Novell
Linux Point of Service

9

www.novell.com

ADMINISTRATION GUIDE

August 15, 2006

Novell®

Legal Notices

Novell, Inc. makes no representations or warranties with respect to the contents or use of this documentation, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to revise this publication and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes.

Further, Novell, Inc. makes no representations or warranties with respect to any software, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc. reserves the right to make changes to any and all parts of Novell software, at any time, without any obligation to notify any person or entity of such changes.

You may not use, export, or re-export this product in violation of any applicable laws or regulations including, without limitation, U.S. export regulations or the laws of the country in which you reside. See the Novell International Trade Services Web page (http://www.novell.com/info/exports/) for more information on exporting Novell software.

Novell, Inc. has intellectual property rights relating to technology embodied in the product that is described in this document. In particular, and without limitation, these intellectual property rights may include one or more of the U.S. patents listed on the Novell Legal Patents Web page (http://www.novell.com/company/legal/patents/) and one or more additional patents or pending patent applications in the U.S. and in other countries.

Novell, Inc.
404 Wyman Street, Suite 500
Waltham, MA 02451
U.S.A.
www.novell.com

Online Documentation: To access the online documentation for this and other Novell products, and to get updates, see www.novell.com/documentation.

Novell Trademarks

For a list of Novell trademarks, see the online Trademark and Service Mark List (http://www.novell.com/company/legal/trademarks/tmlist.html).

Third-Party Materials

All third-party trademarks are the property of their respective owners.

Contents

About This Guide

Welcome to Novell® Linux Point of Service. This Administration Guide provides information on how to manage a Novell Linux Point of Service retail system.

Audience

This documentation targets Linux* system administrators. It assumes a proficient knowledge of the Linux operating system and administration procedures.

Feedback

We want to hear your comments and suggestions about this manual and the other documentation included with this product. To contact us, use the User Comment feature at the bottom of each page of the online documentation, or go to www.novell.com/documentation/feedback.html and enter your comments there.

Documentation Updates

For the most recent version of the documentation, see the *Novell Linux Point of Service Administration Guide* (http://www.novell.com/documentation/nlpos9/index.html)

Additional Documentation

For information on installing and configuring Novell Linux Point of Service components, see the *Novell Linux Point of Service 9 Installation Guide*.

Documentation Conventions

In Novell documentation, a greater-than symbol (>) is used to separate actions within a step and items in a cross-reference path.

A trademark symbol (®, ™, etc.) denotes a Novell trademark. An asterisk (*) denotes a third-party trademark.

When a single pathname can be written with a backslash for some platforms or a forward slash for other platforms, the pathname is presented with a backslash. Users of platforms that require a forward slash, such as Linux or UNIX, should use forward slashes as required by your software.

Product Overview

1

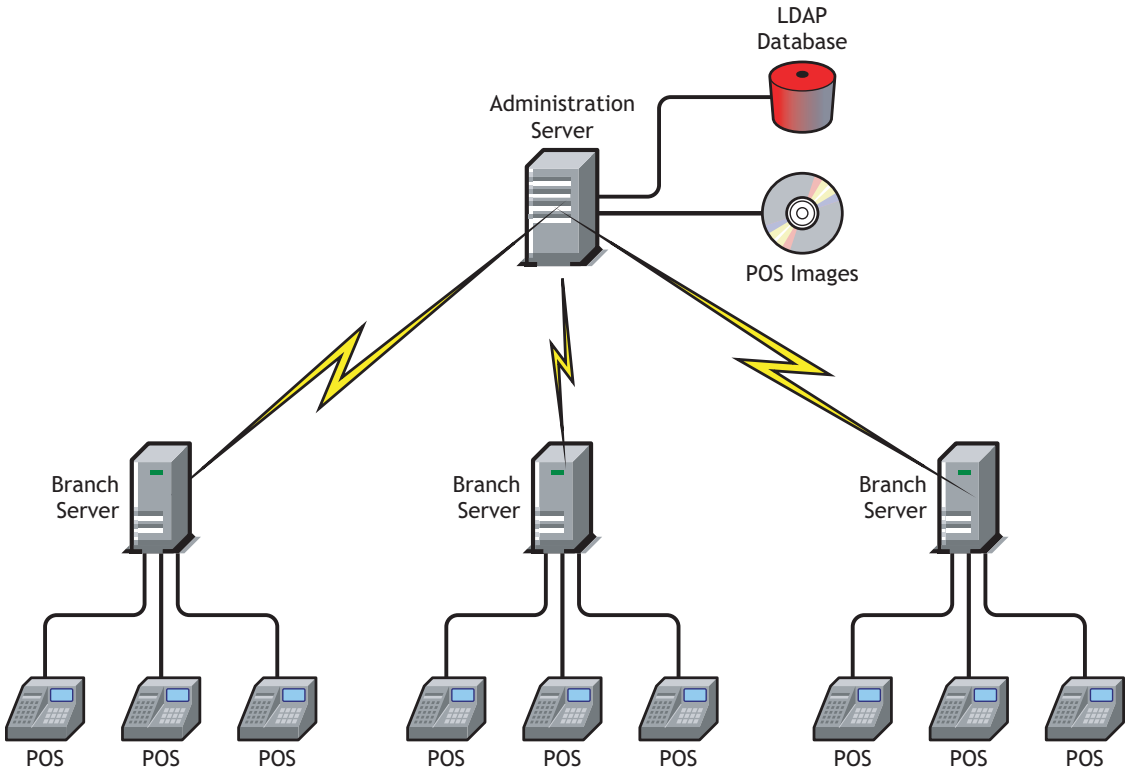
Novell® Linux Point of Service 9 is a secure and reliable Linux platform optimized for enterprise retail organizations. Built on the solid foundation of SUSE® Linux Enterprise Server 9 and Novell Linux Desktop 9, it is the only enterprise-class Linux operating system tailored specifically for retail point of service terminals, in-store servers, kiosk and self-service systems, and reverse-vending systems. It features a scalable deployment infrastructure and a centralized management system.

