Netra High Availability Suite Foundation Services 2.1 6/03 SMCT Installation Guide

---

Sun Microsystems, Inc.
4150 Network Circle
Santa Clara, CA 95054
U.S.A.

Part No: 817–1766–11
September 2004

Adobe PostScript

040812@9495

Contents

Tables

Figures

Examples

Preface

Caution – Do not use the SMCT tools with the current version of the product.

The *Netra High Availability Suite Foundation Services 2.1 6/03 SMCT Installation Guide* explains how to use the SMCT. The SMCT enables you to install, configure, and deploy the Netra™ High Availability (HA) Suite Foundation Services 2.1 6/03 product with associated user applications on one or more clusters.

The SMCT enables you to create a deployable cluster configuration that can be reused with multiple clusters. However, to customize and fine-tune the components of the Foundation Services that you install on a cluster, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Custom Installation Guide*.

Who Should Use This Book

This book is for system administrators who are responsible for the installation and deployment of the Foundation Services software and associated user applications onto a cluster. This guide includes information related to the following tasks:

- How to plan your cluster addressing, disk layout, file system configuration, and hardware and software requirements
- How to configure the software to be deployed by using the SMCT commands
- How to verify that the software configuration is valid
- How to deploy the software to one or more clusters

Before You Read This Book

To configure and deploy the Foundation Services, you must have experience with the following:

- Solaris™ operating system installation tools
- Cluster environments
- Networking concepts

Before reading this book, consult the *Netra High Availability Suite Foundation Services 2.1 6/03 Overview* for an introduction to the Foundation Services.

Related Documentation

You will require some of the following books from the Foundation Services documentation set:

- *Netra High Availability Suite Foundation Services 2.1 6/03 Overview*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Glossary*
- *What's New in Netra High Availability Suite Foundation Services 2.1 6/03*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Quick Start Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Custom Installation Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Cluster Administration Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Troubleshooting Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 CMM Programming Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 NMA Programming Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Reference Manual*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Standalone CGTP Guide*
- *Netra High Availability Suite Foundation Services 2.1 6/03 Release Notes*
- *Netra High Availability Suite Foundation Services 2.1 6/03 README*

The following books are not part of the Foundation Services documentation set, but these books provide related information:

- *Solaris System Administration Guide, Volume 2*
- Solaris 8 Installation Collection
- Solaris 9 Installation Collection

- Sun StorEdge™ Network Data Replicator (SNDR) 3.1 documentation
- Java™ Dynamic Management Kit (DMK) 5.0 documentation
- Netra CT 410 and Netra CT 810 hardware documentation

How This Book Is Organized

This book is divided into parts that contain related chapters. Two appendixes are also provided.

Part I enables you to choose an appropriate installation method. This part also provides an overview of the process for installing and deploying the software on a cluster by using the SMCT.

- Chapter 1 describes how to choose the installation method best suited to your requirements.
- Chapter 2 outlines the three stages of the installation and deployment process by using the SMCT.

Part II describes the requirements for the installation process.

- Chapter 3 explains how to install the hardware and software.
- Chapter 4 describes what you need to plan before starting the configuration stages and deploying the software load to a cluster.

Part III describes the three software configuration stages.

- Chapter 5 describes stage 1 of the deployment process. This chapter explains how to configure the generic configuration files and how to create generic flash archives for the cluster.
- Chapter 6 describes stage 2 of the deployment process. This chapter explains how to add user-defined configuration data to the generic flash archives. This stage is optional.
- Chapter 7 describes stage 3 of the deployment process. This chapter explains how to link flash archives to specific cluster hardware and network definitions for deployment across the cluster.

Part IV explains how to deploy a software load across a cluster by using Solaris JumpStart™.

- Chapter 8 describes how to do the following:
  - Verify that the flash archive is valid
  - Create the Solaris JumpStart environment for the target cluster
  - Deploy the software load on the target cluster by using Solaris JumpStart
- Chapter 9 lists the reconfiguration options that are available.

The appendixes provide the following reference material to enable you to configure the SMCT.

- Appendix A lists the elements of the configuration files.
- Appendix B provides examples of the disk layout and file system configuration sections of the `machine.conf` configuration file.

Accessing Sun Documentation Online

The docs.sun.com^{SM} Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Ordering Sun Documentation

Sun Microsystems offers select product documentation in print form. For a list of documents and how to order them, see “Buy printed documentation” at `http://docs.sun.com`.

Typographic Conventions

The following table describes the typographic changes that are used in this book.

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories, and onscreen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

TABLE P–1 Typographic Conventions *(Continued)*

Typeface or Symbol	Meaning	Example

`AaBbCc123`	What you type, contrasted with onscreen computer output	`machine_name%` **`su`**
		`Password:`

AaBbCc123	Command-line placeholder: replace with a real name or value	The command to remove a file is `rm` *filename*.
AaBbCc123	Book titles, new terms, and terms to be emphasized	Read Chapter 6 in the *User's Guide*. These are called *class* options. Do *not* save the file. (Emphasis sometimes appears in bold online.)

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

PART I

Introduction to the Installation and Deployment Process

Caution – Do not use the SMCT tools with the current version of the product.

Before you begin to install and deploy the software on your cluster, you must choose a method of installation. If you choose to install and deploy the software by using the SMCT, familiarize yourself with the SMCT installation and deployment process.

For details about these topics, see the following chapters:

- Chapter 1 describes how to choose an installation method.
- Chapter 2 describes the stages for installing and deploying the software by using the SMCT. The tasks and results for each stage are also described in this chapter.

CHAPTER **1**

Planning Your Installation

To successfully install the Solaris operating system and the Foundation Services on your cluster, first choose an installation method best suited to your requirements.

The Foundation Services product provides three ways to install the software on the cluster:

- **SMCT installation.** The *SMCT* enables you to generate a cluster configuration that can be installed and deployed on multiple nodes groups or on multiple clusters. You can install clusters containing *master-eligible nodes* and *diskless nodes*, or *dataless nodes*.
- **Installation with the `nhinstall` tool.** The `nhinstall` tool enables you to install the Foundation Services on a cluster. This tool is flexible and provides various configuration options that you can adapt to your requirements. However, the `nhinstall` tool only supports the installation of master-eligible nodes and diskless nodes.
- **Manual installation.** You can manually install the software on a cluster containing master-eligible nodes and diskless nodes or dataless nodes.

  Manual installation provides greater flexibility when installing components of the Foundation Services. However, a manual installation can result in a cluster configuration that is not easily reproducible on other clusters.

For more information on choosing an installation method, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Overview*.

CHAPTER **2**

Introduction to the SMCT Installation and Deployment Process

The SMCT installation and *deployment* is a three-stage process. At each stage, you configure the cluster environment, including the software, to generate flash archives. The software distributions and configuration files are used by the SMCT commands to generate the flash archives and software loads.

For details on the process of installing and deploying the software by using the SMCT, see the following sections:

Overview of the SMCT Installation and Deployment Process

To install and deploy the software by using the SMCT, you need a *build server*, a *prototype machine*, and an *installation server*. One system can be used both as an installation server and as a build server. A master-eligible or dataless node from the cluster can be used as a prototype machine. For more information on the hardware required, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

The following diagram illustrates the configuration stages of the installation and deployment process. Each configuration stage is described in detail in the following sections.

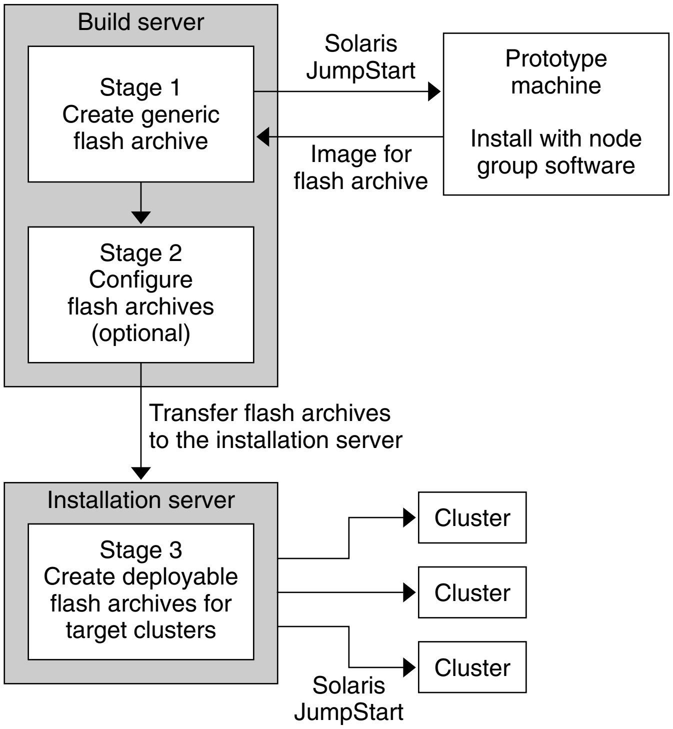

FIGURE 2–1 Overview of the SMCT Installation and Deployment Process

Configuration Stage 1: Creating a Generic Flash Archive

In stage 1 of the installation and deployment process, you configure your cluster environment and generate a *generic flash archive*. A generic flash archive is not adapted to any particular cluster and is not deployable.

The following diagram illustrates stage 1.

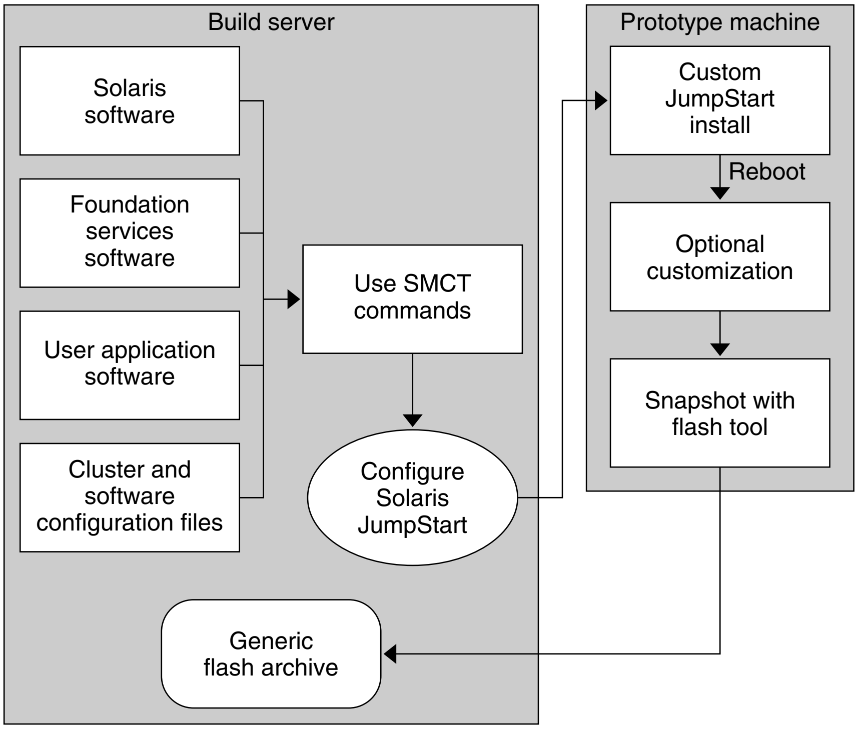

FIGURE 2–2 Overview of Configuration Stage 1

Stage 1 Configuration Tasks

To complete stage 1, you perform the following configuration tasks. You need the indicated software and configuration files to create the software repository, software load respository, and the SMCT environment.

1. You need the software distributions that are to be installed on your cluster:
   - Foundation Services packages and patches, consisting of the runtime packages and patches supplied with the Foundation Services.
   - Solaris packages and patches, which are in the Solaris distribution that is included as part of the Solaris operating system.
   - Optionally, user application packages and patches. These packages and patches are processed as Solaris add-on products.
   - Optionally, third-party packages and patches. These packages and patches are processed as Solaris add-on products.

   For information about installing the distribution software, see "Installing Software on the Installation Hardware" on page 39.

2. You configure the following *cluster configuration files* to define your cluster topology and hardware environment:

   - Cluster model configuration file

     The `cluster.conf` configuration file contains a logical view of the cluster in terms of nodes, *node groups*, domain, and services. Each node in the cluster belongs to a *node group*. A node group is a group of nodes of the same type, with the same configuration. A node group can contain master-eligible nodes, diskless nodes, or dataless nodes, but not a combination of diskless nodes and dataless nodes.

     For more information, refer to the `cluster.conf`(4) man page.

   - Machine model configuration file

     The `machine.conf` configuration file describes the cluster hardware and hardware components. This file also contains definitions of disk partitions and file systems for master-eligible nodes and dataless nodes. For more information, refer to the `machine.conf`(4) man page.

   For information about how to configure these files, see Chapter 5. For information about defining node groups, see "Defining Nodes and Node Groups" on page 66.

3. You also configure the Solaris JumpStart utility to define the Solaris JumpStart environment for the prototype machine.

4. Finally, you configure the *software configuration files*. These files enable you to define the packages and patches for additional applications that you might want to install on node groups in a cluster. For more information on software configuration files, refer to the `software.conf`(4) man page. See also "Creating a Software Configuration File for a Node Group" on page 74.

5. After you define your cluster and software in the configuration files, you use the following SMCT commands to create a *software load* that contains the Foundation Services, the Solaris operating system, and optionally, user applications:

   - You run the following commands each time you create a new SMCT environment:

     `nhsmctsetup`

   - You run the following commands each time you create a generic flash archive:

     - `slcreate`
     - `slexport`
     - `flinstall`
     - `flcreate`

   The software load contains the software for each node group. The node group software is transferred to the prototype machine from the build server by using the Solaris JumpStart mechanism. The Solaris environment, the Foundation Services packages, and any user applications are installed on the prototype machine, but they are not started.

Stage 1 Results

At the end of stage 1, the following elements are created:

- A generic flash archive is created and contains a snapshot of all the software installed on the prototype machine.

  The *flash archive* is created using standard Solaris Flash tools. The flash archive can be deployed on any machine with a similar architecture to the machine on which the flash archive was created.

  A flash archive is required for each master-eligible node group or dataless node group that is to be deployed to the cluster. The following rule determines the number of flash archives required for a cluster:

  (Flash Archive) = 1 + N

  - *1* is the mandatory flash archive associated with the master-eligible node group. This flash archive can embed the environment for diskless node groups.
  - *N* is the number of dataless node groups associated with the cluster.

  A generic flash archive has the following characteristics:

  - Is site independent.
  - Is not adapted to any particular cluster and is, therefore, not deployable.
  - Helps generate deployable flash archives for a set of similar clusters. These flash archives are based on the topology defined in the `cluster.conf` file.
  - Is created by using the `flcreate` command.
- A software repository and a software load repository are created.

  The *software repository* is the largest repository in the deployment environment. The software repository stores the software packages that are to be deployed to the cluster. The software repository includes the Foundation Services packages, patches, and any user applications and user-defined configuration data. By having all the software packages in one repository, flash archives can be rebuilt if necessary. In addition, several versions of the same packages can be managed from the software repository location.

  The *software load repository* stores the software loads that have already been created. In addition, the software load repository stores the Foundation Services and user-defined configuration data associated with these software loads. A software load can be restored by using the data stored in this repository.

  The software repository and software load repository are located on the build server. Both of these repositories are managed by the SMCT software load commands, that is, the `sl`*xxx* commands.
- The Solaris JumpStart configuration files that are used by the Solaris JumpStart mechanism to deploy node group software to the prototype machine are created.

For more information on stage 1, see Chapter 5.

Configuration Stage 2: Adding User-Defined Data to the Flash Archive

If you do not plan to deploy user applications, go to stage 3. See "Configuration Stage 3: Configuring the Flash Archive for Deployment" on page 31.

In stage 2, you define data specific to user applications in a configuration file. These applications are to be deployed to node groups on your cluster and are included in the generic flash archive. You use the SMCT commands to add this data to the generic flash archive.

The following diagram illustrates stage 2.

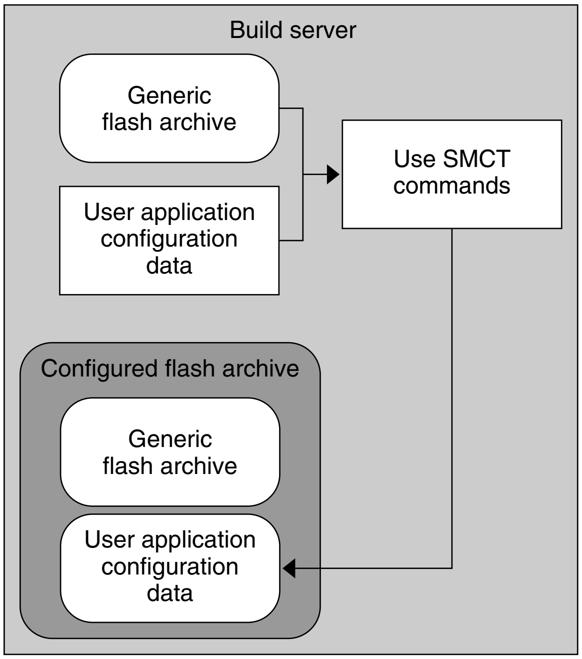

FIGURE 2–3 Overview of Configuration Stage 2

The user-defined configuration data consists of user application data and installation scripts. These installation scripts add the user application data at deployment time. There is one *user application configuration file* for each node group. For more information, refer to the `userapp.conf`(4) man page.

After you configure the `userapp.conf` file, you use the `slconfig`, `slexport`, and `flconfig` commands to add the user-defined configuration data to the generic flash archive.

configured flash archive. At the end of stage 2, you have a A configured flash archive is a generic flash archive that also contains user applications and user-defined configuration data. In addition, the software load repository on the build server is updated with the user-defined configuration data.

For more information on stage 2, see Chapter 6.

Configuration Stage 3: Configuring the Flash Archive for Deployment

In stage 3, you configure the flash archive to prepare to create the *deployable flash archive* for the cluster. The following diagram illustrates stage 3.

