpfilter* **to translate PROG files.**

AUTHOR

Srmdlpfilter was developed by HP.

FILES

/usr/bin/srmdlpfilter
/etc/srmdconf
/usr/bin/rmb

SEE ALSO

rmb(1), srmd(1M), srmdconf(4).

NAME

srmdreconfig - reconfigure the SRM/UX daemon

SYNOPSIS

srmdreconfig

DESCRIPTION

Srmdreconfig causes a currently executing SRM/UX daemon to reread **/etc/srmdconf** and reconfigure itself without having to shut down, and start over. This allows SRM/UX clients whose configurations are not modified by a revised **/etc/srmdconf** to continue their normal operations without an interruption caused by the SRM/UX daemon shutting down.

It is recommended that modifications to **/etc/srmdconf** be validated with *srmdconfcheck*(1M) before executing *srmdreconfig.*

Some modifications to the SRM/UX system configuration can be done only if the affected clients are not active. *Srmdreconfig* will report any modifications that it cannot carry out in the file **/usr/adm/srmd.log.** The SRM/UX administrator can choose to do an *srmdreset*(1M) on those clients which could not be changed, and then re-execute *srmdreconfig*, or the administrator can choose to wait until the next time that the SRM/UX daemon is halted by *srmdshut*(1M) and restarted for the new configuration to take full effect. (The SRM/UX daemon can be restarted by executing **/etc/srmd.**)

Only the superuser can execute *srmdreconfig.*

DIAGNOSTICS

Errors will occur under the following condition:

> If the SRM/UX daemon is not running when *srmdreconfig* is executed.
>
> *Srmdreconfig* will report any configuration changes that it could not carry out because of SRM/UX client activity in **/usr/adm/srmd.log.**

AUTHOR

Srmdreconfig was developed by HP.

FILES

/etc/srmdconf
/usr/lib/srmux/srmdpipe
/usr/adm/srmd.log

SEE ALSO

srmd(1M), srmdconf(4), srmdconfcheck(1M), srmdreset(1M), srmdshut(1M), srmdstat(1).

NAME

srmdreset - reset SRM/UX clients

SYNOPSIS

srmdreset [**-a**] [**-n** *name*] [**-s** *dev-file node*] [**-i** *internet*] [*files* ...]

DESCRIPTION

Srmdreset provides the SRM's REMOVE_USER functionality.

One of the options (**-a, -n, -s,** or **-i**) must be specified.

Options

-a specifies that all clients are to be reset. If no *files* are specified, then all open files for all clients are unlocked and closed. If one or more *files* are specified, then the specified *files* are unlocked and closed for all clients which have them open.

-n *name* specifies that user *name* is to be reset, where *name* is the client name from the LAN-CLIENTS or SRM-CLIENTS blocks of **/etc/srmdconf.** If no *files* are specified, then all of the client's open files are unlocked and closed; otherwise only the specified *files* are unlocked and closed.

-s *dev-file node* specifies that the SRM client with node number *node*, on the SRM card specified by *dev-file*, is to be reset. If no *files* are specified, then all of the client's open files are unlocked and closed; otherwise only the specified *files* are unlocked and closed.

-i *internet* specifies that the LAN client with internet address *internet* is to be reset. If no *files* are specified, then all of the client's open files are unlocked and closed; otherwise only the specified *files* are unlocked and closed.

Files in spooling directories will not be spooled when they are closed by *srmdreset.* If the spooling directory has a sub-directory "ABORTED", then the partial spool file will be moved to that directory (replacing any previous file with the same name). If there is no "ABORTED" sub-directory, or the move operation fails, then the partial spool file will be purged. A list of spool files moved to a "ABORTED" sub-directory, or purged, will be written to **/usr/adm/srmd.log**.

Only the superuser can execute *srmdreset*.

DIAGNOSTICS

Errors will occur under the following conditions:

If the SRM/UX daemon is not running when *srmdreset* is executed.

If the client specified by **-n, -s,** or **-i** cannot be found in the LAN-CLIENTS or SRM-CLIENTS tables of **/etc/srmdconf.**

If one or more of the *files* does not exist. *Files* that do exist will still be closed.