This section provides an architectural overview of Novell Linux Point of Service, along with a discussion of dependencies within the product.

1.1 Architectural Overview

The Novell Linux Point of Service architecture consists of one centralized Administration Server, one or more Branch Servers, and Point of Service terminals, which can be standard PCs running retail check-out applications or specialized point-of-sale machines such as cash registers and customer kiosks (Figure 1-1).

Figure 1-1 *Novell Linux Point of Service system architecture*



All system information (system structure, the configuration and deployment method for each Branch Server and Point of Service terminal, image information, and so forth) is stored in an LDAP database on the Administration Server. The Administration Server is also the master repository for the images required to boot and configure Point of Service terminals and it provides the utilities required to build those images.

NOTE: By default, the utilities required to build Point of Service images are installed as part of the Administration Server installation. If you have a large system and want to offload the image building function from the Administration Server, you can create a dedicated image building server. For more information, see "Setting Up the Administration Server" or "Setting Up a Dedicated Image Building Server"in the *Novell Linux Point of Service 9 Installation Guide*.

During the initial configuration, each Branch Server downloads the system information and images required for its local Point of Service terminals from the Administration Server. The Point of Service terminals, in turn, download their respective images from the Branch Server when they boot.

WARNING: Because Branch Servers contain sensitive information, they should be secured. You should close unused ports and allow only the root user to have access to the server console.

Novell Linux Point of Service is broadly scalable so that a small shop with five Point of Service terminals can be managed just as well as a large chain with a thousand branches. For organizations with several Branch Servers, the link between the branch and administrative servers is maintained over WAN links. During execution of administrative tasks, such as installation of new Point of Service terminals in a branch, steps must be taken to ensure that the WAN link to the Administration Server is available.

The Novell Linux Point of Service architecture is highly centralized; however, administrative tasks can be delegated to subunits for role-based administration. Moreover, although the LDAP directory is not replicated on the Branch Server, the Branch Server provides all the services necessary for the operation and management of the Point of Service terminals. Consequently, the Branch Server and Point of Service terminals can function independently of the Administration Server in the event of server failure or a downed connection.

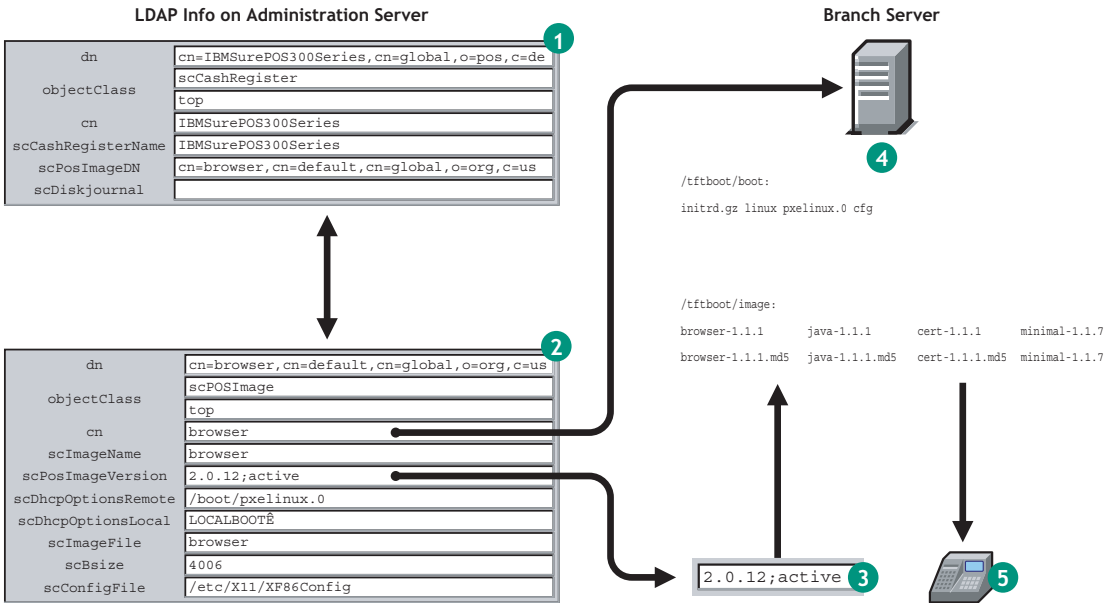
The following sections review each component of the Novell Linux Point of Service architecture.