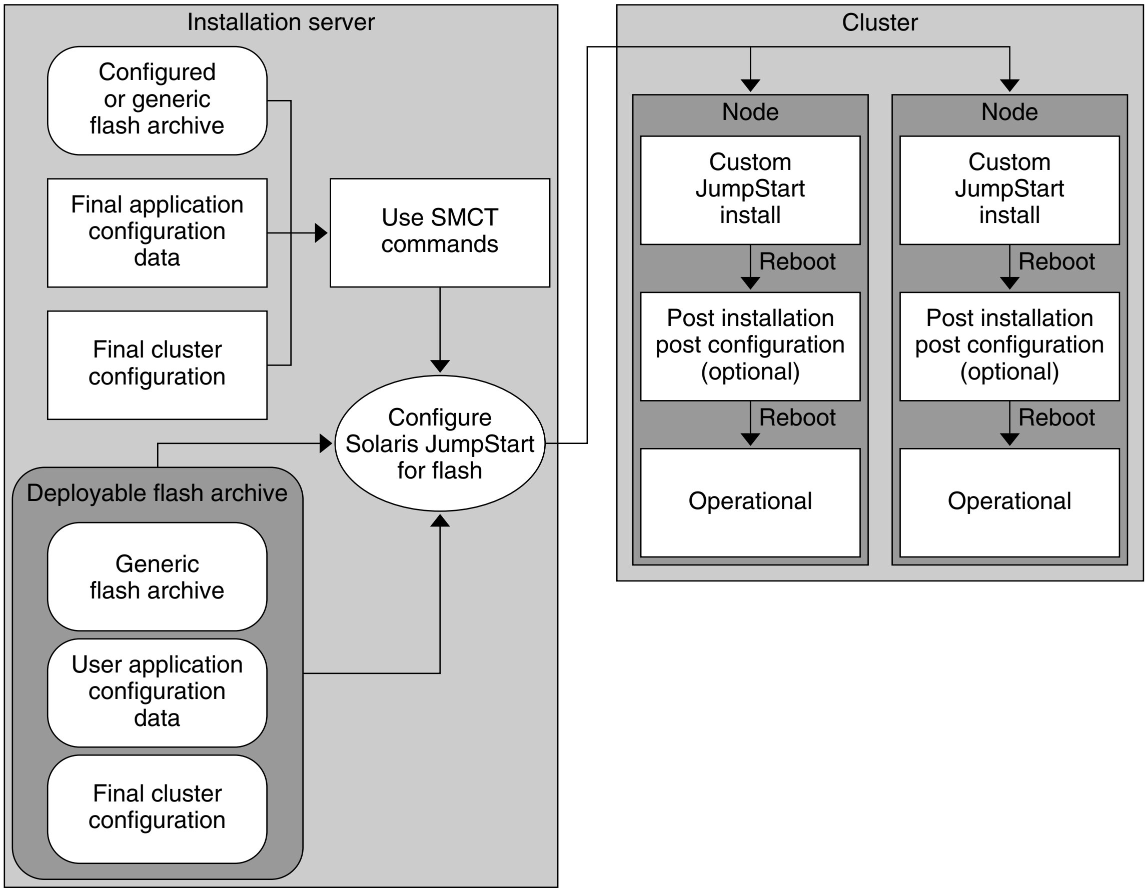
Installation server
Configured or generic flash archive
Final application configuration data
Final cluster configuration
Use SMCT commands
Configure Solaris JumpStart for flash
Deployable flash archive
Generic flash archive
User application configuration data
Final cluster configuration
Cluster
Node
Custom JumpStart install
Reboot
Post installation post configuration (optional)
Reboot
Operational
Node
Custom JumpStart install
Reboot
Post installation post configuration (optional)
Reboot
Operational

FIGURE 2–4 Overview of Configuration Stage 3

Stage 3 Configuration Tasks

To complete stage 3, you perform the following tasks:

1. You transfer the SMCT environment from the build server to the installation server.
2. You modify the cluster configuration files, `cluster.conf` and `machine.conf`, to add specific target cluster data to the generic or configured flash archive. In addition, you configure the network model configuration file, `network.conf`, to define the network environment of your cluster. The `network.conf` configuration file is associated with the network model. For more information, refer to the `network.conf`(4) man page.
3. If you have a configured flash archive, you can modify the user-defined configuration data in order to manage application-to-site dependencies. For more information, refer to the `userapp.conf`(4) man page.

4. You use the `sldeploy`, `slexport`, and `fldeploy` SMCT commands to create the deployable flash archive.

Stage 3 Results

At the end of stage 3, the following elements are created or updated:

- A deployable flash archive is created for each master-eligible node group or dataless node group. Each node group has a cluster-specific configuration. Diskless node groups are embedded in the master-eligible node groups. The deployable flash archives are created by using the `fldeploy` command.
- The Solaris JumpStart configuration files are created for each node of each master-eligible node group and dataless node group in the cluster. These files are used by the Solaris JumpStart mechanism to deploy the software load to the target cluster.
- The software load repository is updated.
- The Foundation Services configuration files are generated.

  The configuration files for the Foundation Services are used to configure the individual Foundation Services, such as Reliable NFS and Cluster Membership Manager (CMM).

For more information on stage 3, see Chapter 7.

Note – After completing the three configuration stages and deploying the Foundation Services to the target cluster, you can rerun the configuration stages individually. The redeployment depends on the reconfiguration that you want to perform. For more information on rerunning the deployment stages, see Chapter 9.

PART II

Preparing Your Deployment Environment

This part describes requirements for installing and deploying the software on the cluster. This part contains the following chapters:

- Chapter 3 describes how to install the necessary hardware and software.
- Chapter 4 describes what you need to plan before starting the configuration stages and deploying the software load to a cluster.

CHAPTER **3**

Installing the Hardware and Software

This chapter describes how to install the installation hardware and software:

- "Hardware and Software Requirements" on page 37
- "Installing Software on the Installation Hardware" on page 39

Hardware and Software Requirements

For information about how to install and configure the cluster and installation hardware, refer to the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*. For information about how to install the Foundation Services software, see "Installing Software on the Installation Hardware" on page 39.

The following figure illustrates the hardware required:

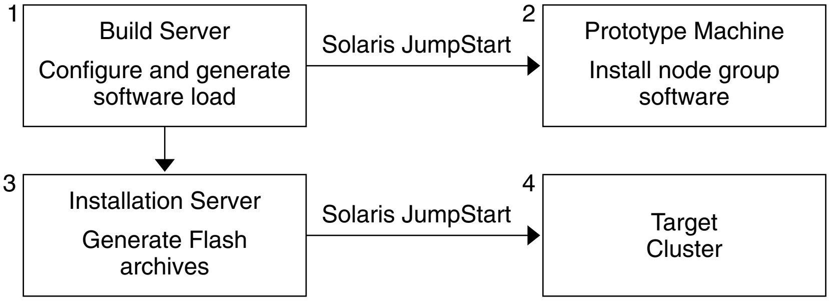

FIGURE 3–1 Hardware Requirements for the Deployment Environment

---

Note – For simplicity, this guide describes the build server and installation server as separate machines. However, it is possible to set up your deployment environment with the build server and installation server on the same machine.

Similarly, the prototype machine can be the same hardware that is used in the target cluster.

Cluster Hardware Requirements

The cluster hardware is a collection of peer nodes connected together by a high-speed redundant local network such as a redundant, switched Ethernet network. The Foundation Services have been tested on Netra™ servers installed with the Solaris operating system.

For a detailed description of the required cluster hardware, refer to the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*

Cluster Software Requirements

The Foundation Services run on a cluster installed with the Solaris operating system.

Solaris 8 2/02, Solaris 8 Platform Specification Release (PSR) 1, Solaris 9, and Solaris 9 9/02 are the supported versions. For information about the Solaris versions supported on particular hardware, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

Install the Solaris operating system and other supplementary software to enhance the *reliability*, *availability*, and *serviceability* of your cluster.

You install the following software on the nodes of the cluster:

- Software included in the Solaris distribution
  - Java™ Software Development Kit (SDK) Standard Edition for all nodes
    - Version 1.3.1 for Solaris 8 2/02 and above
    - Version 1.4.0 for Solaris 9 and above
  - Volume management software for master-eligible nodes
    - Solstice DiskSuite™ 4.2.1 for Solaris 8 2/02 and above
    - Solaris Volume Manager for Solaris 9 and Solaris 9 9/02 and above
  - Solaris Dynamic Host Configuration Protocol (DHCP) for diskless nodes
- Software included in the Foundation Services distribution
  - Java Dynamic Management Kit (DMK) 5.0

- Sun StorEdge™ Network Data Replicator (SNDR) Version 3.1

Installation Hardware and Software Requirements

The installation hardware consists of servers used to build, install, and deploy the software on a cluster. The installation hardware includes a build server, a prototype machine, and an installation server. The installation hardware and software requirements are described in the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

Installing Software on the Installation Hardware

When you have installed and have configured the installation hardware, install the Foundation Services software as described in this section.

Installing Software on the Build Server

This section describes how to install the Foundation Services and the Solaris operating system software on the build server.

▼ To Install the Foundation Services Software on the Build Server

1. **Copy the Foundation Services packages to a temporary directory on the build server.**

   This step is not necessary if the packages are available from a CD-ROM directory or accessible by means of an NFS mounted directory.

   a. **If the Foundation Services packages are in an electronic distribution, copy them to the local disk on the build server.**

   b. **Ensure that you have access to any required patches or third party packages. For information, see the** *Netra High Availability Suite Foundation Services 2.1 6/03 Release Notes***.**

2. **Copy any user applications to a temporary application directory on the build server.**

A software distribution containing user applications can be packaged as a Sun package or an archive file, or unpacked as a file tree.

This step is not necessary if the user applications are available by means of an NFS mounted directory.

3. **Install the Foundation Services package, `SUNWnhsmc`, on the build server:**

   ```
   # pkgadd -d electronic_distribution_directory/NetraHAS2.1/Packages SUNWnhsmc
   ```

   Only the `SUNWnhsmc` package is installed on the build server. Other packages are distributed at a later stage using the Solaris JumpStart mechanism.

4. **Install the package containing the Foundation Services man pages on the build server:**

   ```
   # pkgadd -d electronic_distribution_directory/NetraHAS2.1/Packages SUNWnhman
   ```

   The man pages are installed in the `/opt/SUNWcgha/man` directory.

5. **To access the man pages from the command line on your build server, add this directory to your `MANPATH` environment variable:**

   ```
   $ export MANPATH=${MANPATH}:/opt/SUNWcgha/man
   ```

The following procedure describes how to install the Solaris distribution on the build server:

▼ To Install the Solaris Distribution on the Build Server

To integrate the Solaris distribution in the software load to be deployed on the cluster, install the Solaris distribution on the build server.

1. **Log in to the build server as a superuser.**

2. **Ensure you have enough disk space on the build server for a Solaris distribution.**

   The Solaris distribution requires approximately 1.5 Gbytes.

3. **Create a directory for the Solaris distribution:**

   ```
   # mkdir Solaris-Distribution
   ```

   Where *Solaris-Distribution* is the directory where the distribution is installed on the build server.

4. **Move to the directory where the `setup_install_server` script is located.**

   Typically, this is in a Solaris tools directory. For example:
   Solaris-Dir/*Solaris_Version*/`Tools`

   Where:

   - *Solaris-Dir* is the directory containing the Solaris installation software. This could be a CD-ROM or an NFS-shared directory.

- *Solaris_Version* is `Solaris_8` or `Solaris_9`

5. **Run the `setup_install_server` script.**

   For example:

   ```
   # ./setup_install_server Solaris-Distribution
   ```

   For more information, refer to the `setup_install_server`(1M) man page, the *Solaris 8 Advanced Installation Guide*, or the *Solaris 9 Installation Guide*.

Installing Software on the Installation Server

This section describes how to install the Foundation Services, the Solaris operating system and the JumpStart Boot Environment software on the installation server.

▼ To Install the Foundation Services Software on the Installation Server

1. **Copy the `SUNWnhsmc` package to a temporary directory on the installation server.**

   This step is not necessary if the packages are available from a CD-ROM directory or accessible by means of an NFS mounted directory.

   a. **If the `SUNWnhsmc` package is in an electronic distribution, copy it to the local disk on the installation server.**

   b. **Ensure that you have access to any required patches or third party packages. For information, see the** *Netra High Availability Suite Foundation Services 2.1 6/03 Release Notes*.

2. **Install the Foundation Services package, `SUNWnhsmc`, on the installation server.**

   ```
   # pkgadd -d electronic_distribution_directory/NetraHAS2.1/Packages SUNWnhsmc
   ```

▼ To Install the Solaris Operating System Distribution on the Installation Server

If you plan to transfer the entire SMCT environment from the build server to the installation server during deployment, perform the following procedure. For more information, see "To Transfer the Complete SMCT Environment From the Build Server to the Installation Server" on page 95.

To integrate the Solaris distribution in the software load to be deployed on the cluster, install the Solaris distribution on the installation server.

1. **Log in to the installation server as a superuser.**

2. **Ensure that you have enough disk space on the installation server for a Solaris distribution.**

   The Solaris distribution requires approximately 1.5 Gbytes.

3. **Create a directory for the Solaris distribution:**

   ```
   # mkdir Solaris-Distribution
   ```

   Where *Solaris-Distribution* is the directory where the distribution is installed on the installation server.

4. **Change to the directory where the `setup_install_server` script is located.**

   Typically, this is in the Solaris tools directory. For example: *Solaris-Dir*/*Solaris_Version*/`Tools`

   Where:

   - *Solaris-Dir* is the directory containing the Solaris installation software. This could be a CD-ROM or an NFS-shared directory.
   - *Solaris_Version* is `Solaris_8` or `Solaris_9`

5. **Run the `setup_install_server` script:**

   ```
   # ./setup_install_server Solaris-Distribution
   ```

   For more information, refer to the `setup_install_server`(1M) man page, the *Solaris 8 Advanced Installation Guide*, or the *Solaris 9 Installation Guide*.

▼ To Install the Solaris JumpStart Boot Environment on the Installation Server

If you plan to transfer only exported data and flash archives from the build server to the installation server during deployment, perform the following procedure. For more information, see "To Transfer the Exported Data and Flash Archives From the Build Server to the Installation Server" on page 94.

- **On the installation server, perform the following command:**

  ```
  # cdrom-mnt-pt/Solaris_Version/Tools/setup_install_server \
  -b install_dir_path
  ```

  This option allows you to deploy the software load using the `fldeploy` command only. You cannot make any changes to software load data or run the `flinstall` command.

  For more information, refer to the `install_script`(4) man page.

CHAPTER 4

Planning Your Deployment

This chapter describes what you need to plan before starting the configuration stages and deploying the software load to a cluster. This chapter includes the following sections:

Defining Your Cluster

When choosing your hardware and software configuration, you must decide the following:

- The services you want to run on each node group
- Whether you want to deploy user applications or third party software on the cluster

Planning Your Cluster

You must define your cluster hardware configuration before you start to configure your deployment environment. If you use an example hardware configuration, you can use the template files for configuring the cluster that are provided for that hardware configuration. For information about the example hardware configurations, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

Planning Node Groups

Nodes belonging to a node group share the following properties:

- Node type. For example, master-eligible nodes, diskless nodes, or dataless nodes.
- Operating system. For example, Solaris 9 operating system.
- Architecture. For example, `SPARC`.
- Class. For example, `sun4u`.
- Foundation Services configuration. For example:
  - `NHAS_MASTER_ELIGIBLE`
    - `RBS`
    - `NSM`
    - `WDT_MASTER_ELIGIBLE`
  - `NHAS_DATALESS`
    - `WDT_DATALESS`
  - `NHAS_DISKLESS`
    - `MAC_ADDR_POLICY` or `STATIC_CLIENT_ID_POLICY`
    - `WDT_DISKLESS`
- Software. For example, the same user applications.
- Peripheral capabilities. For example, the same disk and I/O card capabilities.

The node groups are defined so that groups of hardware can be configured to run particular services and applications. For example, you can define a node group for two master-eligible nodes called `master_el`, that can be configured to run a selection of Foundation Services.

For information on how to configure node groups, see "Defining Nodes and Node Groups" on page 66.

Planning Services for Your Cluster

Each node group must be assigned core services that are defined by *core services lists*. The core service list that you assign to a node group depends on the role of the node group. The following table summarizes the core service lists.

TABLE 4–1 Predefined Core Service Lists

Node Group	Core Service List	Description
Master-eligible	`NHAS_MASTER_ELIGIBLE`	Full range of services, including: ■ CGTP ■ CMM ■ SNDR ■ Reliable NFS ■ Daemon Monitor ■ Node Management Agent
Diskless	`NHAS_DISKLESS`	■ CGTP ■ CMM ■ Daemon Monitor ■ Node Management Agent
Dataless	`NHAS_DATALESS`	■ CGTP ■ CMM ■ Daemon Monitor ■ Node Management Agent

When the core services have been defined, optional services can be added. The following table describes the optional service lists.

TABLE 4–2 Optional Service Lists

Node Group	Service List	Description
Master-eligible node group	`NSM`	Node State Manager
	`WDT_MASTER_ELIGIBLE`	Watchdog timer for master-eligible nodes
Master-eligible node group that contains diskless nodes	`RBS`	Reliable Boot Service. To enable a diskless node to boot, you must assign the `RBS` option.
Diskless node group	`WDT_DISKLESS`	Watchdog timer for diskless nodes
Dataless node group	`WDT_DATALESS`	Watchdog timer for dataless nodes

The services are assigned in the `cluster.conf` file. For information on how to configure the services, see "Choosing Services for Each Node Group" on page 68.

Carrier Grade Transport Protocol

One of the Foundation Services that SMCT automatically installs and configures on your cluster is the CGTP. The CGTP enables a redundant network for your cluster. If you do not require a redundant network for your cluster and you do not want to install andconfigure the CGTP, you must install manually or by using the `nhinstall` tool. In this case, if the single network link fails, there is no back up network and you might lose information.

For a complete description of the CGTP, see "Carrier Grade Transport Protocol" in the *Netra High Availability Suite Foundation Services 2.1 6/03 Overview*.

For information about installing manually or by using the `nhinstall` tool, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Custom Installation Guide*.

Watchdog Timer Service

Install and configure the Watchdog Timer provided with the Foundation Services only if you are using Netra servers that have hardware watchdogs at the Lights-Off Management (LOM) level. If you are using Netra servers with hardware watchdogs at the OpenBoot™ PROM (OBP) level, do not install the Watchdog Timer provided with the Foundation Services. These hardware watchdogs are monitored by the server's software. For a list of the types of watchdogs of different Netra servers, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*

Specifying a Diskless Node Boot Policy

During the configuration process, you are asked to specify a *boot policy* for each diskless node group. The following table summarizes the boot policies supported by the SMCT.

TABLE 4–3 Boot Policies for a Diskless Node Group

Boot Policy	Description
DHCP static boot policy	Based on the Ethernet address of the diskless nodes. Use the `MAC_ADDR_POLICY` parameter in the `cluster.conf` file.
DHCP client ID boot policy	Based on the client ID of a slot in a CompactPCI server. Use the `STATIC_CLIENT_ID_POLICY` parameter in the `cluster.conf` file.

TABLE 4–3 Boot Policies for a Diskless Node Group *(Continued)*

Boot Policy	Description
DHCP dynamic boot policy	Not supported by the SMCT.

The SMCT does not support dynamic address assignment. To assign dynamic addresses to diskless nodes, you can either install manually or use the `nhinstall` tool. These installation methods are described in the *Netra High Availability Suite Foundation Services 2.1 6/03 Custom Installation Guide*.

Planning Disk Layout and File Systems

This section describes the disk layout and file system configuration options for master-eligible nodes and dataless nodes. This section provides information about the disk space that you may require and the various configurations that are available.

Disk layout and the configuration of file systems is defined during configuration stage 1, and is described in "Defining the Disk Layout" on page 72.

The way you partition your disk depends on whether the disk is part of a master-eligible node or a dataless node. If your cluster is to contain diskless nodes, the disk partitions of the master-eligible nodes are different. Another condition to consider is volume managment software, which, if used, affects the way a disk is partitioned.

If you plan to logically partition and mirror disks, you require one of the following volume management software:

- Solstice DiskSuite™ 4.2.1 for Solaris 8 2/02
- Solaris Volume Manager for Solaris 9

The following table summarizes the possible disk layout configurations:

TABLE 4–4 Disk Layout Options for the Foundation Services

Disk Type	Diskless Node Configuration	Volume Management Software Configuration	Two Disk Support
Master-eligible	Optional	Optional	Optional
Dataless	No	No	Optional

The SMCT does not support virtual disk partitioning. To use virtual disk partitioning, you must install the Solaris operating system and manually set up the virtual partitioning before following the manual installation procedure. For more information, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Custom Installation Guide*.

For examples of disk layout and file system definition sections of the `machine.conf` file, see Appendix B.

Planning Disk Layout and File Systems for Master-Eligible Nodes

The master-eligible nodes in your cluster must have identical disk partitions. The disk partitions of the master-eligible node depend on the requirements of your cluster, and whether your cluster contains diskless or dataless nodes.