EXAMPLES

Execute:

```
/etc/srmdreset -n design47 /srmuxroot/STATTEST
```

The *name* option is specified. The above command closes the file **/srmuxroot/STATTEST** on client design47 and also releases any LOCKs that design47 had on this file. Any other clients that have the file open, will still have it open. Any other files that client design47 has open will remain open.

Execute:

```
/etc/srmdreset -i 15.99.99.17
```

The *internet* option is specified. The above command closes all files opened by the client whose internet address is 15.99.99.17. It will also release all LOCKs held by this client. All other clients are unaffected.

Execute:

```
/etc/srmdreset -a
```

The *all* option is specified. The above command closes all files that all clients have open.

AUTHOR

Srmdreset was developed by HP.

FILES

/usr/adm/srmd.log
/etc/srmdconf
/usr/lib/srmux/srmdpipe

SEE ALSO

srmd(1M), srmdconf(4), srmdreconfig(1M), srmdshut(1M), srmdstat(1).

NAME

srmdrestore - copy in files from SRM backup tapes

SYNOPSIS

srmdrestore [**-dmtuv**] [-M *mode*] *file* [*patterns*]

DESCRIPTION

Srmdrestore, with various options, provides a mechanism to import files from SRM 3.x backup tapes to SRM/UX.

The SRM backup will be read from *file*, which will typically be the name of a device file for a cartridge tape device.

Path names in the SRM backup are interpreted relative to the current directory.

Only files with names that match *patterns* are selected. *Patterns* are the same as the *sh*(1) command. (See *regexp*(5) under PATTERN MATCHING NOTATION.)

Multiple *patterns* can be specified. If no *patterns* are specified, all files are selected.

Files which are restored will be owned by the current user. The permissions will be set to octal 666 (777 for directories) as modified by his or her current umask, unless overridden by specifying the -M option. SRM passwords on the SRM backup tape are ignored.

The **-d**, **-m**, **-t**, **-u**, and **-v** options are based on *cpio*(1).

Options

-d Create directories as needed. If a directory, which appears as a component of the path name of a file to be restored, does not exist: create that directory.

-m Retain previous file modification time. Normally, the modification time will be set to the current time when the file is restored. If this option is specified, the modification time from file on tape will be restored instead. This option is ineffective on directories that are being restored.

-t Print only a catalog of the SRM backup. No files are created. Print the catalog in the identical format that the SRM uses. When combined, options -t and -v print the address of the file on the tape, flags indicating information about the file, and the tape of a link target.

-u Restore unconditionally. Normally, *Srmdrestore* will only overwrite files on the disc which are older than the one on the tape. If this option is specified, each restored file will replace the one on the disc, if the file already exists, regardless of which file is newer.

-v Be Verbose. *Srmdrestore* prints the full path name to stdout, *after* each file is restored. If combined with the -t option, -v causes the catalog to print additional information.

-M *mode* Set file mode. Change the restored files' permission mask to *mode. Mode* should be an octal number for the desired permission bits. The execute bit will be cleared for non-directory files being restored.

EXAMPLES

Execute:

```
srmdrestore -t /dev/rct
```

This command will catalog the backup tape on the tape drive specified by the device file **/dev/rct.**

Execute:

```
srmdrestore -dvm /dev/rct './user/AMEX.TEXT' './user/AMD.TEXT'
```

This command will restore two files. The single quote characters are used to prevent the HP-UX shell from incorrectly interpreting characters in the file name.

Execute:

```
srmdrestore -dvm /dev/rct '*.TEXT'
```

This command will restore all of the files (in any directory) that have names ending in ".TEXT".

AUTHOR

Srmdrestore was developed by HP.

NAME

srmdshut - terminate the SRM/UX daemon

SYNOPSIS

srmdshut

DESCRIPTION

Srmdshut provides the SRM's SYSTEM DOWN functionality. Note that the SRM's powerfail option is NOT supported.