- Chapter 2, "Novell Linux Point of Service Servers," on page 17 reviews the Administration, Branch, and POSBranch Servers.
- Chapter 3, "Point of Service Terminals," on page 25 provides detailed information on Point of Service terminals.
- Chapter 4, "Point of Service Images," on page 41 provide information about images and the image templates provided with Novell Linux Point of Service.
- Chapter 5, "The Novell Linux Point of Service LDAP Directory," on page 55 provides information about the LDAP objects used to configure and manage Novell Linux Point of Service.

1.2 Dependencies Between LDAP, Branch Server, and Point of Service Terminal

Figure 1-2 illustrates the dependencies between LDAP, the Branch Server, and Point of Service terminal.

Figure 1-2 *Dependencies between LDAP, Branch Server, and Point of Service terminal*



1. Every type of terminal in the Novell Linux Point of Service network (for example, IBM SurePOS 300 Series) must have an associated Hardware Reference object (scCashRegister) in the LDAP directory. The scCashRegister object stores information about Point of Service hardware and associates the terminal type with a specific image object (scPosImage).
2. Every Point of Service image must have an associated Image Reference object (scPosImage) in the LDAP directory.
3. The scPosImageVersion attribute in the Image Reference object (scPosImage) must be set to active in LDAP before the Branch Server can download the corresponding image from the Administration Server. For instructions on activating client images, see Section 6.5.2, “Activating Images,” on page 81.
4. When the Image Reference object is set to active, the Branch Server can download the corresponding image file from the Administration Server to its local `/tftpboot/` directory. The Branch Server stores boot images in the `/tftpboot/boot/` directory. Client images are stored in the `/tftpboot/image/` directory.
5. At boot time, the Branch Server provides the boot and client images required to boot the Point of Service terminal.

Novell Linux Point of Service Servers

2

This section provides an overview of the three types of servers in a Novell® Linux Point of Service system.

2.1 Administration Server

The Administration Server is the central administration point for Novell Linux Point of Service. It provides the following services:

- Maintains the master LDAP directory for the Branch Server systems.

  For more information on the Novell Linux Point of Service LDAP directory, see “The Novell Linux Point of Service LDAP Directory” on page 55.
- Provides the ImageBuilder tools (scr and xscr) to create and customize client images.

  For more information, see Chapter 8, “Building Images with the scr ImageBuilder Tool,” on page 97 or Chapter 9, “Building Images with the xscr ImageBuilder Tool,” on page 121.
- Stores the configuration parameters for the Branch Servers.
- Stores the client images for distribution to the Branch Servers and Point of Service terminals.
- Provides an RSYNC server to distribute the client images and software updates to the Branch Server systems.
- Supports NTP time synchronization for the Branch Servers.
- Consolidates the syslog output from the Branch Servers (optional).

The following sections provide basic information about the Administration Server structure and functions:

For information on installing and configuring the Administration Server, see “Setting Up the Administration Server” in the *Novell Linux Point of Service 9 Installation Guide*.

2.1.1 Operating System

The operating system that runs the Administration Server is built from a standard SUSE® Linux Enterprise Server 9 (SLES 9) source.

2.1.2 LDAP Directory

The Administration Server stores the Novell Linux Point of Service LDAP directory. The LDAP directory is the repository for all system information (system structure, the configuration and deployment method for each Branch Server, client image information, and Point of Service terminal types).

NOTE: The Administration Server does not have an associated object in the LDAP tree structure.

The Novell Linux Point of Service LDAP directory can run on OpenLDAP, Novell eDirectory™, or IBM* Directory Services.

For more information, see Chapter 5, "The Novell Linux Point of Service LDAP Directory," on page 55.