Disk Layout and File Systems for a Cluster With Diskless Nodes

The following table provides the minimum space requirements for an example disk partition of master-eligible nodes in a cluster with diskless nodes:

TABLE 4–5 Example Disk Layout and File Systems for a Cluster With Diskless Nodes

Disk Partition	Contents	Description	Minimum Size
0	`/`	The root file system, boot partition, and volume management software. This partition must be mounted with the `logging` option.	2 Gbytes minimum
1	`swap`	Minimum size when physical memory is less than 1 Gbyte.	1 Gbyte
2	`overlap`	Entire disk.	Size of the entire disk
3	`/export`	Exported file system reserved for diskless nodes. This partition must be mounted with the `logging` option. This slice is partitioned further if diskless nodes are added to the cluster.	1 Gbyte + 100 Mbytes per diskless node
4	`/SUNWcgha/local`	This slice is reserved for NFS status files, services, and configuration files. This partition must be mounted with the `logging` option.	2 Gbytes

TABLE 4–5 Example Disk Layout and File Systems for a Cluster With Diskless Nodes *(Continued)*

Disk Partition	Contents	Description	Minimum Size
5	Reserved for Reliable NFS internal use	Bitmap partition reserved for `nhcrfsd`. This partition is associated with the `/export` partition.	See Table 4–6
6	Reserved for Reliable NFS internal use	Bitmap partition reserved for `nhcrfsd`. This partition is associated with the `/SUNWcgha/local` partition.	See Table 4–6
7	`/mypartition`	For additional applications.	The remaining space

For replication, create bitmap partitions for each partition containing an exported replicated file system on the master-eligible nodes. The size of the bitmap partition must be equal to 512 bytes + 1 bit per 32 Kbytes of the data device size. In this example, the bitmap partitions are created on partitions 5 and 6. The following table gives example bitmap partitions.

TABLE 4–6 Example Bitmap Partitions

File System Name	Bitmap File	File System (Mbytes)	Bitmap Partition (Kbytes)	Bitmap size (block)
`/export`	`/dev/rdsk/c0t0d0s5`	2000	8512	18
`/SUNWcgha/local`	`/dev/rdsk/c0t0d0s6`	1512	6512	14

For information on creating bitmap partitions, see the *Sun StorEdge Availability Suite 3.1 Remote Mirror Software Installation Guide*.

Disk Layout and File Systems for a Cluster Without Diskless Nodes

For a cluster without diskless nodes, the following partitions are not required:

- The `/export` partition
- One partition reserved for Reliable NFS internal use

These partitions correspond to partitions 3 and 5 in Table 4–5.

Planning the Disk Layout and File Systems for Master-Eligible Nodes by Using Volume Management Software

This section describes the disk layout of the master-eligible nodes when using volume management software. You must use volume management software if you are using a Netra-20 as a master-eligible node.

Disk Layout and File Systems for a Cluster With Diskless Nodes

The following table provides the minimum space requirements for example disk partitions for a master-eligible node using volume management software, in a cluster with diskless nodes:

TABLE 4–7 Example Disk Layout and File Systems for a Cluster With Diskless Nodes

Disk Partition	Contents	Description	Minimum Size
0	`/`	The root file system, boot partition, and volume management software. This partition must be mounted with the `logging` option.	2 Gbytes minimum
1	`swap`	Minimum size when physical memory is less than 1 Gbyte.	1 Gbyte
2	`overlap`	Entire disk.	Size of the entire disk
3	`/SUNWcgha/local`	This slice is reserved for NFS status files, services, and configuration files. This partition must be mounted with the `logging` option. For a cluster without diskless nodes, this partition is free. This partition can also be mirrored.	2 Gbytes
4	`/export`	Exported file system reserved for diskless nodes. This partition must be mounted with the `logging` option. This slice is partitioned further if diskless nodes are added to the cluster.	1 Gbyte + 100 Mbytes per diskless node

TABLE 4–7 Example Disk Layout and File Systems for a Cluster With Diskless Nodes *(Continued)*

Disk Partition	Contents	Description	Minimum Size
5	`/mypartition`	Free for user. This partition can be mirrored.	The size of slice 5 (user-defined).
6	`/mypartition`	Free for user. This partition can be mirrored.	The size of slice 6 (user-defined).
7	Reserved	Reserved for SNDR bitmaps and volume management software meta database.	The sum of SNDR bitmaps + volume management software meta database files

Disk Layout and File Systems for a Cluster Without Diskless Nodes

For a cluster without diskless nodes, the following partition is not required:

- The `/export` partition

Planning the Disk Layout and File Systems for Dataless Nodes

The following table provides the minimum space requirements for example disk partitions of a dataless node.

TABLE 4–8 Example Disk Layout and File Systems for Dataless Nodes

Disk Partition	Contents	Description	Minimum Size
0	`/`	The root file system, boot partition, and volume management software. This partition must be mounted with the `logging` option.	2 Gbytes minimum
1	`swap`	Minimum size when physical memory is less than 1 Gbyte.	1 Gbyte
2	`overlap`	Entire disk.	Size of the entire disk
3,4,5,6,7	`/mypartition`	Free for user. This partition can be mirrored.	The remaining space

Disk Space for Additional Diskless Nodes and Additional Node Groups

This section describes the additional disk space you require when configuring a disk layout for additional diskless nodes or diskless node groups.

- Each additional node requires approximately 160 Mbytes
- Each additional node group requires 255 Mbytes

For example, to add a node group that includes two diskless nodes, you require the following disk space:

255 Mbytes + (2 * 160 Mbytes) = 575 Mbytes

Creating the SMCT Environment

An environment must be created for each user planning to deploy the Foundation Services by using the SMCT. Note that you must use the UNIX Korn shell when creating the SMCT environment.

To create the SMCT environment, perform the following procedure on the build server. Perform this procedure on the installation server if it is the same machine as the build server.

▼ To Create the SMCT Environment by Using the `nhsmctsetup` Command

1. **Log in to the build server as superuser in the Korn shell.**

2. **Run the script `nhsmctsetup`:**

   ```
   # ksh /opt/SUNWcgha/nhsmct/bin/nhsmctsetup
   ```

   You are prompted to define your SMCT environment by choosing a series of environment variables. After you have entered values for these variables, they are stored in the environment definition file, `smct.env`.

   In addition to the `smct.env` file, the `nhsmctsetup` command also creates a software repository and a software load repository.

   For more information, refer to the `nhsmctsetup`(1M) man page.

3. **Create an alias in your environment to the `sl`*xxx* and `fl`*xxx* commands.**

   Ensure that you are in the Korn shell and run the following command:

   ```
   # . $SMCT_ENV_DIR/scripts/smct.env
   ```

   This command must be run each time you return to your SMCT environment.

   For more information on SMCT environment variables, refer to the `nhsmctsetup`(1M) man page.

SMCT Directory Contents

The following directories have been created on the build server:

- `/opt/SUNWcgha/nhsmct/etc`
- `/opt/SUNWcgha/nhsmct/bin`
- `/opt/SUNWcgha/nhsmct/lib`

The following table describes the contents of the `/opt/SUNWcgha/nhsmct/etc/` directory.

TABLE 4–9 SMCT Directory Contents

Directory	Content
`jumpstart`	Solaris JumpStart file templates for the Solaris 8 and Solaris 9 operating systems.
`models`	Template files for the following configuration files: ■ `cluster.conf` ■ `machine.conf` ■ `network.conf` ■ `install-server.conf` ■ `master-system.conf`
`services`	Software configuration files for the Solaris 8 and Solaris 9 operating systems.
`hardware`	Hardware configuration descriptions for supported boards, disks, and servers.

Templates for Example Hardware Configurations

There are three configuration files used in the SMCT deployment process: `cluster.conf`, `machine.conf`, and `network.conf`.

All three configuration files must be modified to suit your deployment environment. A set of three templates for each example hardware configuration are provided with the software distribution of the Foundation Services.

For information about the example hardware configurations, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

For the list of templates, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Release Notes*.

▼ To Use the Template Files

1. **Copy the three configuration files for your example hardware configuration from the`/opt/SUNWcgha/nhsmct/etc/models` directory to your chosen destination directory,** ***destination_dir*****.**

   The default directory for the cluster configuration files is `SMCT_DEFAULT_CONFIG_DIR/models`, where `SMCT_DEFAULT_CONFIG_DIR` is the SMCT environment variable defined by `nhsmctsetup`.

   If you do not use the `-c` option when running the SMCT commands, the SMCT looks for the configuration files in the default directory. If you copy the templates to a *destination_dir* directory, you must specify this directory location with the `-c` option when running the SMCT commands.

   For more information, see the `nhsmctsetup`(1M) man page.

2. **Rename the files to remove the `*N.NETRA-*.tmpl` extension.**

   For example, if you are using the example hardware configuration for a six-node cluster:

   ```
   # cp /opt/SUNWcgha/nhsmct/etc/models/cluster.conf.6N.NETRA-CT.tmpl \
   destination_dir/cluster.conf
   # cp /opt/SUNWcgha/nhsmct/etc/models/machine.conf.6N.NETRA-CT.tmpl \
   destination_dir/machine.conf
   # cp /opt/SUNWcgha/nhsmct/etc/models/network.conf.6N.tmpl \
   destination_dir/config/model/network.conf
   ```

If your cluster is not one of the example hardware configurations listed in the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*, you must create the three cluster configuration files. For information, see the following chapters.

PART III

Configuring Your Deployment Environment

This part describes the three SMCT configuration stages. You must complete these stages before you can deploy the software to your cluster.

This part contains the following chapters:

- Chapter 5 describes stage 1 of the deployment process. This chapter describes how to configure the generic configuration files and how to create a generic flash archive for the cluster.
- Chapter 6 describes stage 2 of the deployment process. This chapter describes how to add user-defined configuration data to the generic flash archive. This stage is optional.
- Chapter 7 describes stage 3 of the deployment process. This chapter describes how to link the flash archives to specific cluster hardware and network definitions for deployment across the cluster.

CHAPTER **5**

Configuration Stage 1: Creating a Generic Flash Archive

This chapter describes stage 1 of the configuration process. It describes how to complete the cluster configuration files, software configuration files, and the configuration files for Solaris JumpStart.

This section describes how to create the cluster configuration files `cluster.conf`, `machine.conf`, and `network.conf`. If you are using an example hardware configuration, you can use the templates of the cluster configuration files provided with the product. For the example hardware configuration templates, see "Templates for Example Hardware Configurations" on page 53. Although template files are provided for the example hardware configurations, you may still need to make some modifications. When all the configuration files have been created, create a software load. In addition, create the Solaris JumpStart environment for the prototype machine. Then, you have to boot the prototype machine with the node group software.

When the node group software has been installed on the prototype machine, a generic flash archive is created on the build server.

When stage 1 has been completed, you have one or more generic flash archives for a generic cluster configuration.

This chapter covers the following topics:

- "Booting the Prototype Machine With the Node Group Software" on page 80
- "Creating a Generic Flash Archive" on page 82

Configuration Stage 1 Checklist

Before you perform configuration stage 1, ensure that you have:

- Set up your hardware as described in Chapter 3
- Planned your deployment environment and your disk layout, as described in Chapter 4
- Created the SMCT environment by using the `nhsmctsetup` command, as described in "Creating the SMCT Environment" on page 52

Configuration Stage 1 Procedure

The following figure shows the configuration procedure for stage 1.

Run `slcreate` for each software load

Run `flinstal` for each diskfull node group

A Solaris JumpStart environment is created for each diskfull node group

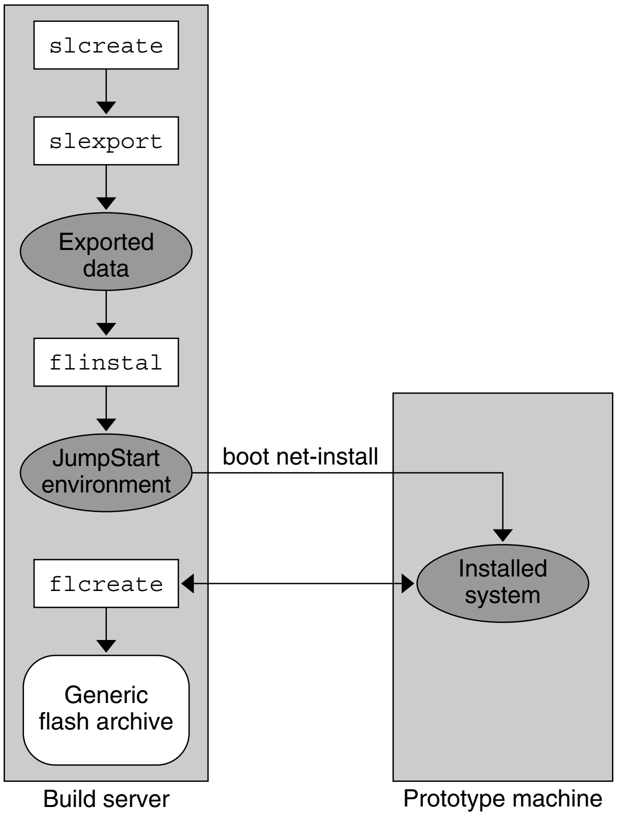

FIGURE 5–1 Configuration Stage 1

Connecting the Prototype Machine to the Build Server

Before you start configuration stage 1, make sure that you have edited the `install-server.conf` and `master-system.conf` files, as described in this section.

For information about connecting the prototype machine to the cluster, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

▼ To Connect the Prototype Machine to the Build Server

The files described in this section are used later in the procedure by the `flinstall` command. For more information on `flinstall`, see the `flinstall`(1M) man page.

1. **Log in to the build server as superuser.**

2. **Copy the `install-server.conf` file from the `/opt/SUNWcgha/nhsmct/etc/models` directory to your chosen destination directory, *smct-config-dir*`/models`, removing the `*.tmpl` extension.**

   For example:

   ```
   # cp /opt/SUNWcgha/nhsmct/etc/models/install-server.conf.tmpl /
   destination_dir/install-server.conf
   ```

3. **Edit the `install-server.conf` file.**

   For more information, see the `install-server.conf`(4) man page.

4. **Copy the `master-system.conf.tmpl` file from the `/opt/SUNWcgha/nhsmct/etc/models` directory to the *smct-config-dir*`/models` directory.**

   While copying the file, remove the `*.tmpl` extension. For example:

   ```
   # cp /opt/SUNWcgha/nhsmct/etc/models/master-system.conf.tmpl /
   destination_dir/master-system.conf
   ```

5. **Edit the `master-system.conf` file with the Ethernet address configuration.**

   For more information, see the `master-system.conf`(4) man page.

Configuring the Installation Server

This section describes how to configure the connectivity and NFS accessibility of the installation server.

▼ To Configure and Activate the Network Interface That Is Connected to the Cluster

Configure and activate the network interface that is connected to the target cluster.

1. **Ensure that the interface does not conflict with the cluster addressing that is assigned during the installation procedure.**

Check that the IP address is not already in use.

2. **If there is an addressing conflict, modify the addressing schema in the `master-system.conf` and `install-server.conf` files to resolve the conflict.**

▼ To Modify the `/etc/nsswitch.conf` File

1. **Log in to the installation server as a superuser.**
2. **Modify the `/etc/nsswitch.conf` file so that `files` is positioned before `nis` in the `hosts`, `ethers` and `bootparams` entries.**

   For example:

   ```
   hosts: files nis
   ethers: files nis
   bootparams: files nis
   ```

▼ To Mount the Installation Server Directories Onto the Prototype Machine and the Cluster

As file systems may already be shared, this section provides guidelines only.

The minimum requirements for sharing files by using the NFS are as follows:

1. **Set the permissions to read and write, `-o rw,anon=0`, for the directory where the flash archives are generated.**

   The default directory for the flash archives is `SMCT_FLASH_DIR`.
2. **Set the permissions to read only, `-o ro,anon=0`, for the directory containing the Solaris JumpStart environment.**

   The default directory for the Solaris JumpStart environment is `SMCT_JUMPSTART_DIR`.
3. **Set the permissions to read only, `-o ro,anon=0`, for the directory containing the Solaris distribution.**

   The default directory for the Solaris distribution is `SMCT_SOL_DIR`.
4. **Set the permissions to read only, `-o ro,anon=0`, for the directory containing the exported software load data.**

   The default directory for the exported software load data is `SMCT_EXPORT_DIR`.
5. **Set the permissions to read only, `-o ro,anon=0`, for the software repository.**

   The default directory for the exported software load data is `SMCT_EXPORT_DIR`. If you ran the `slexport` command with the `-r` option, the default directory is `SMCT_SOFTREP_DIR`.

6. **Set the permissions read-writer permissions, `-o rw,anon=0`, for the SMCT temporary directory.**

   The default temporary directory is `SMCT_TMP_DIR`.

▼ To Disable the Installation Server as a Router

- **Disable the installation server as a router by creating a `/etc/notrouter` file:**

  ```
  # touch /etc/notrouter
  ```

  If a system running the Solaris operating system has two network interfaces, it is configured as a router by default. However, a Foundation Services cluster network must not be routed for security reasons.

▼ To Reboot the Installation Server to Start the Solaris Daemons

- **Reboot the installation server to start or restart the Solaris daemons:**

  ```
  # init 6
  ```

  The following daemons are started or are restarted:

  - `rarpd`
  - `rpc.bootparamd`
  - `lockd`
  - `statd`
  - `mountd`
  - `nfsd`
  - `inetd`

Configuring the Prototype Machine

This section describes how to erase a previous DHCP boot configuration on the prototype machine.

▼ To Set the Boot Device on the Prototype Machine

1. **Get the `ok` prompt on the prototype machine.**

   For example:

```
telnet> send brk
ok>
```

2. **Configure the OBP parameters:**

```
ok> setenv boot-device disk net
```

SMCT Configuration File Locations

The following files are the SMCT configuration files:

- `cluster.conf`
- `machine.conf`
- `network.conf`
- `master-system.conf`
- `install-server.conf`

Each SMCT configuration file has a template file in the directory `/opt/SUNWcgha/nhsmct/etc/models`. Make sure that you have copied the template files to a directory of your choice as described in the *Netra High Availability Suite Foundation Services 2.1 6/03 Release Notes*.

The default directory for the cluster configuration files is the *smct-config-dir*`/models` directory. If the `-c` option is not used when running SMCT commands, the cluster configuration files are expected to be located in this directory.

If you copy the templates to a user-defined configuration directory, you must use this directory location with the `-c` option when running the SMCT commands.

When you run an SMCT command, the SMCT looks for the configuration files first in the directory specified with the `-c` option if used. If the `-c` option is not used, the SMCT looks for the necessary configuration files in the *smct-config-dir*. If there are no files in the *smct-config-dir*, the SMCT looks for the configuration files in the `/opt/SUNWcgha/nhsmct/etc/models/` directory.

For information about the configurable elements of these files, see Appendix A.

Configuring the `cluster.conf` Configuration File

The `cluster.conf` file requires the most information. The data in this file enables you to create the `machine.conf` and `network.conf` files.