Srmdshut will do a "graceful" shutdown of the SRM/UX daemon. Incoming client requests will be disabled, and then all open files will be unlocked and closed before terminating the daemon process. The daemon will wait for a brief time for spooling activity to complete so that spool files can be removed from spool directories after they have been spooled. If spooling operations have not completed when the daemon terminates, then spool files will remain in the spool directory. (A list of these files will be put in **/usr/adm/srmd.log** before the daemon terminates.) *Srmdshut* will wait for the daemon to terminate.

Files in spooling directories will not be spooled when they are closed by *srmdshut*. If the spooling directory has a sub-directory "ABORTED", then the partial spool file will be moved to that directory (replacing any previous file with the same name). If there is no "ABORTED" sub-directory, or the move operation fails, then the partial spool file will be purged. (A list of spool files moved to a "ABORTED" subdirectory, or purged, will be written to **/usr/adm/srmd.log** before the daemon terminates.)

The SRM/UX daemon can be restarted by executing **/etc/srmd.**

Only the superuser can execute *srmdshut*.

DIAGNOSTICS

Errors will occur on the following condition:

> If the SRM/UX daemon is not running when *srmdshut* is executed.

AUTHOR

Srmdshut was developed by HP.

FILES

/usr/adm/srmd.log
/usr/lib/srmux/srmdpipe

SEE ALSO

srmd(1M), srmdreconfig(1M), srmdreset(1M), srmdstat(1).

NAME

srmdstat - obtain status information on SRM/UX clients

SYNOPSIS

srmdstat [**-n** *name* | **-s** *dev-file node* | **-i** *internet*]

srmdstat [**-f** *files* ...]

srmdstat [**-u**]

DESCRIPTION

Srmdstat, with various options, provides the SRM's FILES, USERS, and NODES functionality.

Options

No more than 1 option is allowed per execution of *srmdstat*.

If no option is specified, *srmdstat* outputs a list of all active clients, and lists all of the files each client has open.

-n *name* lists the files that user *name* has open, where *name* is the client name from the LAN-CLIENTS or SRM-CLIENTS tables of **/etc/srmdconf**.

-s *dev-file node* lists the files that the SRM client with node number *node*, on the SRM card specified by *dev-file*, has open.

-i *internet* lists the files that the LAN client with internet address *internet* has open.

-f *files* lists the clients which have the specified *files* open.

-u lists all clients which have open files and lists how many files each client has open.

DIAGNOSTICS

Errors will occur under the following conditions:

If the SRM/UX daemon is not running when *srmdstat* is executed.

If the client specified by **-n**, -s, or **-i** cannot be found in the LAN-CLIENTS or SRM-CLIENTS tables of **/etc/srmdconf**.

If one or more of the *files* specified by **-f** does not exist. *Files* that do exist will still be listed.

EXAMPLES

Execute:

```
/etc/srmdstat -u
```

The **-u** option produces a condensed version of the default *srmdstat* output; it reports only those clients that have open files, and tells how many files each client has open.

Execute:

```
/etc/srmdstat -n design33
```

The **-n** (*name*) option reports the status of a single client, the one with the specified name. *Srmdstat* reports the same information (for the named client) with this option as it reports with no options specified.

Execute:

```
/etc/srmdstat -i 15.99.99.42
```

The **-i** (*internet*) option reports the status of a single client, the one with the specified internet address. *Srmdstat* reports the same information (for the specified client) with this option as it reports with no options specified.

Execute:

```
/etc/srmdstat -f /srmuxroot/STATTEST
```

The **-f** option is used to show which clients are using particular file(s). It also tells which client, if any, has a file LOCKed.

The SRM/UX System Administrator can use this option to see if one client is holding onto a LOCK inappropriately.

AUTHOR

Srmdstat was developed by HP.

FILES

/etc/srmdconf
/usr/lib/srmux/srmdpipe

SEE ALSO

srmd(1M), srmdconf(4), srmdreconfig(1M), srmdreset(1M), srmdshut(1M).

Glossary

Note More general definitions of some of the following terms, as well as definitions of related terms, can be found in the following HP-UX documentation:

- Section 9: Glossary of the *HP-UX Reference*
- The Glossary of the *HP-UX System Administration Concepts* manual
- The Glossary of the *ARPA Services/300 User's Guide*
- The "NFS Services Overview" chapter of the *Installing and Administering NFS Services* manual.

cluster server (also called root server)

The HP-UX system which acts as a boot and file system server for diskless HP-UX systems. The SRM/UX server may also be a cluster server.

daemon

A process that runs quietly in the background, doing some useful task. For example, the SRM/UX server process, *srmd*(1M), is a daemon.