2.1.3 Administrative Tasks

The Administration Server is the central administration point for Novell Linux Point of Service. Because it maintains the LDAP directory, the Administration Server provides the information required to set up the Branch Servers and Point of Service terminals. It is also the staging point for creating and distributing client images, unless you install this functionality on a dedicated image building server.

Branch Server Setup

After Novell Linux Point of Service is installed on the Branch Server, the Branch Server must be able to connect to the Administration Server to download its own configuration settings as well as the system information and images required for its local Point of Service terminals.

For more information on Branch Server configuration, see "Setting Up a Branch Server" in the *Novell Linux Point of Service 9 Installation Guide*.

Client Image Creation and Distribution

Client images are created using the ImageBuilder tools (scr or xscr) that can be installed either on the Administration Server or on a dedicated image building server. In either case, images are typically stored on the Administration Server in the `/opt/SLES/POS/image/` directory. Active images—that is, images that are available to distribute to Branch Servers—are located in the server's RSYNC directory. Specifically, boot images are located in `/opt/SLES/POS/rsync/boot/`; client images are located in `/opt/SLES/POS/rsync/image/`.

For information on creating, storing, and distributing client images, see Chapter 8, "Building Images with the scr ImageBuilder Tool," on page 97 or Chapter 9, "Building Images with the xscr ImageBuilder Tool," on page 121.

For more information on creating a dedicated image building server, see "Setting Up a Dedicated Image Building Server" in the *Novell Linux Point of Service 9 Installation Guide*.

2.1.4 Services

The Administration Server provides two important services in a Novell Linux Point of Service system:

- LDAP is the protocol for accessing the Novell Linux Point of Service directory, which stores all system information.
- RSYNC is a remote data synchronization service that is used to transfer images from the Administration Server to the Branch Servers.

2.2 Branch Server

The Branch Server provides the network boot and system management infrastructure for the Point of Service terminals. It can also serve as a generic system platform for in-store applications such as database systems and back-ends for Point of Service applications.

In a Novell Linux Point of Service system, the Branch Server provides the following services:

- Runs DNS services for the local network.
- Runs DHCP so it can control the network boot process.
- Provides a multicast boot infrastructure for Point of Service terminals.
- Transfers client images from the Administration Server to the Point of Service terminals.

  The Branch Server uses a software distribution mechanism based on RSYNC to pull new client images from the Administration Server. It then uses TFTP to download client images and configuration files to the Point of Service terminals.
- Manages diskless and disk-based Point of Service terminals. Configuration data is taken from the LDAP directory on the Administration Server.
- Provides AutoYaST installation and online updates for the Branch Server operating system.
- Provides system redundancy and failover. A pair of Branch Servers can be configured as a two-node high availability cluster with replicated data.
- Supports NTP for time synchronization from the Administration Server.
- Supports SNMP. Standard MIB2 monitoring is set up with net-snmp (optional).
- Logs syslog output from the Point of Service terminals (optional).

The following sections provide basic information about Branch Server structure and functions:

- Section 2.2.1, “Operating System,” on page 20
- Section 2.2.2, “LDAP Branch Server Object,” on page 20
- Section 2.2.3, “LDAP Access,” on page 20
- Section 2.2.4, “Administrative Tasks,” on page 20
- Section 2.2.5, “Services,” on page 21
- Section 2.2.6, “High Availability Configuration,” on page 21
- Section 2.2.7, “TFTP Server Directory Structure,” on page 22

For information on installing and configuring the Branch Server, see “Setting Up a Branch Server” in the *Novell Linux Point of Service 9 Installation Guide*.

2.2.1 Operating System

The operating system for the Branch Server is built from a standard SLES 9 source.

If the Branch Server is required to run only the Point of Service infrastructure, it can be deployed as a control terminal running on Point of Service hardware. For more information on this configuration, see Section 2.3, “POSBranch Server,” on page 23.

2.2.2 LDAP Branch Server Object

Each Branch Server has a corresponding Branch Server object (scBranchServer) in the LDAP directory. This object stores configuration information that is specific to each Branch Server.

For more information on the scBranchServer object, see Chapter 5, “The Novell Linux Point of Service LDAP Directory,” on page 55.

2.2.3 LDAP Access

To complete its initial configuration and perform basic functions such as registering Point of Service terminals and downloading client images and configuration files, the Branch Server must have administrator level access to the LDAP directory. This admin account and password are created by the `posInitLdap.sh` or `posInitEdir.sh` script during the initial configuration of the Administration Server. Once created, this account is not accessible in the LDAP tree.

LDAP communications can be secured with SSL. When you run the `posInitLdap.sh` script, you can enable or disable SSL communication. If SSL is enabled, you must configure the scPubKey attribute in the scBranchServer object.

NOTE: The posInitEdir script does not provide SSL functionality.