To configure the `cluster.conf` file, you must do the following:

- Define the cluster composition
- Define the cluster domain
- Define the node groups
- Define the services for node groups
- Define the number of disks for the master-eligible nodes and dataless nodes
- Define the boot policy for diskless nodes

The following figure shows the components that are contained in the `cluster.conf` file:

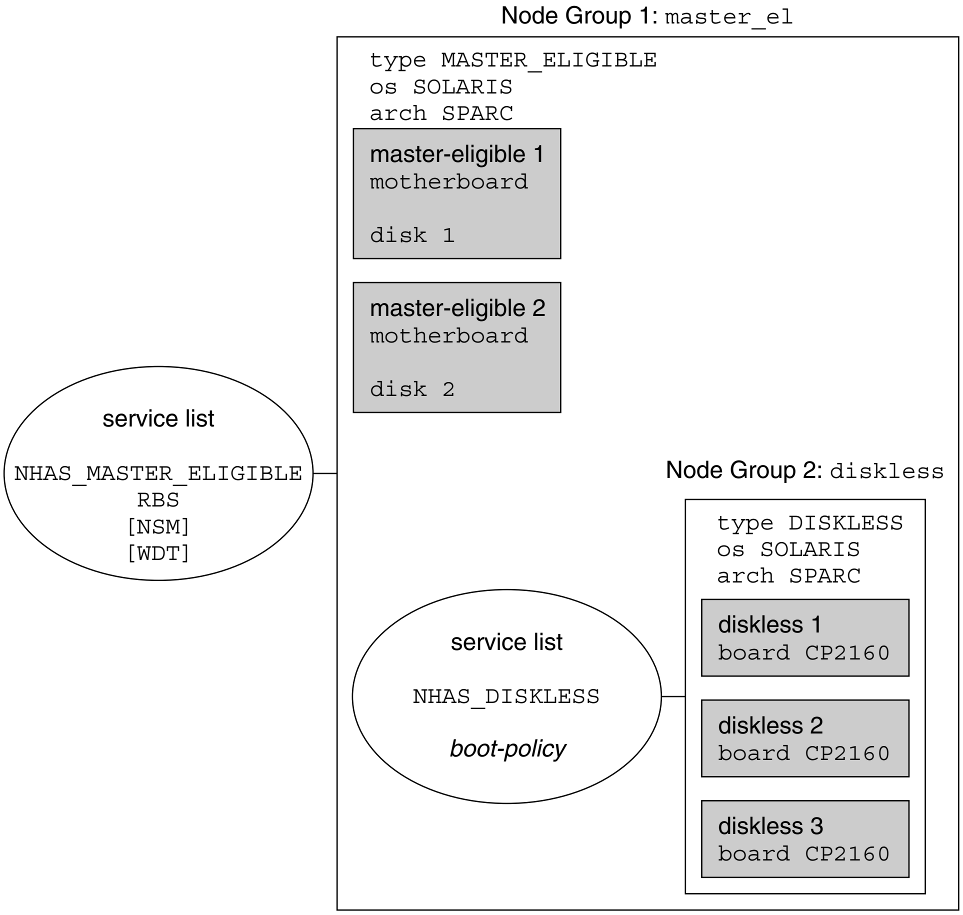

FIGURE 5–2 The `cluster.conf` Configuration File

Defining the Cluster Composition

The cluster composition section of the `cluster.conf` file contains the name and contents of the cluster.

The hardware contents and domain name of the cluster are described by the `INVOLVE` keyword.

The following program listing is a cluster composition section for a twelve node cluster spanned across two shelves:

```
# Cluster composition
#
ELEMENT cluster 12N_cluster
        INVOLVE shelf shelf_1
                shelf shelf_2
                domain cluster_domain
                nodeGroup master_el
                nodeGroup dataless_N120
                nodeGroup dataless_N200
```

All configuration elements of the cluster definition must be defined in the `cluster.conf` file, during stage 1. The exception is the `shelf` attributes that are defined in the `machine.conf` file.

Defining the Cluster Domain

The cluster domain definition section contains a logical definition of the cluster network parameters.

In the cluster domain definition section:

- The name and ID of the cluster domain are described by the `ELEMENT` keyword
- The cluster network interfaces are described by the `INVOLVE` keyword
- The floating addresses are described by the `USE` keyword

The following program listing is a cluster domain section for a twelve node cluster:

```
# Cluster domain definition
#
# id -> CMM domainId
# ip -> master master-nic0 master-nic1 floating addresses
#
ELEMENT domain cluster_domain id 100
        INVOLVE network phys-A
                network phys-B
                network cgtp
        USE ip master-cgtp
            ip master-nic0
            ip master-nic1
```

Defining Nodes and Node Groups

Every node in a cluster belongs to a node group. Node and node group attributes are defined in the `cluster.conf` file.

Nodes belonging to a node group share the following properties:

- Type
- Operating system
- Architecture
- Class
- Foundation Services configuration
- Software
- Peripheral capabilities

The following table lists the attributes that must be defined for each node group.

TABLE 5–1 Mandatory Attributes for Each Node Group Definition

Keyword	Description
`node`	Name of the node to be included in the node group.
`service`	Foundation Services list
`type`	This can be master-eligible, diskless, or dataless.
`os`	Operating system.
`arch`	System architecture.

For information on the syntax to use for each of the keywords, see the `cluster.conf`(4) man page.

For information on the concepts of node groups, see "Planning Node Groups" on page 44.

The nodes that are included in a node group are described by the `INCLUDE` keyword. The `INCLUDE` keyword is also used to embed a diskless node group in a master-eligible node group.

The services that are to be run with the node group are described by the `RUN` keyword.

Each cluster must have at least one node group consisting of the two master-eligible nodes.

The following program listing is an example node group and node definition for a master-eligible node group in a two-node cluster:

```
# MEN group and related nodes definitions
ELEMENT nodeGroup master_el type MASTER_ELIGIBLE os SOLARIS arch SPARC
        RUN service NHAS_MASTER_ELIGIBLE
#
```

```
# MEN definitions
ELEMENT node peerNode1-2N
        USE board N120@peerNode1
            disk disk1@peerNode1
#
ELEMENT node peerNode2-2N
        USE board N120@peerNode2
            disk disk1@peerNode2
```

The following program listing is a node group and node definition for a diskless node group for a four-node cluster:

```
# diskless group and related nodes definitions
ELEMENT nodeGroup diskless type DISKLESS os SOLARIS arch SPARC
        INCLUDE node peerNode3-4N
                node peerNode4-4N
        RUN service NHAS_DISKLESS
            service MAC_ADDR_POLICY

#
# diskless nodes definitions
ELEMENT node peerNode3-4N
        USE board CT@peerNode3

ELEMENT node peerNode4-4N
        USE board CT@peerNode4
```

The following program listing is a node group and node definition for a dataless node group for a twelve-node cluster:

```
# Node Groups definitions
#
# NMEN group and related nodes definitions
ELEMENT nodeGroup dataless_N120 type DATALESS os SOLARIS arch SPARC
        INCLUDE node peerNode3-12N
                node peerNode4-12N
                node peerNode5-12N
                node peerNode6-12N
        RUN service NHAS_DATALESS

# Nodes definitions
ELEMENT node peerNode3-12N
        USE board N120@peerNode3
            disk disk1@peerNode3
#
ELEMENT node peerNode4-12N
        USE board N120@peerNode4
            disk disk1@peerNode4
#
ELEMENT node peerNode5-12N
        USE board N120@peerNode5
            disk disk1@peerNode5
#
ELEMENT node peerNode6-12N
        USE board N120@peerNode6
        disk disk1@peerNode6
```

Choosing Services for Each Node Group

The services to be run on each node group are specified in the node group definition section of the `cluster.conf` file.

For a complete list of the core services, the optional services, and the diskless node boot policies, see "Planning Services for Your Cluster" on page 44.

▼ To Choose Services for Each Node Group

1. **Define the core services for each node group.**

   These services are mandatory. These services are defined by predefined core services lists.

   The core service list that you assign to each node group depends on the role of the node group. That is, `NHAS_MASTER_ELIGIBLE` for a master-eligible node group, `NHAS_DISKLESS` for a diskless node group, or `NHAS_DATALESS` for a dataless node group.

2. **Define any optional services to run on each node group.**

   This step is optional.

   The optional services are as follows:

   - `NSM` for the Node State Manager. This services can only be assigned to master-eligible node groups.
   - The watchdog timer service can be assigned to any type of node group. The watchdog timer services per node group are: `WDT_MASTER_ELIGIBLE`, `WDT_DISKLESS`, and `WDT_DATALESS`

3. **Specify the boot policy and Reliable Boot Service for each diskless node group.**

   - The boot policy must be defined for each diskless node group. The boot policy can be `STATIC_CLIENT_ID_POLICY` or `MAC_ADDR_POLICY`.
   - Ensure that the `RBS` has been assigned in the relevant master-eligible group service list. This service is required for master-eligible node groups containing diskless node groups. In addition, this service can only be assigned to master-eligible node groups with diskless node groups.

In the following example of a service definition, the core service list for the master-eligible node is `NHAS_MASTER_ELIGIBLE`, and the optional service is `NSM`:

```
RUN service NHAS_MASTER_ELIGIBLE
service NSM
```

For information on the full range of services and service lists available for the node groups, see the `cluster.conf`(4) man page.

Defining the Boot Policy for a Diskless Node Group

You must specify one of the following boot policies for each diskless node group:

- `MAC_ADDR_POLICY`

  Static address assignment, based on the diskless node Ethernet address of the diskless nodes. For information on how to configure the diskless nodes for the `MAC_ADDR_POLICY`, see "To Configure DHCP Static Boot Policy for a Diskless Node" on page 114.
- `STATIC_CLIENT_ID_POLICY`

  Address assignment for CompactPCI boards. For information on how to configure the diskless nodes for the `STATIC_CLIENT_ID_POLICY`, see "To Configure DHCP Client ID Boot Policy for a Diskless Node" on page 114.

If your diskless nodes are CompactPCI boards you can choose between `MAC_ADDR_POLICY` or `STATIC_CLIENT_ID_POLICY`.

If your diskless nodes are other types of nodes, you must choose `MAC_ADDR_POLICY`.

In the node group definition section of the `cluster.conf` file, use the following syntax to specify the boot policy:

```
RUN service core_service_ list
    service boot_policy
```

For example:

```
RUN service NHAS_DISKLESS
    service MAC_ADDR_POLICY
```

For more information, see the `cluster.conf`(4) man page.

Configuring the `machine.conf` Configuration File

The `machine.conf` configuration file describes the target cluster in terms of hardware components, disk layout for master-eligible nodes and dataless nodes, and file system configuration. The network interface parameters can be defined in configuration stage 3.

You must configure `machine.conf` to define the following parameters:

- Define the cluster hardware

- Define the disk layout
- Define the file system definitions
- Optionally, you can configure the network interfaces

Note – If you configure the network interfaces in stage 1, you must also complete the `network.conf` file in stage 1. Alternatively, you can configure the network interfaces in the `machine.conf` and `network.conf` files in stage 3. For more details, see "Adding Cluster-Specific Configuration Data to the `machine.conf` File" on page 96 and "Adding Cluster-Specific Configuration Data to the `network.conf` File" on page 97.

In addition, the disk controllers can be changed during stage 3 by modifying the configuration element `disk.device`.

The following figure shows the components contained in the `machine.conf` configuration file.

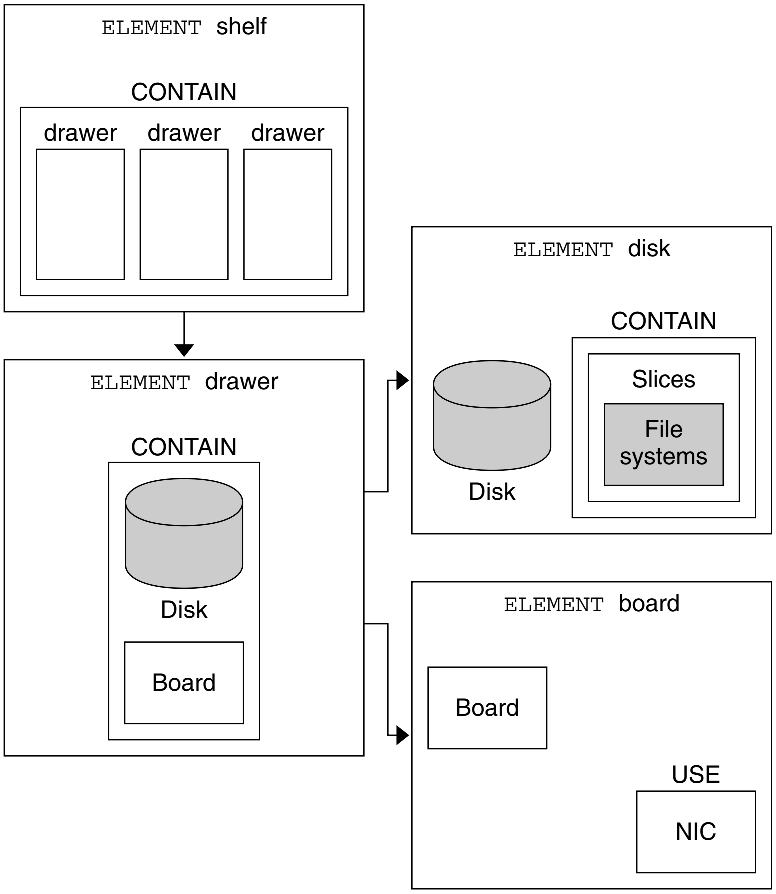

FIGURE 5–3 Components Contained in the `machine.conf` Configuration File

Defining the Cluster Hardware

The cluster hardware definition is a hierarchical description of the cluster hardware. The definition begins with a top level description of the contents of the shelf. The shelf typically contains drawers and switches. The `ELEMENT` keyword identifies the element that is being defined, the `CONTAIN` and `USE` keywords describe the content of the element. For example:

```
ELEMENT shelf shelf_1
        CONTAIN drawer NETRA-20-1
                drawer NETRA-20-2
                switch SWITCH-1
ELEMENT shelf shelf_2
        CONTAIN drawer N120-1
```

```
                drawer N120-2
                drawer N120-3
                drawer N120-4
                switch SWITCH-2
```

Each drawer is then described in terms of disks and boards. These disks and boards are referred to by the node defined in the `cluster.conf` file. The `ELEMENT` keyword is used to describe each node in terms of disks, boards, and interfaces. The content of each `ELEMENT` is described by the `CONTAIN` keyword. For example, the `ELEMENT disk` is described in terms of `slices`. The one exception to this is the `ELEMENT board` that is described in terms of `nic` interfaces by the `USE` keyword.

The following program listing is an example of a drawer that contains a master-eligible node:

```
ELEMENT drawer NETRA-20-1 type NETRA_20
        CONTAIN board mother@peerNode1
                disk disk1@peerNode1
# MEN peerNode1 disk
ELEMENT disk disk1@peerNode1 device c0t0d0 type SCSI size 36
        CONTAIN slice s0@disk1
                slice s1@disk1
                slice s3@disk1
                slice d3@disk1
...
```

Network configuration parameters are added to the `machine.conf` file in stage 3. For more information, see "Adding Cluster-Specific Configuration Data to the `machine.conf` File" on page 96.

Defining the Disk Layout

This section describes the disk layout of master-eligible nodes and dataless nodes. You must define your disk partitions as described in "Planning Disk Layout and File Systems" on page 47. Each partition on the disk is described by the `ELEMENT` keyword in terms of the following parameters:

- Size of partition
- File system to be mapped to the partition
- Whether the partition is replicated or not

Note – If you are using the volume management software, see "Planning Disk Layout and File Systems for Master-Eligible Nodes" on page 48. You must use volume management software if you are using a Netra-20 as a master-eligible node.

Each time you define a physical partition by using volume management software, define a logical partition to be managed by the volume management software.

For examples of disk layout definition sections of the `machine.conf` file, see Appendix B.

For information on the syntax used in the `machine.conf` file, see the `machine.conf`(4) man page.

Defining File Systems

The file system definitions specify the file systems on each disk. The file system definitions are associated to the partition definitions through the `MAP` configuration block.

Each file system is described by the `ELEMENT` keyword in terms of root directories, read-write permissions, and whether the file system is exported or shared.

For information on defining file systems, see "Planning Disk Layout and File Systems" on page 47.

You must define the following roles in the file system definition section by using the `role` attribute:

- `root`
- `swap`
- `shared`
- `data`
- `services`
- `database`
- `export`

  Only define the `export` role if diskless nodes are defined in the node group.
- `user`

  This role is assigned to user-defined file systems.

Only the following parameters can be modified by the user:

- In the `root`, `swap`, `shared`, and `export` file systems:

  `size` but with values greater than or equal to those in the template.
- In the partition associated to the `root`, `swap`, `shared`, and `export` file systems:

  `size` but with values greater than or equal to those in the template.
- In the `user` file system (non-replicated):

  `fsck`, `size`, `mntPt`, `mntOpt`, `mntBt`.
- In the `user` file system (replicated):

  `fsck`, `size`, `mntPt`.
- In the slice associated to the `user` file system:

`size` but with values greater or equal to those in the template.

For information on the syntax used in the `machine.conf` file, see the `machine.conf`(4) man page.

For examples of file system definition sections of the `machine.conf` file, see Appendix B.

Configuring the `network.conf` File

You must configure the `network.conf` file in stage 1 if you have:

- Configured the network interfaces in the `machine.conf` file in stage 1
- Configured a `domain` that defines IP, router or network information in the `cluster.conf` file

If you did not configure network interfaces in the `machine.conf` or `cluster.conf` files, you can configure the `network.conf` file in stage 3. For more information, see "Adding Cluster-Specific Configuration Data to the `network.conf` File" on page 97.

For more information, see the `network.conf`(4) man page.

Creating a Software Configuration File for a Node Group

A software configuration file is an optional file that defines software for installation on a specific node group in addition to the Foundation Services and Solaris operating system.

▼ To Create a Software Configuration File

- **Create a software configuration file for each node group that requires additional software packages or software patches.**

  For more information on software configuration files, including examples, see the `software.conf`(4) man page.

A software configuration file is used as a parameter of the `slcreate` command. For more information, see the `slcreate`(1M) man page.

Customizing Solaris JumpStart Files for the Prototype Machine

Solaris JumpStart files are required by the master-eligible node groups and dataless node groups. At this stage, you have the option to customize the Solaris JumpStart files for the prototype machine.

If you do not want to customize the Solaris JumpStart files for the prototype machine, skip this section. To customize the Solaris JumpStart files, you can do the following:

- Add additional Solaris packages from the Solaris distribution installed on the installation server
- Adapt to the target cluster environment, for example choice of language, timezone
- Change the default superuser password

Templates for Solaris JumpStart files for the prototype machine are provided in `/opt/SUNWcgha/nhsmct/etc/jumpstart`, and contain the `.proto` extension:

- `profile.proto`
- `begin.proto`
- `sysidcfg.proto`

Do not modify the `rules.proto` or `finish.proto` files.

For information about changing parameters in Solaris JumpStart files, see the Solaris JumpStart documentation.

▼ To Configure the Solaris JumpStart `profile` File for the Prototype Machine

Each `profile` file defines the set of Solaris operating system packages to be installed on each master-eligible node and dataless node.

A template for the `profile` file, named `profile.proto`, is provided in `/opt/SUNWcgha/nhsmct/etc/jumpstart/`*Solaris_version*.

1. **Log in to the build server.**
2. **Copy the `profile.proto` template from `/opt/SUNWcgha/nhsmct/etc/jumpstart/`*Solaris_version* to one of the following directories:**
   - *smct-config-dir*`/jumpstart/`*Solaris_version*
   - *user-config-dir*`/jumpstart/`*Solaris_version*

Each `profile` file must be named in the following format:

`profile.`*target*[`.`*extension*]

Where:

- *target* is `.proto` for the prototype machine
- *extension* is the name of the master-eligible or dataless node group as defined in the `cluster.conf` file

For example, a Solaris JumpStart `profile` file for the node group `soft_switch` is `profile.proto.soft_switch`.

3. **Modify the following parameters:**
   - `geo`

     This must be set to your geographical location. For example, `W_Europe`.
   - `locale`

     This must be set to your required language. For example, `en_US`.

4. **(Optional) Modify the following parameters:**
   - `cluster`

     This must include the default value `SUNWcreq`
   - `package`

     The list of Solaris operating system packages to be installed on each master-eligible node or dataless node. You can add to the list of packages but do not delete any packages already defined in the template.

5. **Do not configure the following parameters:**
   - `backup_media`
   - `client_arch`
   - `client_root`
   - `client_swap`
   - `fdisk`
   - `isa_bits`
   - `boot_device`

     This is managed by the SMCT
   - `root_device`

     This is managed by the SMCT
   - `use_disk`

     This is managed by the SMCT
   - `filesys`

     This is managed by the SMCT
   - `dontuse`

- `layout_constraint`
- `num_clients`

6. **Do not modify the following parameters:**
   - `install_type`
   - `system_type`
   - `partitioning`

▼ To Configure the Solaris JumpStart `sysidcfg.proto` File for the Prototype Machine

The system identification configuration file contains information for the target cluster such as name service, timezone and superuser password. Note that the Foundation Services default superuser password is `sunrules`.