ENVIRONMENT file

A file that contains printer or plotter control information. This file can be prefixed to a spool file, in which case it defines the print or plot environment in which the spool file is output.

file system

A collection of files and supporting data structures residing on a mass storage volume. In this manual, the HP-UX file system refers to the collection of all local mass storage volumes and NFS file systems mounted under the root directory on the HP-UX computer which is acting as an

SRM/UX server. The SRM/UX file system refers to the collection of one or more subtrees of the HP-UX file system that the SRM/UX System Administrator chooses to make available to SRM/UX clients.

front-panel commands

A set of commands, used by the SRM/UX System Administrator to perform SRM/UX administration tasks. They include *srmdconfcheck*(1M), *srmdreconfig*(1M), *srmdreset*(1M), *srmdshut*(1M), and *srmdstat*(1). They are known as the front-panel commands for two reasons:

- They emulate some of the SRM's console ("front-panel") commands which the administrator uses to administer the SRM.
- They provide a "virtual front-panel" for the SRM/UX Administrator to communicate with the *srmd*(1M) process which is running in the background on HP-UX, and cannot be communicated with directly from a terminal or window.

HP-UX root directory

The highest level directory of the HP-UX file system, from which all other files branch. In HP-UX, the slash (/) character refers to the root directory. The root directory is the only directory in the file system that is its own parent directory. (See also **SRM root directory**)

HP-UX spooling

The HP-UX spooling system controls output requests for printers and plotters, so that multiple programs do not output to the same physical device simultaneously. (See the "Managing the LP Spooler" chapter of the *HP-UX System Administration Tasks* manual for more information on HP-UX spooling.) (See also **SRM/UX spooling**)

HP-UX System Administrator

The HP-UX System Administrator installs, configures, and administers HP-UX. The SRM/UX System Administrator will work with the HP-UX System Administrator to configure the HP-UX kernel, the HP-UX file system, the HP-UX spoolers, and the HP-UX network to meet the needs of the SRM/UX system configuration.

Internet Address (also IP Address)

A four-byte address used by NS_ARPA Services and SRM/UX. It consists of two parts: a network number and a host number. For example, an Internet Address of 15.99.99.1 has a network number of 15.99.99 and a host number of 1. Nodes on the same LAN subnet will have the same network number and distinct host numbers.

LAN interface (or LAN card)

Refers to the built-in LAN interface on some Series 300 and Series 400 computers, or to one of the following HP LAN card part numbers: HP98643A, HP98171A, HP36967A, or HP91786B.

Link Level Address (or LLA)

A unique 12-digit hexadecimal number which is part of every LAN card.

NFS (Network File System)

An industry standard system for allowing many systems to share the same files. The SRM/UX System Administrator may chose to give SRM/UX clients access to files that the SRM/UX server accesses via NFS Remote File Access. (See the *Installing and Administering NFS Services* manual.) (NFS is a trademark of Sun Microsystems, Inc.)

Node Address

A Node Address is a 2-digit number that you can set with switches on the 50962A or 98629A SRM cards, and is emulated for LAN card clients by SRM/UX.

parent directory

A directory's parent directory is the directory one level above it in the file hierarchy, with the exception of the directory entries for . (dot) and .. (dot-dot). All directories except the root directory (/) have one (and only one) parent directory.

process

An invocation of a program. For example, the SRM/UX server process is an invocation of the program /etc/srmd.

root server

See **cluster server**.

root user

See **super-user**.

shared file

A file which can be concurrently accessed by multiple SRM/UX clients. May also refer to a file which is accessed by one or more cooperating HP-UX processes concurrently with one or more SRM/UX clients.

Shared Resource Multiplexer

A switching device that controls message flow between computers, allowing one computer to send information, while others receive. It is usually referred to as a multiplexer or a mux. Each SRM multiplexer (HP 98028A) in the system connects one or more workstation computers (up to four) to one or more servers (up to four). This device is now obsolete.