2.2.4 Administrative Tasks

Other than emergency handling, no system administration is necessary on the Branch Server. All administrative tasks are controlled from the central Administration Server or are regularly executed by daemons running on the Branch Server. For emergencies and debugging, all administrative functions can be triggered locally or via SSH login by calling scripts with no or few command line parameters.

If you need to update the Point of Service images stored on the Branch Server, you can run `possyncimages.pl` to manually trigger the RSYNC update process and download new image files from the Administration Server. For more information, see Section A.3.10, “possyncimages.pl,” on page 208.

NOTE: The Branch Server can simultaneously distribute SLRS 8 and Novell Linux Point of Service 9 Point of Service images.

Similarly, if you need to update the Point of Service hardware configuration information stored on the Branch Server, run either `posldap2crconfig.pl --dumpall` or `posAdmin --updateconfig`. These commands regenerate the hardware configuration and `config.`*`MAC_address`* files for all Point of Service terminals found in LDAP.

For more information on the `posldap2crconfig.pl` script, see Section A.3.5, “posldap2crconfig.pl,” on page 206.

For more information on the `posAdmin --updateconfig` command, see Section 6.9, “Updating config.MAC_address and Hardware Configuration Files,” on page 85.

2.2.5 Services

In a Novell Linux Point of Service system, Branch Servers provide the services listed in Table 2-1.

Table 2-1 *Branch Server services*

Service	Description
DNS	Every Branch Server runs a DNS master for that branch. The posldap2dns script generates the zone files for the BIND name server from the data in the LDAP directory and then reloads the zone files on each Branch Server.
DHCP	A DHCP server is installed on the Branch Server. The posldap2dhcp script generates the `dhcpd.conf` file from branch data in the LDAP directory.
NTP	The NTP service for the Branch Servers synchronizes with the Administration Server NTP, which must be configured to get time from a reliable source.
RSYNC	RSYNC is used to transfer SLRS 8 and Novell Linux Point of Service 9 images to the Branch Servers. The Branch Servers pull the images from the Administration Server by using the possyncimages script.
TFTP	The TFTP service on the Branch Server is structured with boot, image, Point of Service, and upload directories. There is a PXE default configuration with which all the Point of Service terminals first load the same initial initrd and the same kernel. For more information, see Section 2.2.7, “TFTP Server Directory Structure,” on page 22. If there is an error with a TFTP action, the service waits 60 seconds, then restarts.
Syslog	The Branch Server can define syslog logging services for Point of Service terminals. This service must be manually defined; the configuration information is stored in the `/etc/syslog.conf` file, not in LDAP.

2.2.6 High Availability Configuration

For high availability, Branch Servers can be configured in two-node heartbeat pairs. The primary node runs all of the scripts and services required to download Branch Server configuration information, synchronize time, and download client images from the Administration Server. The secondary node stays synchronized with the primary, ready to take over and run the scripts and services if the primary fails.

To make the Branch Server services highly available, either the generic mechanisms of the server services (DNS, DHCP, etc.) are used or a combination of heartbeat, virtual IP, and DRDB is employed. The configuration data (DHCP leases) and application data (Point of Service application database back-end tables) are synchronized with DRBD.

For information on installing a high availability Branch Server pair, see “Setting Up High Availability Branch Servers” in the *Novell Linux Point of Service 9 Installation Guide*.

For information on adding high availability Branch Server objects to the LDAP directory, see Section 6.3.4, “Adding a Branch Server with High Availability Services (scHAService),” on page 71.

2.2.7 TFTP Server Directory Structure

Novell Linux Point of Service uses `/tftpboot` as the *tftp_root* path for the TFTP server on the Branch Server. Table 2-2 outlines the main areas that the directory structure is divided into under the TFTP root directory.

Table 2-2 *TFTP directory structure on the Branch Server*

Directory	Contents
`/tftpboot/CR/`	Contains `config.`*`MAC_ Address`* image configuration files for every registered Point of Service terminal on the current Branch Server.
`/tftpboot/CR/` *`MAC_Address/`*	Contains system configuration files, such as `XF86config`, for the individual Point of Service terminals.
`/tftpboot/boot/`	Contains the following boot images and configuration files for Point of Service terminals: `initrd.gz`, `linux`, the PXE loader (`pxelinux.0`), and the PXE configuration folder (`pxelinux.cfg`).
`/tftpboot/image/`	Contains client image files and their checksums.
`/tftpboot/upload/`	Serves as the destination directory to upload `hwtype.`*`MAC_ Address`* files for newly registered Point of Service terminals. These files are used to create the Point of Service terminal’s workstation object in LDAP. This directory also stores the *`bootversion.MAC_address`* files that the posleases2ldap daemon uses to provide image install notification. When an image is successfully installed on a Point of Service terminal, the linuxrc script creates a *`bootversion.MAC_Address`* file in the `/tftpboot/upload` directory on the Branch Server. posleases2ldap then transfers the information to the scNotifiedimage attribute in the scWorkstation object in LDAP and deletes the *`bootversion.MAC_Address`* file.