1. **Log in to the build server.**
2. **Copy the `sysidcfg.proto` template from `/opt/SUNWcgha/nhsmct/etc/jumpstart/` to one of the following directories:**
   - *smct-config-dir*`/jumpstart/`
   - *user-config-dir*`/jumpstart/`
3. **Modify the file `sysidcfg.proto` to suit your local environment.**

   You can configure the following parameters:
   - `system_locale`
   - `timezone`
   - `terminal`
   - `root_password`
   - `monitor`
   - `keyboard`
   - `display`
   - `pointer`

Note – Although the template files for `rules.proto` and `finish.proto` are provided, do not configure these files. The template for the `begin.proto` file is supported by the SMCT. If required, modify this file. For more information, see the Solaris JumpStart documentation.

Creating the Software Load

This section describes how to create the software load by using the `slcreate` command. The `slcreate` command does the following:

- Compiles the cluster configuration files
- Copies the Foundation Services packages and any user applications into the software repository
- Generates the Solaris JumpStart configuration files for master-eligible node groups and dataless node groups
- Creates a software load in the software load repository

▼ To Create the Software Load by Using the `slcreate` Command

1. **Log in to the build server.**

2. **Run the command `slcreate`:**

```
$ slcreate -n SWL1 -v ha_v1
```

The `slcreate` command is run once for each software load because all node groups are processed in one operation. If `slcreate` stops due to an error, you can run the command again. The `slcreate` command automatically restarts from the stage when the error occurred.

If a package in a software load already exists in the software repository, it is not overwritten. Several versions of the same package can be stored in the software repository.

After the `slcreate` command has been run, you must restart the entire configuration process and deployment process to add or remove user applications.

Note – To add user applications to your software load, declare them in the software configuration files before you run `slcreate`. Software configuration files are described in "Creating a Software Configuration File for a Node Group" on page 74.

The Solaris operating system can be modified according to the rules described in "To Configure the Solaris JumpStart `profile` File for the Prototype Machine" on page 75.

For more information, see the `slcreate`(1M) man page.

Copying the Software Load to an Export Directory

When the `slcreate` command has been run successfully, use the `slexport` command to copy the software load into an export directory.

The export directory is used by the `flinstall` and `flcreate` commands to create the generic flash archives.

▼ To Copy the Software Load to the Export Directory by Using the `slexport` Command

1. **Log in to the build server.**
2. **Run the `slexport` command:**

   ```
   $ slexport -n SWL1 -v ha_v1
   ```

   This command exports data from the software load `SWL1`, to the directory `SMCT_EXPORT_DIR/SWL1@ha_v1`.

   The data is exported to the directory or in the directory specified by the `-e` option.

   If your installation server is a different machine than the build server, use the remote option, `-r`, of the `slexport` command. Then, copy the exported data to the installation server.

For more information, see the `slexport`(1M) man page.

Creating the Solaris JumpStart Environment for the Prototype Machine

When the software load has been copied to the export directory, create the Solaris JumpStart environment for the prototype machine by using the `flinstall` command.

The `flinstall` command generates a Solaris JumpStart environment for each master-eligible node group and dataless node group to be installed on the prototype machine. The command uses data from the export directory.

Note – Make sure that you have performed the necessary steps to connect the prototype machine to the build server. For more information, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

▼ To Create the Solaris JumpStart Environment for the Prototype Machine by Using the `flinstall` Command

1. **Log in to the build server as superuser.**
2. **Run the command `flinstall`:**

   ```
   # flinstall -n SWL1 -v ha_v1 -g master_el
   ```

   The above example creates the Solaris JumpStart environment configuration files in the directory `SMCT_JUMPSTART_DIR/jumpstart`:

   Run the `flinstall` for each master-eligible node group and dataless node group because each node group is processed individually.

For more information on using the `flinstall` command, see the `flinstall`(1M) man page.

When the `flinstall` command has been run, the Solaris JumpStart environment is created to enable the Solaris JumpStart mechanism on the prototype machine.

Verifying That the Installation Server Directories Are Shared

Before you boot the prototype machine, verify that the installation server directories are mounted on the prototype machine and the cluster. For more information, see "To Mount the Installation Server Directories Onto the Prototype Machine and the Cluster" on page 61.

Booting the Prototype Machine With the Node Group Software

This section describes how to install the node group software on the prototype machine by using the Solaris JumpStart environment and a boot net installation.

The node group software is transferred to the prototype machine from the build server by using Solaris JumpStart. The Solaris operating system, the Foundation Services packages and any user applications are installed on the prototype machine.

▼ To Boot the Prototype Machine With the Node Group Software

1. **Log in to the prototype machine.**
2. **At the `ok` prompt, boot the prototype machine:**

   ```
   ok> boot net - install
   ```

   The prototype machine performs a Solaris JumpStart from the build server. The prototype machine then installs the node group software, but does not start the Foundation Services.

When the prototype machine has been installed with the node group software, a flash archive of the prototype system can be created.

Verifying That the Node Group Software Has Successfully Installed

To verify that the node group software has been successfully installed on the prototype machine, perform the following procedure:

▼ To Verify That the Node Group Software Has Been Successfully Installed on the Prototype Machine

1. **Log in to the prototype machine.**
2. **Examine the Solaris JumpStart files `/var/sadm/system/logs/finish.log` and `/var/sadm/system/logs/install_log`.**

   Ensure that these logs contain confirmation that the prototype machine was successfully installed with the node group software.
3. **Run the `nhadm` tool to check the installation.**

   ```
   # nhadm check installation
   ```

Creating a Generic Flash Archive

When the node group software has been installed on the prototype machine, use the `flcreate` command to create the flash archive. The flash archive is created on the build server using the system image from the prototype machine.

The `flcreate` command creates a generic flash archive. The command is started on the build server but runs remotely on the prototype machine by way of a mounted directory.

▼ To Build a Generic Flash Archive by Using the `flcreate` Command

1. **Log in to the build server as superuser.**
2. **Run the command `flcreate`:**

   ```
   # flcreate -n SWL1 -v ha_v1 -g master_el -f FLASH_1
   ```

   This example creates the generic flash archive in the directory *smct-config-dir*`/flash`. This is the default directory defined by the `SMCT_FLASH_DIR` environment variable.

   Run the `flcreate` command for each master-eligible node group and dataless node group because each node group is processed individually.

   Note – If an error occurs when `flcreate` is running, manually unmount the `/mnt/flash` directory from the prototype machine and run the `flcreate` command again.

For more information on using the `flcreate` command, see the `flcreate`(1M) man page.

CHAPTER 6

Configuration Stage 2: Adding User-Defined Data to the Flash Archive

This chapter describes how to add user-defined data and user-specific installation scripts to an existing flash archive. The flash archive can be:

- A generic flash archive created in stage 1, as described in Chapter 5.
- A deployable flash archive that has already been through stage 3, as described in Chapter 7.

Stage 2 is optional. If you do not plan to add user applications and user-defined configuration data, you do not need to read this chapter.

The `slconfig` command prepares the software load configuration data, and the `flconfig` command incorporates the user-defined configuration data into the flash archives.

When stage 2 is complete, you have one or more flash archives that include user applications and user-defined configuration data. These archives are called *configured flash archives*.

This chapter contains the following sections:

Configuration Stage 2 Checklist

Before you perform configuration stage 2, ensure you have:

- Created a generic flash archive as described in Chapter 5.
- Declared your user applications in the software configuration file(s) described in "Creating a Software Configuration File for a Node Group" on page 74.

Configuration Stage 2 Procedure

The following figure shows the configuration procedure for stage 2, resulting in a configured flash archive.

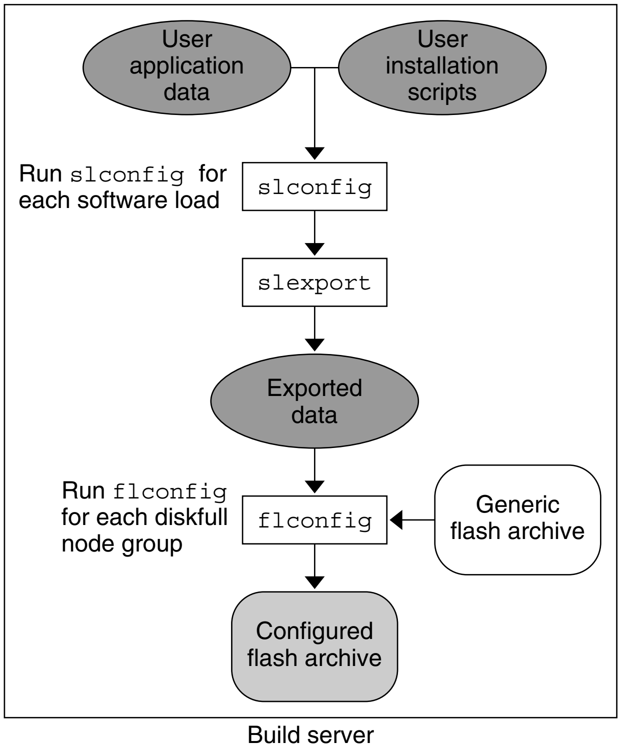

FIGURE 6–1 Configuration Stage 2

Creating a User Application Configuration File for a Node Group

The user application configuration file contains a structured list of user-defined configuration data to be run at specific stages of cluster startup. User-defined configuration data can be:

- User application data files that contain data to be added to the user application

  For example, if the user application is a database package, the user application data file could be data to populate the database.
- Application installation scripts that add the user application data to the user applications

  For example, you can specify a script or scripts to populate a database with the application data to create new records and new entries.

A user application configuration file is required for each node group that is to be installed with user applications. If no applications are to be deployed, it is not necessary to have any user application configuration files.

The application installation scripts can be executed at any one of three stages during the boot phase of the target cluster:

- At Solaris JumpStart installation, during the execution of the `finish` script
- At run level 2, when the Foundation Services packages are installed but they are not started
- At run level 3, when the Foundation Services are started and are running

The following figure illustrates the stages at which the installation scripts run during the final system boot sequence.

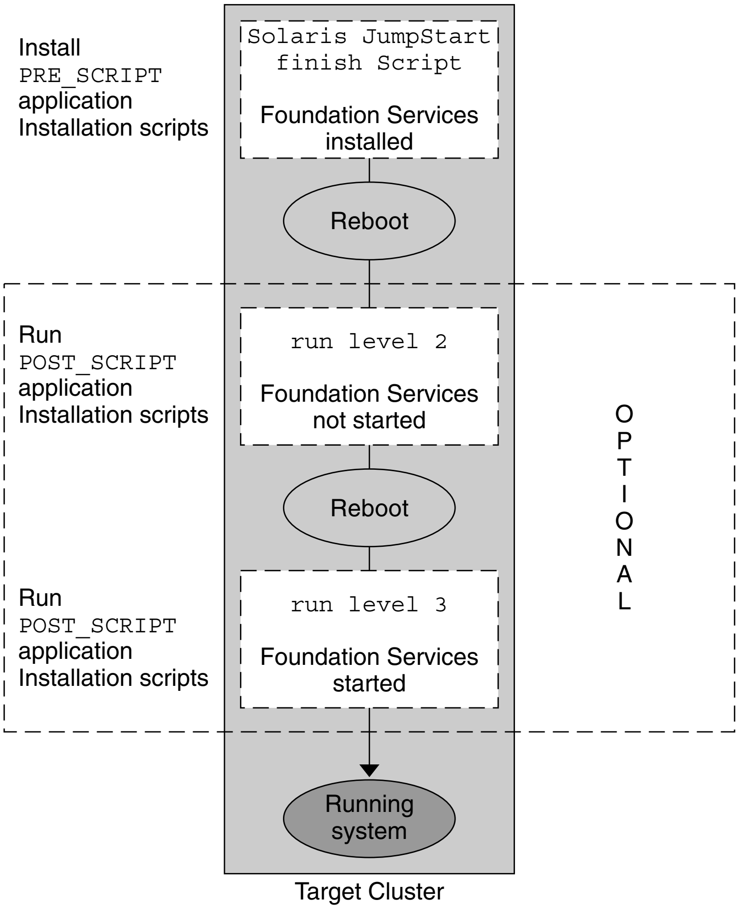

FIGURE 6–2 Boot Sequence for User Application Configuration Files

---

Note – When creating the application configuration file, note that the `PRE_SCRIPT` scripts do not run under the root directory (/) during the Solaris Jumpstart finish script. For more information, see the *Solaris Custom JumpStart Guide.*

The following example is a user application configuration file `config.soft_switch`, for a node group `soft_switch`.

EXAMPLE 6–1 Example of a User Application Configuration File

```
# Configuration of node group softSwitch - first stage -
#
# Script softSwitch_1.sh will be executed during Jumpstart finish stage
#ELEMENT config softSwitch_1
```

```
        INVOLVE file data_1_1
                file data_1_2
                file script_1_3

ELEMENT file data_1_1
        type DATA
        location 'file://config/softSwitch_1_1.conf'

ELEMENT file data_1_2
        type DATA
        location 'file://config/softSwitch_1_2.conf'

ELEMENT file script_1_3
        type PRE_SCRIPT
        location 'file://config/softSwitch_1.sh'

#
# Configuration of node group softSwitch - second stage -
#
# Script softSwitch_2.sh will be executed at init run-level 2
# The node will automatically reboot
#
ELEMENT config softSwitch_2
        INVOLVE file data_2_1
                file script_2_2

ELEMENT file data_2_1
        type DATA
        location 'file://config/softSwitch_2_1.conf'

ELEMENT file script_2_2
        type POST_SCRIPT
        runLevel 2
        location 'file://config/softSwitch_2.sh'

#
# Configuration of node group softSwitch - third stage -
#
# Scripts will be executed at init run-level 3 in the following order:
#    1. softSwitch_3.sh
#    2. softSwitch_4.sh
#
ELEMENT config softSwitch_3 next  softSwitch_4
        INVOLVE file script_3_1

ELEMENT file script_3_1
        type POST_SCRIPT
        location 'file://config/softSwitch_3.sh'

ELEMENT config softSwitch_4
        INVOLVE file script_4_1

ELEMENT file script_4_1
```

EXAMPLE 6–1 Example of a User Application Configuration File *(Continued)*

```
        type POST_SCRIPT
        location 'file://config/softSwitch_4.sh'
```

For more information on creating user application configuration files, see the `userapp.conf`(4) man page.

Preparing User-Defined Configuration Data

This section describes how to prepare user-defined configuration data by using the `slconfig` command.

The `slconfig` command compiles and adds user-defined configuration data to the software load repository. The user-defined configuration data, including application installation scripts are added as parameters for each node group.

▼ To Prepare the User-Defined Configuration Data by Using the `slconfig` Command

1. **Log in to the build server as superuser.**
2. **Run the command `slconfig`:**

```
$ slconfig -n SWL1 -v ha_v1
```

The `slconfig` command is run once because all node groups are processed in one operation.

For more information on the `slconfig` command, see the `slconfig`(1M) man page.

Copying User-Defined Configuration Data to an Export Directory

Use the `slexport` command to copy the user-defined configuration data created by `slconfig` into an export directory. The export directory is used by the `flconfig` command to add the user-defined configuration data to the flash archive.

▼ To Copy the User-Defined Configuration Data to the Export Directory by Using the `slexport` Command

1. **Log in to the build server.**
2. **Run the command `slexport`:**

   ```
   $ slexport -n SWL1 -v ha_v1
   ```

   This command exports user-defined configuration data in `SWL1`, to the directory `SMCT_EXPORT_DIR/SWL1@ha_v1`.

The `slexport` command copies all software load data to the export directory, including data previously generated by the `slcreate` command.

For more information on using the `slexport` command, see the `slexport`(1M) man page.

Adding User-Defined Configuration Data to the Flash Archive

When the user-defined configuration data has been copied to the export directory, it must be added to the flash archive by using the `flconfig` command.

▼ To Add User-Defined Configuration Data to the Flash Archive by Using the `flconfig` Command

1. **Log in to the build server as a superuser.**
2. **Run the `flconfig` command:**

   ```
   # flconfig -n SWL1 -v ha_v1 -g master_el -f FLASH_1
   ```

   The `flconfig` command uses the user-defined configuration data from the export directory to generate *configuration sections*. A configuration section contains all the user-defined configuration data for each node group. The configuration sections are added to the flash archive.

   The `flconfig` command is run for each flash archive on which you want to replace the user-defined configuration data.

   The `flconfig` command must be run for each flash archive associated to a master-eligible node group or dataless node group because each node group is processed individually.

The diskless node groups are configured when the master-eligible node group or dataless node group that embeds them is being processed.

For information about the `flconfig` command, see the `flconfig`(1M) man page.

CHAPTER 7

Configuration Stage 3: Configuring the Flash Archive for Deployment

This chapter describes how to configure your flash archive for a specific cluster hardware and network configuration. The flash archive can be one of the following:

- A generic flash archive created in stage 1, as described in Chapter 5.
- A configured flash archive created in stage 2, as described in Chapter 6.

Configuration stage 3 generates the Foundation Services configuration data and a Solaris JumpStart environment for each master-eligible node group and dataless node group.

This configuration stage can be performed on a flash archive that has already been deployed but which requires an updated networking and cluster hardware configuration. Stage 2 can be rerun before performing stage 3.

When stage 3 is complete, you have one or more deployable flash archives with a cluster-specific configuration, and, optionally, user-defined configuration data and application installation scripts.

This chapter contains the following sections:

- "Creating a Deployable Flash Archive and the Solaris JumpStart Environment" on page 106
- "Modifying User-Defined Configuration Data" on page 107

Configuration Stage 3 Checklist

Before you perform configuration stage 3, ensure that:

- You have created a generic flash archive or configured flash archive as described in Chapter 5 and Chapter 6.
- You have connected the installation server to the master-eligible nodes and dataless nodes on the cluster.
- The network parameters in the `install-server.conf` file are consistent with the network parameters in the `network.conf` file.

Configuration Stage 3 Procedure

The following figure illustrates the configuration procedure for stage 3, resulting in a running cluster.

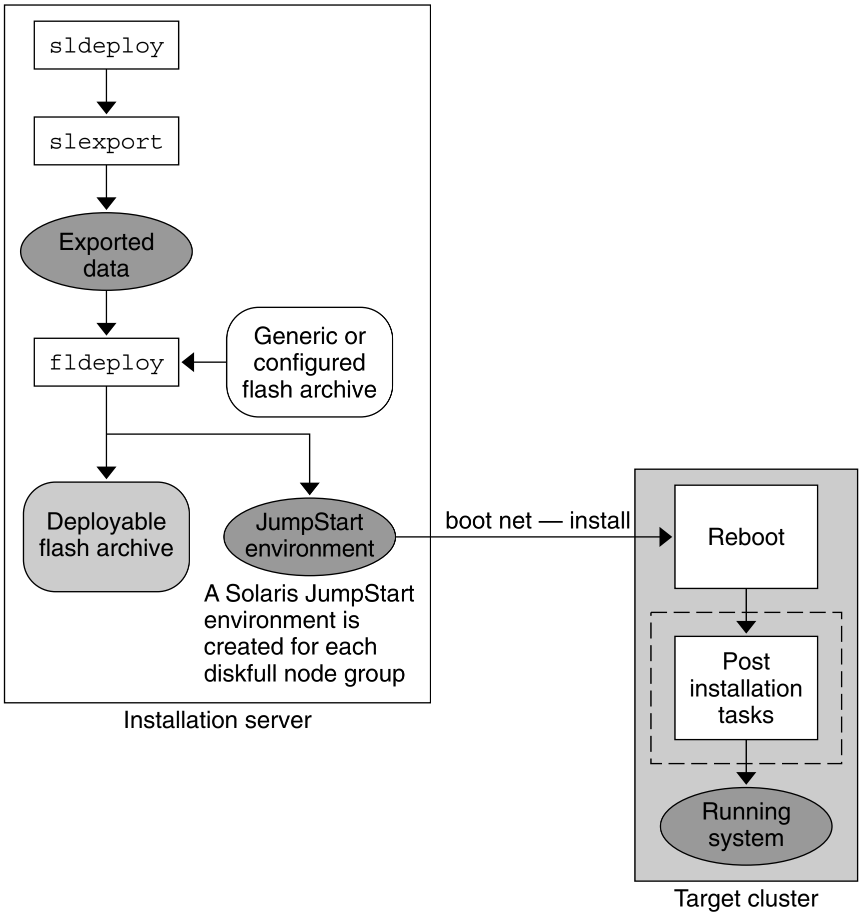

FIGURE 7–1 Configuration Stage 3

The post-installation tasks are described in detail in Figure 6–2.