SRM

Stands for Shared Resource Management, typically used to refer to the SRM server (HP50960A), its operating system software, and the peripherals attached to the SRM server. SRM/UX is an emulation of the SRM on HP-UX.

SRM card (or SRM interface)

Refers to one of the following HP SRM card part numbers: HP98629A, HP98629 + HP50961U combination, or HP50962A.

SRM Coax Adapter

Refers to the HP 50961 Resource Management Coax Adapter. This adapter attaches to an HP 98629 to allow the use of this SRM interface on a coax connection.

SRM coax connection

Describes the SRM hardware configuration that uses SRM coax interfaces or HP 50961 Resource Management Coax Adapters at the client and server workstations, and uses coax cabling to connect the client(s) and server(s) in a bus fashion.

SRM coax interface

Refers to the HP50962A SRM card. This interface is functionally equivalent to the HP98629 SRM card and HP50961U Coax Adapter combination, which is also used on a coax connection.

SRM file name

A string containing between 1 and 16 characters naming a file.

SRM multiplexer connection

Describes the SRM hardware configuration in which the HP 98028A Resource Management Multiplexer and HP 97061 cables are used to connect clients and the SRM server. This configuration is obsolete.

SRM password

A string containing between 1 and 16 characters which the SRM uses to limit access capabilities to SRM files and directories. SRM/UX ignores SRM passwords. The HP-UX read, write, and execute permissions control access capabilities to SRM/UX files and directories.

SRM root directory

The highest level directory of an SRM volume, from which all other files on that volume branch. In SRM, the slash (/) character refers to the root directory. The root directory is the only directory on an SRM volume that is its own parent directory. (See also HP-UX root directory)

SRM volume

A collection of files and supporting data structures residing on a mass storage volume. This will usually correspond to one physical disk attached to the SRM. The SRM/UX System Administrator configures the SRM/UX server to emulate one or more SRM volumes by designating sub-trees of the HP-UX file system as SRM/UX volumes.

SRM/UX client

An SRM/UX client is an HP Series 200 or 300 workstation executing HP Basic/WS or the Pascal Workstation System, which uses the SRM/UX server for booting, file system requests, and spooling.

SRM/UX configuration file

The SRM/UX System Administrator uses the file /etc/srmdconf on the SRM/UX server to configure SRM/UX.

SRM/UX spooling

A method by which a file can be sent to a shared printer or plotter by means of a spooling directory. In SRM/UX, the file is usually sent to one of the HP-UX spoolers for actual output on a printer or plotter. Once the file is written to the spooling directory, the client workstation is free to perform other tasks. (See **HP-UX spooling**)

SRM/UX server

An SRM/UX server is an HP Series 300, 400 or 800 HP-UX computer running a special application (*srmd*(1M)) that provides booting, file system, and spooling services to SRM/UX clients. It provides these services such that in most cases SRM/UX clients do not have to know that they are using an SRM/UX server instead of an SRM.

SRM/UX System Administrator

The SRM/UX System Administrator installs, configures, and administers the SRM/UX system. This person will work with the HP-UX System Administrator on tasks which involve the underlying HP-UX configuration on the SRM/UX Server.

sub-directory

A directory that is one (or perhaps more) levels lower in the file system hierarchy than a given directory. Sometimes called a subordinate directory.

subnet

A network consisting of nodes with the same network number in their Internet Addresses. SRM/UX clients which use the LAN connection to access their SRM/UX server must be on the same subnet as the server. (See **Internet Address**)

super-user

The HP-UX System Administrator. This user has access to all files, and can perform privileged operations. The super-user has a real and effective user ID of 0, and, by convention, the user name of *root*. (The SRM/UX System Administrator will need to have super-user capability, even if he or she is not the System Administrator for the whole HP-UX system which is acting as the SRM/UX server.)

workstation

In this manual a workstation refers to either an SRM/UX server workstation, or an SRM/UX client workstation.

Index

HEWLETT
PACKARD

HP Part Number
E2085-90000

Printed in U.S.A. E0191

E2085-90600 Manufacturing Number