An example of a Branch Server TFTP structure is shown below:

```
/tftpboot/CR
  00:02:55:E8:FA:C9   config.00:02:55:E8:FA:C9
  00:03:56:01:D5:5F   config.00:03:56:01:D5:5F
  00:09:6B:3B:01:07   config.00:09:6B:3B:01:07
  00:02:55:23:F3:93   config.00:02:55:23:F3:93

/tftpboot/CR/00:02:55:E8:FA:C9
  XF86Config

/tftpboot/CR/00:03:56:01:D5:5F
  XF86Config

/tftpboot/CR/00:09:6B:3B:01:07

/tftpboot/boot
```

```
  initrd.gz
  linux
  pxelinux.0
  pxelinux.cfg

/tftpboot/boot/pxelinux.cfg
  default

/tftpboot/image
  browser-2.0.21    browser-2.0.21.md5
  desktop-2.0.21    desktop-2.0.21.md5
  java-2.0.21       java-2.0.21.md5
  minimal-2.0.21    minimal-2.0.21.md5

/tftpboot/upload
  hwtype.00:02:55:E8:FA:C9
```

NOTE: The Point of Service control file `hwtype.00:02:55:E8:FA:C9` is deleted after successful registration in LDAP. For more information, see Section 3.5.3, “The hwtype.MAC_address File,” on page 34.

2.3 POSBranch Server

For small stores where the Branch Server runs only the Point of Service infrastructure, the Branch Server can be deployed as a control terminal running on Point of Service hardware. Although the POSBranch Server configuration is designed for systems that do not run Point of Service applications, it can run some applications if the terminal has sufficient memory and disk space.

NOTE: This implementation of the POSBranch Server allows the Point of Service applications to run under a non-root account.

The POSBranch Server definition is predefined in the `branch.xml` image template.

The following sections provide basic information about POSBranch Server structure and functions:

- Section 2.3.1, “Hardware,” on page 24
- Section 2.3.2, “Operating System,” on page 24
- Section 2.3.3, “High Availability Configuration,” on page 24
- Section 2.3.4, “LDAP Objects,” on page 24
- Section 2.3.5, “LDAP Access,” on page 24
- Section 2.3.6, “Administrative Tasks,” on page 24
- Section 2.3.7, “Services,” on page 24
- Section 2.3.8, “TFTP Directory Structure,” on page 24

For detailed information on the POSBranch image, see Section 4.5, “POSBranch Images,” on page 51.

For information on building a POSBranch image, see Section 10.2, “Building POSBranch Images,” on page 176.

2.3.1 Hardware

The Point of Service terminal used for the Branch Server implementation must be a disk-based system and can have an optional added communications adapter.

2.3.2 Operating System

The operating system for the POSBranch Server is built from a SLES 9 source.

2.3.3 High Availability Configuration

You can provide system redundancy and failover for the POSBranch Server in the same way that you would for a standard Branch Server.

For further information, see Section 2.2.6, “High Availability Configuration,” on page 21.

2.3.4 LDAP Objects

Because the POSBranch Server fills the function of both a Branch Server and a Point of Service terminal, you must create a Branch Server object (scBranchServer) and a workstation object (scWorkstation) for the same physical box.

For more information on these objects, see Chapter 5, “The Novell Linux Point of Service LDAP Directory,” on page 55.

2.3.5 LDAP Access

The parameters under which the POSBranch Server accesses the LDAP directory are the same as a standard Branch Server.

For further information, see Section 2.2.3, “LDAP Access,” on page 20.

2.3.6 Administrative Tasks

You can perform the same administrative tasks on the POSBranch Server as a standard Branch Server.

For further information, see Section 2.2.4, “Administrative Tasks,” on page 20.

2.3.7 Services

The POSBranch Server provides the same services as a standard Branch Server.

For further information, see Section 2.2.5, “Services,” on page 21.

2.3.8 TFTP Directory Structure

The TFTP directory structure on the POSBranch Server is the same as a standard Branch Server.

For further information, see Section 2.2.7, “TFTP Server Directory Structure,” on page 22.

Point of Service Terminals

3

Point of Service terminals are the end point in the Novell® Linux Point of Service architecture. They provides customer service functions such as a Point of Sale terminal or bank teller workstation.

This section provides general information on Point of Service terminals.