Transferring Your SMCT Environment From the Build Server to the Installation Server

This section describes how to transfer flash archives and software load data from the build server to an installation server. The build server can have either a local connection to the installation server or the installation server can be at a remote site. If you are using your build server as an installation server, you do not need to read this section.

Choose one of the following ways to transfer your SMCT environment:

- Transfer the exported data and flash archives only

  This option allows you to deploy the software load using `fldeploy` only. You cannot make any changes to software load data.
- Transfer the complete SMCT directory

  This option allows you to modify the existing software load for deployment to another cluster.

▼ To Transfer the Exported Data and Flash Archives From the Build Server to the Installation Server

---

Note – It is not necessary to copy the software load data or the cluster configuration files to the installation server. The `sldeploy` command has already been executed on the build server.

If you follow this procedure, you cannot modify software load data or run the `sl`*xxx* commands. To modify software load data, follow the procedure "To Transfer the Complete SMCT Environment From the Build Server to the Installation Server" on page 95.

1. **Log in to the build server.**
2. **Export the software load data:**

```
# slexport -r -n swl-name -v swl-version -e export-dir
```

3. **Ensure that the** *export-dir* **is NFS mounted and readable from the installation server.**

4. **Copy the exported data from the build server to the installation server:**

   ```
   # cp -r export-dir install-server-dir
   ```

5. **On the installation server, copy the flash archives from the build server:**

   ```
   # cp mounted_dir/flash_archive1 SMCT_FLASH_DIR
   ```

6. **On the installation server, run the `nhsmctsetup` command:**

   ```
   # /opt/SUNWcgha/nhsmct/bin/nhsmctsetup -w working-dir -s solaris-dist
   ```

   Where *working-dir* specifies the root directory of the SMCT environment that is to be created.

   This command enables you to run the `fl`*xxx* commands on the installation server.

   For more information, see the `nhsmctsetup`(1M) man page.

7. **Run the `fldeploy` command.**

   The `fldeploy` command uses the exported data copied from the build server.

   For information on how to run the `fldeploy` command, see "To Create a Deployable Flash Archive and the Solaris JumpStart Environment by Using the `fldeploy` Command" on page 107.

▼ To Transfer the Complete SMCT Environment From the Build Server to the Installation Server

Use this procedure if you plan to modify the existing software load on the build server to build new flash archives on the installation server for deployment to another cluster.

1. **Ensure that the entire SMCT environment on the build server can be accessed from the installation server.**

   Therefore, make sure that the SMCT environment on the build server is shared and exported.

2. **Copy the complete SMCT environment on the build server to a directory on the installation server:**

   ```
   # cp -r build-server-smct-dir installation-server-dir
   ```

   Where *build-server-smct-dir* is the entire SMCT environment on the build server.

   If the flash archives are already stored in the *smct-dir*`/flash` directory, do not transfer them again.

3. **Copy the flash archive(s) from the build server to a directory on the installation server.**

   If the flash archives are already located in *smct-dir*`/flash` on the build server and have already been copied, you do not need to perform this step.

4. **Complete the cluster configuration files as described in Chapter 5.**

5. **On the installation server, run the `nhsmctsetup` command:**

   ```
   # /opt/SUNWcgha/nhsmct/bin/nhsmctsetup -d -w smct-dir -s solaris-dir
   ```

   The `nhsmctsetup` command installs all the `fl`*xxx* and `sl`*xxx* tools on the installation server.

   For more information, see the `nhsmctsetup`(1M) man page.

Modifying `/etc` Files

If one of the following is true, perform the following procedure before running the `fldeploy` command:

- You are using your build server as an installation server
- Your prototype machine is also acting as a master-eligible or a dataless node at deployment time

▼ To Modify the `/etc` Files on the Build Server

- **Modify the `/etc/ethers` and `/etc/hosts` files on the build server to remove any references to the prototype machine.**

  If the existing configuration information is not removed from these files, an error is generated by the `fldeploy` command.

Adding Cluster-Specific Configuration Data to the `machine.conf` File

This section describes how to add cluster-specific configuration data to the `machine.conf` file.

▼ To Add Cluster-Specific Configuration Data to the `machine.conf` File

- **Add the cluster-specific information to the `machine.conf` file for each node group by using the information in the following table.**

TABLE 7–1 The `machine.conf` Parameters Configured in Stage 3

Data	Master-Eligible	Dataless	Diskless
IP Addresses	Yes	Yes	Yes
Ethernet to IP Address links	Yes	Yes	Yes, only if `MAC_ADDR_POLICY` is configured for the diskless node boot policy.
Board Parameters	Yes	Yes	Yes
Switch port Configuration Data	Optional	Optional	Optional
Client ID parameter	No	No	Yes, only if `STATIC_CLIENT_ID_POLICY` is configured for the diskless node boot policy.

▼ To Configure the Ethernet Address of the NIC Connected to the Installation Server

- **In the `machine.conf` file, configure the Ethernet address of the NIC connected to the installation server.**

  In the `network.conf` file, check that the IP address assigned to this NIC is on the same subnet as the installation server.

Adding Cluster-Specific Configuration Data to the `network.conf` File

This section describes how to add cluster-specific network data to the `network.conf` file.

▼ To Add Cluster-Specific Network Data to the `network.conf` File

- **Add the cluster-specific data to the `network.conf` file for each node group by using the information in the following table.**

TABLE 7–2 The `network.conf` Parameters Configured in Stage 3

Data	Master-eligible	Dataless	Diskless
IP addresses	Yes	Yes	Yes
Default routes used to access the cluster	Optional	Optional	Optional
Network address and netmask information for physical network interfaces and virtual network interfaces	Yes	Yes	No

The IP addressing is referred to in the `cluster.conf` and `machine.conf` files with the following keywords:

- `domain` keyword in the `cluster.conf` file
- `nic` and `switch` keywords in the `machine.conf` file

Configuring the CMM and CGTP

The following procedure describes how to configure CMM and CGTP in the `network.conf` file. This must match the CMM and CGTP configuration in the `machine.conf` file.

▼ To Configure CMM and CGTP

1. **Define the three subnetworks required by CGTP.**

   These are two physical networks and one CGTP network, using the `network` configuration element in the `network.conf` file, for example:

   ```
   ELEMENT network phys-A type IPV4 number 10.101.1.0 netmask 255.255.255.0
   ELEMENT network phys-B type IPV4 number 10.101.2.0 netmask 255.255.255.0
   ELEMENT network cgtp type IPV4 number 10.101.3.0 netmask 255.255.255.0
   ```

2. **Assign the following roles to the NIC connections on each node:**

   `CGTP`, `CGTP_NIC0`, and `CGTP_NIC1` in the `machine.conf` file, for example:

   ```
   ELEMENT nic nic0@peerNode1 device hme0 address 8:0:20:da:8a:16
           type PHYSICAL class PRIMARY role CGTP_NIC0
           USE ip peerNode1-2N-nic0

   ELEMENT nic nic1@peerNode1 device hme1
           type PHYSICAL class SECONDARY role CGTP_NIC1
           USE ip peerNode1-2N-nic1

   ELEMENT nic cgtp@peerNode1 device cgtp0
   ```

```
        type VIRTUAL class SECONDARY role CGTP
        USE ip peerNode1-2N-cgtp
```

3. **Configure an address triplet for each node in the `network.conf` file.**

   This is one address per interface. Each address is based on the appropriate subnet, depending on the role of the interface. For example:

```
ELEMENT ip peerNode1-2N-nic0 address 10.101.1.10 type STATIC alias node1
        BELONG_TO network phys-A
ELEMENT ip peerNode1-2N-nic1 address 10.101.2.10 type STATIC
        BELONG_TO network phys-B
ELEMENT ip peerNode1-2N-cgtp alias peerNode1-2N-cgtp.localdomain
        address 10.101.3.10 type STATIC
        BELONG_TO network cgtp
```

4. **Configure the cluster domain.**

   Add the *domainid*, the floating address triplet, abd tge CGTP network to the `cluster.conf` and `network.conf` files.

   For example, in the `cluster.conf` file, specify the following:

```
ELEMENT domain cluster_domain id 101
        INVOLVE network phys-A
                network phys-B
                network cgtp
        USE ip master-cgtp
            ip master-nic0
            ip master-nic1
```

   For example, in the `network.conf` file, specify the following:

```
ELEMENT ip master-cgtp address 10.101.3.1 type FLOATING alias master
        BELONG_TO network cgtp

ELEMENT ip master-nic0 address 10.101.1.1 type FLOATING
        BELONG_TO network phys-A

ELEMENT ip master-nic1 address 10.101.2.1 type FLOATING
        BELONG_TO network phys-B
```

Configuring the Node State Manager

The following procedure describes how to configure the Node State Manager in the `network.conf` file.

▼ To Configure the Node State Manager

1. **In the `machine.conf` file, do the following:**
   - Add a dedicated NIC to the master-eligible nodes.
   - Assign the `role` NSM to the dedicated NIC.

For example:

```
ELEMENT nic nic2@peerNode1 device qfe0
        type PHYSICAL class SECONDARY role NSM
        USE ip external-access
```

2. **In the `network.conf` file, define the associated address and the related network.**

   This address must be configured as a floating address in the `network.conf` file, for example:

```
ELEMENT network external type IPV4 number 129.157.201.0 netmask 255.255.255.0

ELEMENT ip external-access address 129.157.201.10 type FLOATING
        BELONG_TO network external
```

3. **In the `cluster.conf` file, configure the cluster domain.**

```
ELEMENT domain cluster_domain id 100
        INVOLVE network phys-A
                network phys-B
                network cgtp
                network external
        USE ip master-cgtp
            ip master-nic0
            ip master-nic1
```

Configuring a Switch

The following procedure describes how to configure a switch in the `network.conf` file.

▼ To Configure the Switch

1. **Add the configuration element `switch` in the `machine.conf` file.**
2. **Add the port elements with their connections to the NICs on the board in the `machine.conf` file.**
3. **In the `network.conf` file, add the IP addresses of the switches.**

   The IP addresses of the switches may be on a different subnet to the three CGTP subnets. In this case, add the subnet to the cluster domain definition in the `cluster.conf` file and to the `network.conf` file.

Reconfiguring the Disk Controller Type

If your prototype machine does not have exactly the same specification as your target cluster machine, you may need to reconfigure the disk controller type.

▼ To Reconfigure the Disk Controller Type

- **In the configuration element `disk`, change the device parameter value.**

Configuring the Boards and NICs According to the Diskless Node Boot Policy

The following procedure describes how to configure the boards and NICs in the `network.conf` file, according to the diskless node boot policy.

▼ To Configure the Boards and NICs According to the Diskless Node Boot Policy

1. **For the `STATIC_CLIENT_ID_POLICY` boot policy, configure the `clientId` parameter in the configuration element `board`.**
2. **For the `MAC_ADDR_POLICY` boot policy, configure the Ethernet address in the configuration element `nic` for the two physical NICs.**

Adding Cluster-Specific Configuration Data to the `cluster.conf` File

This section describes how to add cluster-specific network data to the `cluster.conf` file. You can define the node IDs in the `cluster.conf` file if you did not do so in stage 1. For more information, see the `cluster.conf`(4) man page.

▼ To Configure the Floating Address Triplet

1. **In the `cluster.conf` file, configure the floating address triplet.**

   The floating address triplet must be a CGTP address named `master-cgtp`, and two IP addresses associated to the two physical networks. In the following program listing, `id` is the ID of the CMM domain.

```
# Cluster domain definition
#
# id -> CMM domainId
# ip -> master master-nic0 master-nic1 floating addresses
#
```

```
ELEMENT domain cluster_domain id 100
        INVOLVE network phys-A
                network phys-B
                network cgtp
                network external
                router default-router
        USE ip master-cgtp
            ip master-nic0
            ip master-nic1
```

2. **Ensure that the floating address triplet is defined in the `network.conf` configuration file.**

   For example:

```
#
# Master floating addresses
#
ELEMENT ip master-cgtp address 10.101.3.1 type FLOATING alias master
    BELONG_TO network cgtp

ELEMENT ip master-nic0 address 10.101.1.1 type FLOATING
    BELONG_TO network phys-A

ELEMENT ip master-nic1 address 10.101.2.1 type FLOATING
    BELONG_TO network phys-B
```

Configuring the Bitmap Partitions

The SMCT automatically configures the bitmap partitions to write updates to memory. The information thus stored in memory is written to the disk when the node is shut down.

However, you also have the option writing the updates to the disk directly. When an update is made to a mounted directory, the update is stored on the disk in the bitmap partition. To change the way the bitmaps are configured, perform the following procedure.

▼ To Change the Configuration of Bitmap Partitions

1. **Log in to the build server.**
2. **Copy the following files from the `/opt/SUNWcgha/nhsmct/etc/services/` directory to your *smct-config-dir*`/services` directory:**
   - `sndr.conf.sh`

- `config.NHAS_MASTER_ELIGIBLE.SPARC.SOLARIS`

3. **Change to the** *smct-config-dir*`/services` **directory.**
4. **Edit the** *smct-config-dir*`/services/sndr.conf.sh` **file:**

   Replace the following line:

   ```
   /usr/bin/sed -e "s,rdc_bitmap_mode=0,rdc_bitmap_mode=2," > $RDC_CONF
   ```

   With the following line:

   ```
   /usr/bin/sed -e "s,rdc_bitmap_mode=0,rdc_bitmap_mode=1," > $RDC_CONF
   ```

5. **Edit the** *smct-config-dir*`/services/config.NHAS_MASTER_ELIGIBLE.SPARC.SOLARIS` **file:**

   Replace the line:

   ```
   location 'file:/NHAS_PROD_DIR/nhsmct/etc/services/sndr.conf.sh'
   ```

   With the line:

   ```
   location 'file:/smct-config-dir/services/sndr.conf.sh'
   ```

Generating the Foundation Services and Solaris Configuration Files

This section describes how to generate the Foundation Services and Solaris operating system configuration files for the cluster by using the `sldeploy` command.

The `sldeploy` command does the following:

- Generates the Foundation Services configuration files.

  The Foundation Services configuration files include the `nhfs.conf` file. The Foundation Services configuration files are generated from the cluster configuration files `cluster.conf`, `machine.conf`, and `network.conf`.
- Generates the Solaris operating system configuration files.

▼ To Generate the Foundation Services and Solaris Configuration Files by Using the `sldeploy` Command

1. **Log in to the build server.**
2. **Run the `sldeploy` command:**

```
$ sldeploy -n SWL1 -v ha_v1
```

The `sldeploy` command is run once for each software load because all node groups are processed in one operation.

For more information on the `sldeploy` command, see the `sldeploy`(1M) man page.

Copying the Foundation Services and Solaris Configuration Files to an Export Directory

Use the `slexport` command to copy the generated Foundation Services and Solaris operating system configuration files created by `sldeploy`, into the export directory. The export directory is used by the `fldeploy` command to generate a deployable flash archive and associated Solaris JumpStart environment.

▼ To Copy the Generated Foundation Services and Solaris Configuration Files to the Export Directory by Using the `slexport` Command

1. **Log in to the build server.**
2. **Run the command `slexport`:**

   ```
   $ slexport -n SWL1 -v ha_v1
   ```

   This command exports the generated Foundation Services and Solaris operating system configuration files in `SWL1`, to the directory `SMCT_EXPORT_DIR/SWL1@ha_v1`.

The `slexport` command copies the software load data to the export directory, including data previously generated by the `slcreate` and `slconfig` commands.

For information, see the `slexport`(1M) man page.

Modifying the Installation Server Configuration File

This section describes how to configure the installation server configuration file, `install-server.conf`, so that the Solaris JumpStart environment can be created by the `fldeploy` command.

▼ To Enable Solaris JumpStart Environment Creation by Configuring the Installation Server Configuration File

1. **Log in to the installation server.**
2. **Modify the `install-server.conf` file to include the cluster addressing schema.**

   The `install-server.conf` file must contain the same network address and same netmask as the target cluster. It must also include an IP address that is not used by the cluster nodes.

   A template for the `install-server.conf` file is provided in the `/opt/SUNWcgha/nhsmct/etc/models` directory on the build server.

   For information, see the `install-server.conf`(4) man page.

Customizing the Solaris JumpStart Files for a Target Cluster

Solaris JumpStart files are required by the master-eligible nodes and dataless nodes. The system identification configuration file contains information for the target cluster such as name service, timezone and superuser password. Templates for the target cluster are provided in the `/opt/SUNWcgha/nhsmct/etc/jumpstart` directory. These templates contain the `.cluster` extension.

Note that the Foundation Services default superuser password is `sunrules`.

▼ To Configure the Solaris JumpStart System Identification File

1. **Copy the `sysidcfg.proto` template from the `/opt/SUNWcgha/nhsmct/etc/jumpstart/` directory to one of the following directories:**
   - *smct-dir*`/jumpstart/`
   - *user-config-dir*`/jumpstart/`
2. **Modify the file `sysidcfg.cluster` to suit your local environment.**

   You can configure the following parameters:
   - `system_locale`
   - `timezone`
   - `terminal`
   - `root_password`
   - `monitor`
   - `keyboard`
   - `display`
   - `pointer`

Note – Although the template files for the `proto.cluster`, `rules.cluster` and `finish.cluster` files are provided, do not configure them. The template for the `begin.cluster` file is supported by the SMCT. Modify this file if required. For more information, see the Solaris JumpStart documentation.

Creating a Deployable Flash Archive and the Solaris JumpStart Environment

This section describes how to create a deployable flash archive by using the `fldeploy` command.

The `fldeploy` command is run on the build server. This command does the following:

- Adds the exported data from the `sldeploy` command to the generic or configured flash archive. The exported data are configuration files for the Foundation Services, including the common configuration file, `nhfs.conf`.
- Generates the Solaris JumpStart environment for each master-eligible node group and dataless node group. The `fldeploy` command is run once for each master-eligible node group and dataless node group in the software load, because

each node group is processed individually.

- Adds a complete cluster definition to the flash archive. This enables the flash archive to be reused without being regenerated. For example, IP addressing and any changes made before running the `sldeploy` command. For further information, see the `sldeploy`(1M) man page.
- Creates a deployable flash archive.

▼ To Create a Deployable Flash Archive and the Solaris JumpStart Environment by Using the `fldeploy` Command

1. **Log in to the installation server as a superuser.**

2. **Run the `fldeploy` command:**

   ```
   # fldeploy -n SWL1 -v ha_v1 -g master_el -f FLASH_1
   ```

   This command generates the flash archive in the default directory defined by the `SMCT_FLASH_DIR` environment variable. It also generates the Solaris JumpStart files in the default directory defined by the `SMCT_JUMPSTART_DIR` environment variable.

   If you are using configuration data to create a flash archive on the installation server, use the `-e` option of the `fldeploy` command. For example:

   ```
   # fldeploy -n swl1 -v ha_v1 -e export-dir -g master_el -f FLASH_1
   ```

The `fldeploy` command can be run several times. You can configure the flash archive for one cluster to test it, then reconfigure the cluster archive information and rerun `fldeploy`.