3.1 Operating System

The Point of Service terminal operating system is a minimal operating environment for specialized Point of Service applications. There are different levels of Point of Service operating environments ranging from an extremely small console-based system, to a feature-rich Java* and browser-capable system, to a system with a customized desktop environment.

The type of operating system that can be installed on a Point of Service terminal is determined by the type of hardware that is available. For example, diskless systems can support only a minimal operating environment such as a console-based system, while Point of Service terminals that have a hard drive can support browser-based or customized desktop environments.

Point of Service operating systems are downloaded to Point of Service terminals in client image files. Each Point of Service terminal gets a client image based on its associated hardware type configuration defined in the scCashRegister object.

NOTE: If a Point of Service does not have an scCashRegister object for its specific hardware type, it uses the configuration for the default scCashRegister object. For more information on defining a default scCashRegister object, see Section 6.4.1, “Adding an scCashRegister Object,” on page 75.

A set of client image templates are provided with Novell Linux Point of Service. Using ImageBuilder, you can customize these templates to provide additional features, software packages, and configuration settings within the image. For a description of the client image templates provided with Novell Linux Point of Service, see Section 4.3, “Point of Service Client Images,” on page 45.

3.1.1 Common Operating System Base

All client images have a common operating system base comprised of the following components:

- Kernel modules for hardware, file system, and network support
- GLIBC and STDLIBC++ libraries
- Bash and base file handling utility
- NTP client for time synchronization

- Multicast TFTP-capable TFTP client (atftp)

These components are created from Novell Linux Desktop (NLD) 9 and SUSE® Linux Enterprise Server (SLES) 9 sources, along with Novell Linux Point of Service 9 software packages.

3.2 Images

Point of Service terminals require a boot image and a client image. These images are stored on the Administration Server under the `/opt/SLES/POS/rsync` directory and are transmitted via the RSYNC server service to Branch Servers, where they can be transmitted to Point of Service terminals at boot time.

For more information, see "Point of Service Images" on page 41.

3.3 Point of Service Terminal LDAP Objects

The configuration parameters for each Point of Service terminal are stored in the central LDAP directory on the Administration Server. Every Point of Service terminal has its own Workstation object (scWorkstation) in the LDAP tree. The Workstation object is automatically created when a Point of Service terminal registers on the Branch Server. posldap2crconfig.pl uses information from the Hardware Reference object (scCashRegister) and Image Reference object (scPosImage) to create the Workstation object. For more information on this process, see Section 3.5.3, "The hwtype.MAC_address File," on page 34.

IMPORTANT: You must create the scCashRegister and scPosImage objects in LDAP before booting the Point of Service terminals. Otherwise, posldap2crconfig.pl cannot create the terminals' associated scWorkstation object. For information on this procedure, see Section 6.4.1, "Adding an scCashRegister Object," on page 75 and Section 6.5.1, "Adding an scPosImage Object," on page 80.

3.4 Hardware

Point of Service terminals are implemented in a variety of hardware forms. The primary difference in Point of Service hardware is whether the terminal has an internal hard drive or other persistent media (such as a flash drive), or whether the terminal is diskless. A system that has a hard disk can be configured to store the image on a disk partition instead of a RAM disk so it can boot from the hard disk if it cannot boot over the network.

3.4.1 Hardware Configuration Files

Point of Service terminal hardware configuration information can be stored in LDAP as scConfigFileTemplate objects, or it can be stored on the Administration Server as a file and distributed over RSYNC. Hardware configuration files that are distributed by the Administration Server over RSYNC must be located in the /opt/SLES/POS/rsync/config/ directory and must have a corresponding scConfigFileSyncTemplate object in the LDAP directory.

NOTE: The hardware configuration files discussed in this section should not be confused with `config.`*`MAC_Address`* Point of Service configuration files. The `config.`*`MAC_Address`* files contain the parameters required to configure a Point of Service terminal during a network PXE or hard disk boot. For more information, see Section 3.5.1, "The config.MAC_address File," on page 28.

The scConfigFileTemplate and scConfigFileSyncTemplate objects are located in LDAP under the scPosImage or scCashRegister objects. In addition to providing Point of Service hardware configuration information, they specify which configuration file a Point of Service terminal should download from the Branch Server at boot time. For information on creating these objects in the LDAP directory, see Section 6.4.2, "Adding an scConfigFileTemplate Object," on page 76 or Section 6.4.1, "Adding an scCashRegister Object," on page 75

The Branch Server initially acquires the hardware configuration information for its local Point of Service terminals in one of two ways:

- `posldap2crconfig.pl` reads the configuration information stored in the scConfigFileTemplate object in LDAP and creates a configuration file in the `/tftpboot/CR/MAC_ Address/` directory on the Branch Server. The hardware configuration file is then distributed to the appropriate Point of Service terminal at boot time.
- `posldap2crconfig.pl` reads where the configuration file is located in the scConfigFileSyncTemplate object. It then triggers an RSYNC call to download the configuration file from the Administration Server. The configuration file is stored in the `/tftpboot/CR/MAC_ Address/` directory on the Branch Server so it can be distributed to the appropriate Point of Service terminal at boot time.

posleases2ldap automatically triggers `posldap2crconfig.pl` the first time a Point of Service terminal registers with the Branch Server. Consequently, you do not have to do anything to initiate these processes except start the posleases2ldap service on the Branch Server after installation.