For information, see the `fldeploy`(1M) man page.

Modifying User-Defined Configuration Data

This section describes how to modify user-defined configuration data added to a flash archive in stage 2. This is the only part of the deployable flash archive that can be modified after it has been created. However, modifications are only possible with a complete SMCT environment. Ensure that you have run the procedure described in "To Transfer the Complete SMCT Environment From the Build Server to the Installation Server" on page 95.

▼ To Modify User-Defined Configuration Data

To modify user-defined configuration data, the `slconfig` and `flconfig` commands must be run again with the modified user application configuration file.

1. **Modify the user application configuration file.**

   For example, `config.master_el`

   For information on how to modify the user application configuration file, see "Creating a User Application Configuration File for a Node Group" on page 85.

2. **Run the `slconfig` command.**

   For information, see "To Prepare the User-Defined Configuration Data by Using the `slconfig` Command" on page 88.

3. **Run the `slexport` command.**

   For information, see "To Copy the User-Defined Configuration Data to the Export Directory by Using the `slexport` Command" on page 89.

4. **Run the `flconfig` command.**

   For information, see "To Add User-Defined Configuration Data to the Flash Archive by Using the `flconfig` Command" on page 89.

5. **If the name or location of the flash archive has changed, run the `fldeploy` command.**

PART IV

Deploying Your Software Load to a Cluster

This part describes how to deploy a software load to a cluster by using the Solaris JumpStart tool.

This part contains the following chapter:

Chapter 8 describes how to create the Solaris JumpStart environment for target cluster. This chapter also describes how to deploy the software load by using Solaris JumpStart.

CHAPTER 8

Deploying Your Software Load to a Cluster

This chapter describes how to create the Solaris JumpStart environment for target cluster. You are then ready to deploy the software load by using Solaris JumpStart.

This chapter contains the following sections:

Software Load Deployment Procedure

The following figure shows the sequence of events that deploy a software load to a cluster:

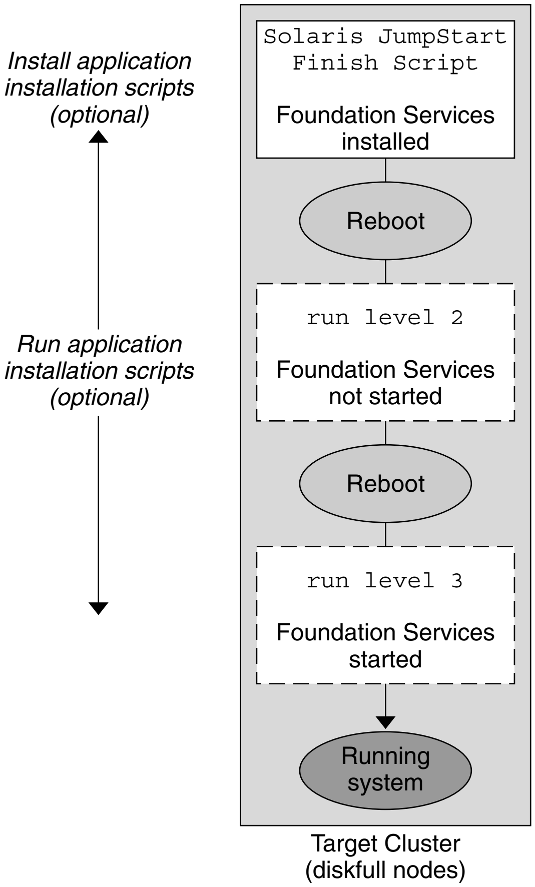
Install application installation scripts (optional)
Run application installation scripts (optional)
Solaris JumpStart Finish Script
Foundation Services installed
Reboot
run level 2
Foundation Services not started
Reboot
run level 3
Foundation Services started
Running system
Target Cluster (diskfull nodes)

FIGURE 8–1 Software Load Deployment Procedure

Verifying that a Flash Archive is Configured Correctly

This section describes how to use the `flar` command to verify that the flash archive is correctly configured.

▼ To Verify that a Flash Archive is Configured Correctly

1. **Log in to the installation server as superuser.**

2. **Run the `flar` command:**

   ```
   # flar -i flash-archive
   ```

   The `flar` command displays information that is stored in the header of a flash archive. This information includes the name and version of the software load and the target node group. The following table gives an example of information in the header of a flash archive:

   TABLE 8–1 Example of Information in the Header of a Flash Archive

Header Information	Description
`X-NHAS-SWL-ID=1`	The software load version
`X-NHAS-SWL-NAME=12N`	Software load name
`X-NHAS-NODE-GROUP-NAME=master_el`	Target node group
`X-NHAS-SOE-RELEASE=Solaris-9-s9_58shwpl3-SPARC`	Solaris operating system associated with the flash archive.

For more information, see the `flar`(1M) man page.

Configuring a Direct Link Between the Master-Eligible Nodes

You can configure a direct link between the master-eligible nodes to prevent a *split brain* situation, where there are two master nodes in the cluster because the cluster network fails.

▼ To Configure a Direct Link Between the Master-Eligible Nodes

1. **Connect the serial ports of the master-eligible nodes.**

   For an illustration of the connection, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

2. **Configure the direct link in the machine.conf configuration file, for example:**

```
ELEMENT board T1105@peerNode1 type T1_105 arch SPARC
                 USE nic nic0@peerNode1
                     nic nic1@peerNode1
                     nic cgtp@peerNode1
                     link serial-b
```

Configuring Diskless Nodes

Before the software load can be deployed, each diskless node must be configured according to the boot policy that was defined in configuration stage 1.

▼ To Configure DHCP Static Boot Policy for a Diskless Node

1. **Get the `ok` prompt on the diskless node.**

   For example, if you are connected to the console with a terminal server using telnet, run:

```
telnet> send brk
ok>
```

2. **Configure the OBP parameters:**

```
ok> setenv boot-device net:dhcp,,,,,5 net2:dhcp,,,,,5
```

▼ To Configure DHCP Client ID Boot Policy for a Diskless Node

1. **In the `machine.conf` file, add the client ID in hexadecimal for the board associated to the diskless node.**

   You can calculate the client ID as follows:

   - Set the board ID at the OBP level
   - Convert it to hexadecimal
   - Add two zeros (`00`) as a prefix

   For example, the ID configured at the OBP level is `11`, therefore, the client ID in hexadecimal in the `machine.conf` file is `003131`:

```
ELEMENT board CP2160@peerNode3 type CP2160 arch SPARC
             clientId 003131
    CONTAIN nic nic0@peerNode3
```

```
            nic nic1@peerNode3
        nic cgtp@peerNode3
```

2. **Get the `ok` prompt on the diskless node.**

   For example, if you are connected to the console with a terminal server using telnet:

   ```
   telnet> send brk
   ok>
   ```

3. **Configure the OBP parameter `dhcp-clientid`:**

   ```
   ok> setenv dhcp-clientid client-id-value
   ```

Verifying that the Installation Server Directories are Shared

Before you boot the cluster, verify that the installation server directories are mounted on the prototype machine and the cluster. For more information, see "To Mount the Installation Server Directories Onto the Prototype Machine and the Cluster" on page 61.

Booting the Target Cluster With the Software Load

This section describes how to deploy the deployable flash archives to a cluster by using the Solaris JumpStart environment and a boot net installation.

The boot net installation installs and starts the deployable flash archives on the cluster.

▼ To Boot the Target Cluster With the Software Load

This procedure boots all of the master-eligible node and dataless nodes at the same time.

1. **Get the `ok` prompt on each master-eligible node and dataless node.**

2. **Perform a boot net install:**

   ```
   ok> boot net - install
   ```

The master-eligible nodes and dataless nodes perform a Solaris JumpStart from the installation server. These nodes then install and run the software load.

PART V

Redeploying a Software Load to a Cluster

This part describes how to redeploy a software load to a cluster. You can also add nodes to a cluster created by using the SMCT.

This part contains the following chapters:

- Chapter 9 lists the reconfiguration options that are available.
- Chapter 10 describes how to add new nodes to your cluster.

CHAPTER 9

Reconfiguring Your Deployment Environment

This chapter describes the reconfiguration tasks that can be performed to reconfigure data in a flash archive.

The following figure describes the reconfiguration options available along with the configuration stages at which these options are run.

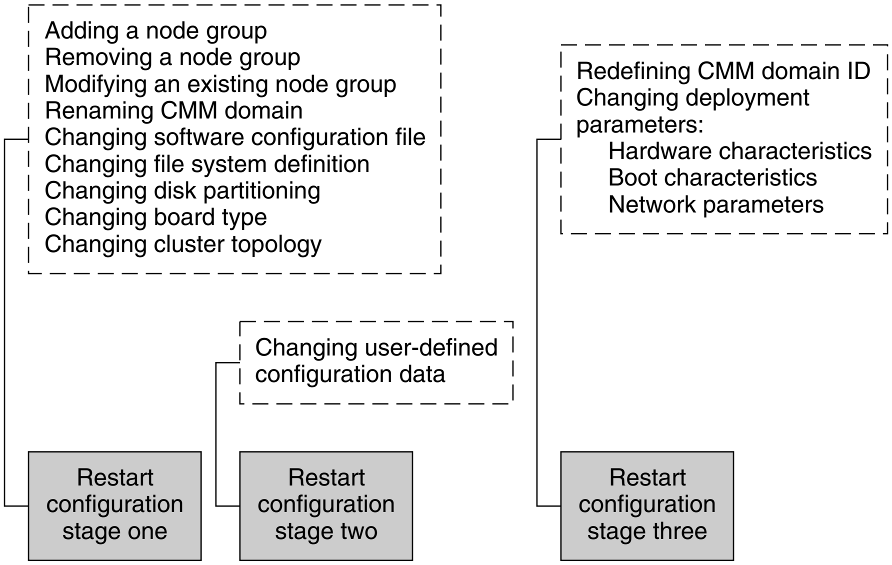
Adding a node group
Removing a node group
Modifying an existing node group
Renaming CMM domain
Changing software configuration file
Changing file system definition
Changing disk partitioning
Changing board type
Changing cluster topology
Redefining CMM domain ID
Changing deployment parameters:
Hardware characteristics
Boot characteristics
Network parameters
Changing user-defined configuration data
Restart configuration stage one
Restart configuration stage two
Restart configuration stage three

FIGURE 9–1 Reconfiguration Options for Each Configuration Stage

---

Note – Do not manually edit Foundation Servicesconfiguration files. If you do manually edit the configuration files, your cluster configuration does not correspond to your flash archive. To change the configuration of a cluster, you must redeploy your flash archive.

CHAPTER 10

Adding a Node to a Cluster Created by the SMCT

For information about how to add a node to a cluster created by the SMCT, see the following sections.

- "Adding a Predefined Diskless Node to a Cluster Created by the SMCT" on page 121
- "Adding a Predefined Dataless Node to a Cluster Created by the SMCT" on page 123
- "Adding an Undefined Node to a Cluster Created by the SMCT" on page 125

Adding a Predefined Diskless Node to a Cluster Created by the SMCT

▼ To Add a Predefined Diskless Node to a Cluster Created by the SMCT

1. **Log in to a peer node as a superuser.**
2. **Determine what nodes are currently running in the cluster:**

   ```
   # nhcmmstat -c all
   ```

   The `nhcmmstat` tool returns information about the nodes currently running in the cluster. This command does not provide information about nodes that are physically present in the cluster, but are down.
3. **Identify the node group and `nodeid` of the node that you want to add.**

   Examine the `cluster.conf` file. The predefined node is present in the `cluster.conf` file, but not present in the output of Step 2.

4. **Verify that the configuration of the node that you want to add corresponds to that of the predefined node.**

   Examine the `machine.conf` file to identify the type of board and network interface card for the predefined node:

   - If the configuration in the `machine.conf` file corresponds to the configuration of the node that you want to add, go to Step 5.
   - If the configuration in the `machine.conf` file does not correspond to the configuration of the node that you want to add, see "Adding an Undefined Node to a Cluster Created by the SMCT" on page 125.

5. **Connect the network interfaces (*NIC0* and *NIC1*) of the new node to the switches.**

   For information, see "Connecting the Cluster Hardware for SMCT Installation" in the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

6. **Access the console of the new node and obtain the `ok` prompt:**

```
# Control-]
telnet> send brk
Type  ’go’ to resume
ok>
```

7. **On the console of the new node, configure the OBP parameters:**

```
ok> setenv local-mac-address ? true
ok> setenv auto-boot-retry? true
ok> setenv diag-switch? false
ok> setenv boot-device net:dhcp,,,,,5 net2:dhcp,,,,,5
```

   If the boot policy set in the `cluster.conf` file is `STATIC_CLIENT_ID_POLICY`, specify the *client-ID*. See "To Configure DHCP Client ID Boot Policy for a Diskless Node" on page 114.

```
ok> setenv dhcp-clientid client-ID
```

8. **Boot the new node:**

```
ok> boot
```

9. **Become superuser on the new node.**

10. **Verify that the node is configured correctly:**

```
# nhadm check
```

Adding a Predefined Dataless Node to a Cluster Created by the SMCT

▼ To Add a Predefined Dataless Node to a Cluster Created by the SMCT

1. **Log in to a peer node as superuser.**

2. **Determine what nodes are currently running in the cluster:**

   ```
   # nhcmmstat -c all
   ```

   The `nhcmmstat` tool returns information about the nodes currently running in the cluster. This command does not provide information about nodes that are physically present in the cluster, but are down.

3. **Identify the node group and `nodeid` of the node that you want to add.**

   Examine the `cluster.conf` file. The predefined node is present in the `cluster.conf` file, but not present in the output of Step 2.

4. **Verify that the configuration of the node that you want to add corresponds to that of the predefined node.**

   Examine the `machine.conf` file to identify the type of board, disk, and network interface card for the predefined node.

5. **Configure the OBP parameters:**

   ```
   ok> setenv local-mac-address ? true
   ok> setenv auto-boot-retry? true
   ok> setenv diag-switch? false
   ```

6. **Verify that the installation server and the new node are on the same subnet and use the same netmask.**

   a. **Identify the network identity and netmask identity for the new node by examining the `network.conf` file.**

   b. **Identify the network identity and netmask identity for the installation server by examining the `install-server.conf` file.**

   c. **Compare the values found in Step a with those found in Step c.**

      - If the values are the same, the installation server and the new node are on the same subnet and use the same netmask.

  - If the values are not the same, modify the content of the `install-server.conf` file. Then repeat stage 3 of the SMCT deployment process.

    For information, see Chapter 7.

d. **Confirm that the hardware is correctly configured.**

   For information, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

7. **Check that the flash archive of the associated node group exists, and do one of the following:**

   - If your flash archive does not exist, rebuild the archive. For information, see Chapter 5.
   - If your flash archive exists but the deployment parameters must be updated, see Chapter 7.
   - If your flash archive exists and the deployment parameters are correct, go to Step 8.

8. **Connect the two Ethernet interfaces (*NIC0* and *NIC1*) of the node to the switches.**

   For information, see the *Netra High Availability Suite Foundation Services 2.1 6/03 Hardware Guide*.

9. **Access the console of the new node and obtain the `ok` prompt:**

```
# Control-]
telnet> send brk
Type  'go' to resume
ok>
```

10. **On the console of the new node, install the flash archive:**

```
ok> boot net - install
```

11. **Become superuser on the new node.**

12. **Verify that the node is configured correctly:**

```
# nhadm check
```

Adding an Undefined Node to a Cluster Created by the SMCT

To add a node to a cluster created by the SMCT, the node must be included in the flash archives. If a node is not included in the flash archives at the initial cluster installation, you must recreate the flash archives to include the new node. When the node is included in the flash archives, the cluster configuration can be redeployed to another cluster or to the same cluster in the case of node failure. For information about redeploying a flash archive, see Chapter 9.

If you add a node to a cluster created by the SMCT without reconfiguring the flash archives, you break the SMCT model. The configuration of that cluster is no longer the same as the flash archive used at cluster startup. It is no longer possible to regenerate the existing configuration by redeploying the flash archive.

APPENDIX **A**

Configuration Element Rules

The following table lists the configuration elements used in the cluster configuration files:

TABLE A–1 Configuration Element Rules

Configuration Element	Configuration Files	Required in Configuration Stage	Can be modified in Configuration Stage
`nodeGroup.*`	`cluster.conf`	1	1
`cluster.*`	`cluster.conf`	1	1
`node.*`	`cluster.conf`	1	1
`node.id`	`cluster.conf`	3	1, 2. 3
`service.*`	`cluster.conf`	1	1
`board.*`	`cluster.conf`	1	1, 2, 3
`domain.*`	`cluster.conf`	3	1, 2, 3
`domain.name`	`cluster.conf`	1	1
`ip.*`	`cluster.conf` `network.conf`	3	1, 2, 3
`router.*`	`cluster.conf` `network.conf`	3	1, 2, 3
`network.*`	`cluster.conf` `network.conf`	3	1, 2, 3
`shelf.*`	`cluster.conf` `machine.conf`	1	1

TABLE A–1 Configuration Element Rules *(Continued)*

Configuration Element	Configuration Files	Required in Configuration Stage	Can be modified in Configuration Stage
`disk.*`	`cluster.conf` `machine.conf`	1	1
`drawer.*`	`machine.conf`	1	1
`disk.device`	`machine.conf`	1	1, 2, 3
`slice.*`	`machine.conf`	1	1
`filesys.*`	`machine.conf`	1	1
`switch.*`	`machine.conf`	3	1, 2, 3
`port.*`	`machine.conf`	3	1, 2, 3
`nic.*`	`machine.conf`	3	1, 2, 3
`board.clientId`	`machine.conf`	3	1,2,3
`link.*`	`machine.conf`	3	1,2,3
`software.*`	`software.conf`	1	1
`config.*`	`userapp.conf`	2	2, 3
`file.*`	`userapp.conf`	2	2, 3

APPENDIX **B**

Configuration File Examples for the `machine.conf` File

The following examples give disk layout and file system configuration sections of the `machine.conf` configuration file:

Example Disk Layout of a Master-Eligible Node Using Volume Management Software

EXAMPLE B–1 Disk Layout for Master-Eligible Node With Diskless Nodes

```
# Disk partitioning (MEN + SVM + diskless)
#
#          Phys. slice  Meta slice  files system
#
#   disk1 -+ s0@disk1 +------------- ROOT
#          + s1@disk1 +------------- SWAP
#          + s3@disk1 +- d3@disk1 +- SHARED
#          + s4@disk1 +- d4@disk1 +- EXPORT
#          + s5@disk1 +- d5@disk1 +- USER2
#          + s6@disk1 +- d6@disk1 +- USER1
#          + s7@disk1 +- d31@disk1
#                     +- d41@disk1
#                     +- d51@disk1
#                     +- d61@disk1
#
ELEMENT disk disk1@peerNode1 device c0t0d0 type SCSI size 18
    CONTAIN slice s0@disk1
        slice s1@disk1
        slice s3@disk1
        slice d3@disk1
        slice d31@disk1
```

EXAMPLE B–1 Disk Layout for Master-Eligible Node With Diskless Nodes *(Continued)*

```
        slice s4@disk1
        slice d4@disk1
        slice d41@disk1
        slice s5@disk1
        slice d5@disk1
        slice d51@disk1
        slice s6@disk1
        slice d6@disk1
        slice d61@disk1
        slice s7@disk1

# Root partition
ELEMENT slice s0@disk1 number 0 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 2048
    MAP filesys ROOT

# Swap partition
ELEMENT slice s1@disk1 number 1 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 1024
    MAP filesys SWAP

# Shared software and data partition (replicated)
ELEMENT slice s3@disk1 number 3 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 2048

ELEMENT slice d3@disk1 number 3 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type META size 2048 role REPLICATED
    USE slice s3@disk1
    MAP filesys SHARED

ELEMENT slice d31@disk1 number 31 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type SOFT size 10 role BITMAP_SNDR
        USE slice s7@disk1
        MANAGE replicatedSlice d3@disk1

# User partition 1 (replicated)
ELEMENT slice s4@disk1 number 4 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 2048

ELEMENT slice d4@disk1 number 4 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type META size 2048 role REPLICATED
    USE slice s4@disk1
    MAP filesys EXPORT

ELEMENT slice d41@disk1 number 41 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type SOFT size 10 role BITMAP_SNDR
        USE slice s7@disk1
        MANAGE replicatedSlice d4@disk1

# User partition 2 (replicated)
ELEMENT slice s5@disk1 number 5 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 2048
```

EXAMPLE B–1 Disk Layout for Master-Eligible Node With Diskless Nodes *(Continued)*

```
ELEMENT slice d5@disk1 number 5 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type META size 2048 role REPLICATED
    USE slice s5@disk1
    MAP filesys USER1

ELEMENT slice d51@disk1 number 51 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type SOFT size 10 role BITMAP_SNDR
        USE slice s7@disk1
        MANAGE replicatedSlice d5@disk1

# User partition 3 (replicated)
ELEMENT slice s6@disk1 number 6 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 2048

ELEMENT slice d6@disk1 number 6 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type META size 2048 role REPLICATED
    USE slice s6@disk1
    MAP filesys USER2

ELEMENT slice d61@disk1 number 61 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type SOFT size 10 role BITMAP_SNDR
        USE slice s7@disk1
        MANAGE replicatedSlice d6@disk1

# Meta-DB + bitmap partition
ELEMENT slice s7@disk1 number 7 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 512 role META_DB
```

EXAMPLE B–2 File Systems Associated With the Master-Eligible Node

This example is for the master-eligible node in Example B–1.

```
# MEN and NMEN file systems
#
#      local       nfs-shared
#
#      ROOT
#      SWAP
#      EXPORT
#      SHARED +- CLUSTER_DATA
#             +- CLUSTER_DATABASE
#             +- SHARED_SOFTWARE
#      USER1
#      USER2
#
# Root file system
ELEMENT filesys ROOT role root type ufs fsck 1 size 2048
        mntPt / mntBt no mntOpt logging
#
# Swap
ELEMENT filesys SWAP role swap type swap  mntBt no size 1024
#
```

EXAMPLE B–2 File Systems Associated With the Master-Eligible Node *(Continued)*

```
# Diskless node shared file system
ELEMENT filesys EXPORT role export type ufs fsck 2 size 2048
        mntPt /export mntBt no mntOpt logging
#
# Shared data and software exported file system
ELEMENT filesys SHARED role shared type ufs fsck 2 size 2048
        mntPt /SUNWcgha/local mntBt no mntOpt logging
    CONTAIN exportedFileSys CLUSTER_DATA
        exportedFileSys CLUSTER_DATABASE
        exportedFileSys SHARED_SOFTWARE
#
# Shared cluster data (e.g. CMM nodes table, ...)
ELEMENT filesys CLUSTER_DATA role data type nfs
        mntPt export/data
        remMntPt /SUNWcgha/remote mntBt no
    mntOpt rw,hard,fg,intr,noac,proto=udp
#
# Shared middleware (e.g. JDMK, application services, ...)
ELEMENT filesys SHARED_SOFTWARE role services type nfs
        mntPt export/services/<--SWLID-->/opt
    remMntPt /SUNWcgha/services mntBt no
    mntOpt rw,hard,fg,intr,noac,proto=udp
#
# Shared application services ldap database
ELEMENT filesys CLUSTER_DATABASE role database type nfs
        mntPt export/services
    remMntPt /SUNWcgha/swdb mntBt no
    mntOpt rw,hard,fg,intr,noac,proto=udp
#
# User 1 defined file system
ELEMENT filesys USER1 role user type ufs fsck 2 size 2048
        mntPt /user1 mntBt no mntOpt logging
#
# User 2 defined file system
ELEMENT filesys USER2 role user type ufs fsck 2 size 2048
        mntPt /user2 mntBt no mntOpt logging
```

EXAMPLE B–3 Disk Layout for Master-Eligible Node Without Diskless Nodes

```
# Disk partitioning (MEN disk1)
#
#          Phys. slice  Meta slice  files system
#
#   disk1 -+ s0@disk1 +------------- ROOT
#          + s1@disk1 +------------- SWAP
#          + s3@disk1 +- d3@disk1 +- SHARED
#          + s4@disk1 +- d4@disk1 +- USER1
#          + s5@disk1 +- d5@disk1 +- USER2
#          + s6@disk1 +- d6@disk1 +- USER1
#          + s7@disk1 +- d31@disk1
#                     +- d41@disk1
#                     +- d51@disk1
#                      +- d61@disk1
#

ELEMENT disk disk1@peerNode1 device c0t0d0 type SCSI size 18
    CONTAIN slice s0@disk1
        slice s1@disk1
        slice s3@disk1
        slice d3@disk1
        slice d31@disk1
        slice s4@disk1
        slice d4@disk1
        slice d41@disk1
        slice s5@disk1
        slice d5@disk1
        slice d51@disk1
        slice s6@disk1
        slice d6@disk1
        slice d61@disk1
        slice s7@disk1
# Root partition
ELEMENT slice s0@disk1 number 0 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 2048
    MAP filesys ROOT

# Swap partition
ELEMENT slice s1@disk1 number 1 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 1024
    MAP filesys SWAP

# Shared software and data partition (replicated)
ELEMENT slice s3@disk1 number 3 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 2048

ELEMENT slice d3@disk1 number 3 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type META size 2048 role REPLICATED
    USE slice s3@disk1
    MAP filesys SHARED

ELEMENT slice d31@disk1 number 31 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type SOFT size 10 role BITMAP_SNDR
        USE slice s7@disk1
        MANAGE replicatedSlice d3@disk1

# User partition 1 (replicated)
ELEMENT slice s4@disk1 number 4 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 2048

ELEMENT slice d4@disk1 number 4 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type META size 2048 role REPLICATED
    USE slice s4@disk1
    MAP filesys USER1

ELEMENT slice d41@disk1 number 41 rawDev /dev/md/rdsk blockDev /dev/md/dsk
```

EXAMPLE B–3 Disk Layout for Master-Eligible Node Without Diskless Nodes *(Continued)*

```
        type SOFT size 10 role BITMAP_SNDR
        USE slice s7@disk1
        MANAGE replicatedSlice d4@disk1

# User partition 2 (replicated)
ELEMENT slice s5@disk1 number 5 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 2048

ELEMENT slice d5@disk1 number 5 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type META size 2048 role REPLICATED
    USE slice s5@disk1
    MAP filesys USER2

ELEMENT slice d51@disk1 number 51 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type SOFT size 10 role BITMAP_SNDR
        USE slice s7@disk1
        MANAGE replicatedSlice d5@disk1

# User partition 3 (replicated)
ELEMENT slice s6@disk1 number 6 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 2048

ELEMENT slice d6@disk1 number 6 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type META size 2048 role REPLICATED
    USE slice s6@disk1
    MAP filesys USER3

ELEMENT slice d61@disk1 number 61 rawDev /dev/md/rdsk blockDev /dev/md/dsk
        type SOFT size 10 role BITMAP_SNDR
        USE slice s7@disk1
        MANAGE replicatedSlice d6@disk1

# Meta-DB + bitmap partition
ELEMENT slice s7@disk1 number 7 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 512 role META_DB
```

Example Disk Layout of a Master-Eligible Node Without Volume Management Software

EXAMPLE B–4 Disk Layout for Master-Eligible Node With Diskless Nodes

```
# Disk partitioning (MEN + diskless)
#
#          Phys. slice              files system
```

```
#
#   disk1 -+ s0@disk1 +------------- ROOT
#          + s1@disk1 +------------- SWAP
#          + s3@disk1 +------------- SHARED (replicated)
#          + s4@disk1 +
#          + s5@disk1 +------------- EXPORT (replicated)
#          + s6@disk1 +
#          + s7@disk1 +------------- USER1
#

ELEMENT disk disk1@peerNode1 device c0t0d0 type SCSI size 18
         CONTAIN slice s0@disk1
                 slice s1@disk1
                 slice s3@disk1
                 slice s4@disk1
                 slice s5@disk1
                 slice s6@disk1
                 slice s7@disk1

# Root partition
ELEMENT slice s0@disk1 number 0 rawDev /dev/rdsk blockDev /dev/dsk
         type PHYS size 2048
    MAP filesys ROOT

# Swap partition
ELEMENT slice s1@disk1 number 1 rawDev /dev/rdsk blockDev /dev/dsk
         type PHYS size 1024
    MAP filesys SWAP

# Shared software and data partition (replicated)
ELEMENT slice s3@disk1 number 3 rawDev /dev/rdsk blockDev /dev/dsk
         type PHYS size 2048 role REPLICATED
    MAP filesys SHARED

ELEMENT slice s4@disk1 number 4 rawDev /dev/rdsk blockDev /dev/dsk
         type PHYS size 10 role BITMAP_SNDR
         MANAGE replicatedSlice s3@disk1

ELEMENT slice s5@disk1 number 5 rawDev /dev/rdsk blockDev /dev/dsk
         type META size 2048 role REPLICATED
    MAP filesys EXPORT

ELEMENT slice s6@disk1 number 6 rawDev /dev/rdsk blockDev /dev/dsk
         type PHYS size 10 role BITMAP_SNDR
         MANAGE replicatedSlice s5@disk1

# User partition 1 (not replicated)
ELEMENT slice s7@disk1 number 7 rawDev /dev/rdsk blockDev /dev/dsk
         type PHYS size 2048
    MAP filesys USER1
```

EXAMPLE B–5 File Systems Associated With the Master-Eligible Node

This example is for the master-eligible node in Example B–4.

```
file systems associated to MEN

# ---------------------------------------------------
# MEN and NMEN file systems
#
#      local       nfs-shared
#
#      ROOT
#      SWAP
#      SHARED +- CLUSTER_DATA
#             +- CLUSTER_DATABASE
#             +- SHARED_SOFTWARE
#      USER1
#
# Root file system
ELEMENT filesys ROOT role root type ufs fsck 1 size 2048
        mntPt / mntBt no mntOpt logging
#
# Swap
ELEMENT filesys SWAP role swap type swap  mntBt no size 1024
#

# Diskless node shared file system
ELEMENT filesys EXPORT role export type ufs fsck 2 size 2048
         mntPt /export mntBt no mntOpt logging

# Shared data and software exported file system
ELEMENT filesys SHARED role shared type ufs fsck 2 size 2048
        mntPt /SUNWcgha/local mntBt no mntOpt logging
    CONTAIN exportedFileSys CLUSTER_DATA
        exportedFileSys CLUSTER_DATABASE
        exportedFileSys SHARED_SOFTWARE
#
# Shared cluster data (e.g. CMM nodes table, ...)
ELEMENT filesys CLUSTER_DATA role data type nfs
        mntPt export/data
        remMntPt /SUNWcgha/remote mntBt no
    mntOpt rw,hard,fg,intr,noac,proto=udp
#
# Shared middleware (e.g. JDMK, application services, ...)
ELEMENT filesys SHARED_SOFTWARE role services type nfs
        mntPt export/services/<--SWLID-->/opt
    remMntPt /SUNWcgha/services mntBt no
    mntOpt rw,hard,fg,intr,noac,proto=udp
#
# Shared application services ldap database
ELEMENT filesys CLUSTER_DATABASE role database type nfs
        mntPt export/services
    remMntPt /SUNWcgha/swdb mntBt no
    mntOpt rw,hard,fg,intr,noac,proto=udp
#
# User 1 defined file system
ELEMENT filesys USER1 role user type ufs fsck 2 size 2048
```

EXAMPLE B–5 File Systems Associated With the Master-Eligible Node *(Continued)*

```
        mntPt /user1 mntBt no mntOpt logging
```

EXAMPLE B–6 Disk Layout for Master-Eligible Node Without Diskless Nodes

```
# Disk partitioning (MEN w/o diskless)
#
#          Phys. slice              files system
#
#   disk1 -+ s0@disk1 +------------- ROOT
#          + s1@disk1 +------------- SWAP
#          + s3@disk1 +------------- SHARED (replicated)
#          + s4@disk1 +
#          + s5@disk1 +------------- USER1 (replicated)
#          + s6@disk1 +
#          + s7@disk1 +------------- USER2
#

ELEMENT disk disk1@peerNode1 device c0t0d0 type SCSI size 18
         CONTAIN slice s0@disk1
                 slice s1@disk1
                 slice s3@disk1
                 slice s4@disk1
                 slice s5@disk1
                 slice s6@disk1
                 slice s7@disk1

# Root partition
ELEMENT slice s0@disk1 number 0 rawDev /dev/rdsk blockDev /dev/dsk
         type PHYS size 2048
    MAP filesys ROOT

# Swap partition
ELEMENT slice s1@disk1 number 1 rawDev /dev/rdsk blockDev /dev/dsk
         type PHYS size 1024
    MAP filesys SWAP

# Shared software and data partition (replicated)
ELEMENT slice s3@disk1 number 3 rawDev /dev/rdsk blockDev /dev/dsk
         type PHYS size 2048 role REPLICATED
    MAP filesys SHARED

ELEMENT slice s4@disk1 number 4 rawDev /dev/rdsk blockDev /dev/dsk
         type PHYS size 10 role BITMAP_SNDR
         MANAGE replicatedSlice s3@disk1

ELEMENT slice s5@disk1 number 5 rawDev /dev/rdsk blockDev /dev/dsk
         type META size 2048
    MAP filesys USER1

ELEMENT slice s6@disk1 number 6 rawDev /dev/rdsk blockDev /dev/dsk
         type PHYS size 10 role BITMAP_SNDR
         MANAGE replicatedSlice s5@disk1
```

```
# User partition 1 (not replicated)
ELEMENT slice s7@disk1 number 7 rawDev /dev/rdsk blockDev /dev/dsk
         type PHYS size 2048
    MAP filesys USER2
```

EXAMPLE B–7 File Systems Associated With the Master-Eligible Node

This example is for the master-eligible node in Example B–6.

```
# MEN and NMEN file systems
#
#      local       nfs-shared
#
#      ROOT
#      SWAP
#      SHARED +- CLUSTER_DATA
#             +- CLUSTER_DATABASE
#             +- SHARED_SOFTWARE
#      USER1
#      USER2
#
# Root file system
ELEMENT filesys ROOT role root type ufs fsck 1 size 2048
        mntPt / mntBt no mntOpt logging
#
# Swap
ELEMENT filesys SWAP role swap type swap  mntBt no size 1024
#

# Diskless node shared file system
ELEMENT filesys EXPORT role export type ufs fsck 2 size 2048
         mntPt /export mntBt no mntOpt logging

# Shared data and software exported file system
ELEMENT filesys SHARED role shared type ufs fsck 2 size 2048
        mntPt /SUNWcgha/local mntBt no mntOpt logging
    CONTAIN exportedFileSys CLUSTER_DATA
        exportedFileSys CLUSTER_DATABASE
        exportedFileSys SHARED_SOFTWARE
#
# Shared cluster data (e.g. CMM nodes table, ...)
ELEMENT filesys CLUSTER_DATA role data type nfs
        mntPt export/data
        remMntPt /SUNWcgha/remote mntBt no
    mntOpt rw,hard,fg,intr,noac,proto=udp
#
# Shared middleware (e.g. JDMK, application services, ...)
ELEMENT filesys SHARED_SOFTWARE role services type nfs
        mntPt export/services/<--SWLID-->/opt
    remMntPt /SUNWcgha/services mntBt no
    mntOpt rw,hard,fg,intr,noac,proto=udp
```

EXAMPLE B–7 File Systems Associated With the Master-Eligible Node *(Continued)*

```
#
# Shared application services ldap database
ELEMENT filesys CLUSTER_DATABASE role database type nfs
        mntPt export/services
    remMntPt /SUNWcgha/swdb mntBt no
    mntOpt rw,hard,fg,intr,noac,proto=udp
#
# User 1 defined file system
ELEMENT filesys USER1 role user type ufs fsck 2 size 2048
        mntPt /user1 mntBt no mntOpt logging
#
# User 2 defined file system
ELEMENT filesys USER2 role user type ufs fsck 2 size 2048
        mntPt /user1 mntBt no mntOpt logging
```

EXAMPLE B–8 Disk Layout for a Dataless Node

```
# Disk partitioning (dataless disk1)
#
#          Phys. slice  files system
#
#   disk1 -+ s0@disk1 +- ROOT
#          + s1@disk1 +- SWAP
#          + s3@disk1 +- USER1
#

ELEMENT disk disk1@peerNode3 device c0t0d0 type SCSI size 18
    CONTAIN slice s0@disk1@dataless
        slice s1@disk1@dataless
        slice s3@disk1@dataless

# Root partition
ELEMENT slice s0@disk1@dataless number 0 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 2048
    MAP filesys ROOT

# Swap partition
ELEMENT slice s1@disk1@dataless number 1 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 1024
    MAP filesys SWAP

# User defined parition
ELEMENT slice s3@disk1@dataless number 3 rawDev /dev/rdsk blockDev /dev/dsk
        type PHYS size 10240
    MAP filesys USER1
```

EXAMPLE B–9 File Systems for Dataless Nodes and Associated Master-Eligible Nodes

This example is for the dataless node in Example B–8.

```
# MEN and NMEN file systems
#
#      local       nfs-shared
```

EXAMPLE B–9 File Systems for Dataless Nodes and Associated Master-Eligible Nodes *(Continued)*

```
#
#       ROOT
#       SWAP
#       SHARED +- CLUSTER_DATA
#              +- CLUSTER_DATABASE
#              +- SHARED_SOFTWARE
#       USER1
#       USER2
#       USER3
#
# Root file system
ELEMENT filesys ROOT role root type ufs fsck 1 size 2048
        mntPt / mntBt no mntOpt logging
#
# Swap
ELEMENT filesys SWAP role swap type swap  mntBt no size 1024
#
# Shared data and software exported file system
ELEMENT filesys SHARED role shared type ufs fsck 2 size 2048
        mntPt /SUNWcgha/local mntBt no mntOpt logging
    CONTAIN exportedFileSys CLUSTER_DATA
        exportedFileSys CLUSTER_DATABASE
        exportedFileSys SHARED_SOFTWARE
#
# Shared cluster data (e.g. CMM nodes table, ...)
ELEMENT filesys CLUSTER_DATA role data type nfs
        mntPt export/data
        remMntPt /SUNWcgha/remote mntBt no
    mntOpt rw,hard,fg,intr,noac,proto=udp
#
# Shared middleware (e.g. JDMK, application services, ...)
ELEMENT filesys SHARED_SOFTWARE role services type nfs
        mntPt export/services/<--SWLID-->/opt
    remMntPt /SUNWcgha/services mntBt no
    mntOpt rw,hard,fg,intr,noac,proto=udp
#
# Shared application services ldap database
ELEMENT filesys CLUSTER_DATABASE role database type nfs
        mntPt export/services
    remMntPt /SUNWcgha/swdb mntBt no
    mntOpt rw,hard,fg,intr,noac,proto=udp
#
# User 1 defined file system
ELEMENT filesys USER1 role user type ufs fsck 2 size 2048
        mntPt /user1 mntBt no mntOpt logging
#
# User 2 defined file system
ELEMENT filesys USER2 role user type ufs fsck 2 size 2048
        mntPt /user2 mntBt no mntOpt logging
#
# User 3 defined file system
ELEMENT filesys USER3 role user type ufs fsck 2 size 2048
```

EXAMPLE B–9 File Systems for Dataless Nodes and Associated Master-Eligible Nodes *(Continued)*

```
        mntPt /user3 mntBt no mntOpt logging
```

Index