However, if the terminal's hardware configuration information changes after its initial registration, you must manually run either `posldap2crconfig.pl --dumpall` or `posAdmin --updateconfig` to update the hardware configuration information on the Branch Server. These commands regenerate the hardware configuration and `config.MAC_Address` files for all Point of Service terminals found in LDAP.

For more information on the `posldap2crconfig.pl` script, see Section A.3.5, "posldap2crconfig.pl," on page 206.

For more information on the `posAdmin --updateconfig` command, see Section 6.9, "Updating config.MAC_address and Hardware Configuration Files," on page 85.

3.4.2 Graphical Display Configuration

The graphics controller depends on the model type, so it can be derived from static tables. Some Point of Service terminals can use multihead X configurations. The corresponding XF86Config files are manufacturer-specific and are not provided as part of the Novell Linux Point of Service software package.

3.5 Point of Service Configuration Files

Each Point of Service terminal has its own configuration file that it loads at boot time. This configuration file determines which hardware drivers and images are loaded on the Point of Service terminal. The following sections review the configuration files for a Point of Service terminal booted from the network, from CD, and the configuration file used to register new Point of Service terminals.

- Section 3.5.1, "The config.MAC_address File," on page 28

- Section 3.5.2, “The config.image File,” on page 31
- Section 3.5.3, “The hwtype.MAC_address File,” on page 34

3.5.1 The config.*MAC_address* File

The `config.MAC_address` files contain the parameters required to configure a specific Point of Service terminal during a network PXE or hard disk boot. Each Point of Service terminal has its own `config.MAC_address` file on the Branch Server.

When the Branch Server connects to the Administration Server, it logs into the LDAP directory, accesses the configuration parameters for its registered Point of Service terminals, and stores the information locally as ASCII configuration files (`config.MAC_address`) in the `/tftpboot/CR` directory. At boot time, each Point of Service terminal connects to the Branch Server over TFTP and loads its associated `config.MAC_address` file.

There is no need to manually create the Point of Service configuration files. When a new Point of Service terminal comes online, its configuration file is automatically created from LDAP entries on the Administration Server.

For more information on this process, see Section 3.5.3, “The hwtype.MAC_address File,” on page 34 and Section A.2, “Core Script Process,” on page 201.

To modify a Point of Service configuration file, you must modify the Point of Service terminal’s entries in LDAP and then run the `posAdmin --updateconfig` command.

For more information, see Section 6.9, “Updating config.MAC_address and Hardware Configuration Files,” on page 85.

The format of the config.*MAC_address* file is as follows:

```
IMAGE=device;image;version;srv_ip;bsize;compressed,...,
SYNC=syncfilename;srv_ip;bsize
CONF=source;dest;srv_ip;bsize,...,source;dest;srv_ip;bsize
PART=size;id;Mount,...,size;id;Mount
JOURNAL=ext3
DISK=device
```

Table 3-1 provides a detailed description of each parameter in config.*MAC_address* and its variables.

Table 3-1 *config.MAC_address configuration file parameters*

Parameter	Variable	Description
IMAGE=		Specifies which image (*image*) should be loaded with which version (*version*) and to which storage device (*device*) it should be linked. Multiple image downloads are possible, but the first listed image must be the main client image. If the hard drive is used, a corresponding partitioning must be performed.

Parameter	Variable	Description
	device	The storage device to which the image is linked; for example, /dev/ram1 or /dev/hda2. RAM devices should not be confused with hard disk devices which use a partition table. On a Point of Service terminal, partition hda1 is used for the Linux swap partition and hda2 defines the root file system (/). On the RAM disk device, /dev/ram0 is used for the initial RAM disk and cannot be used as storage device for the client image. It is recommended that you use /dev/ram1 for the RAM disk.
	image	The name of the image to load on the Point of Service terminal.
	version	The version of the image to load on the Point of Service terminal.
	srv_ip	The server IP address for the TFTP download. This variable must always be included in the IMAGE= parameter.
	bsize	The block size for the TFTP download. If the block size is too small according to the maximum number of data packages (32768), linuxrc automatically calculates a new block size for the download. This variable must always be included in the IMAGE= parameter.
	compressed	Specifies a compressed image boot. If the compressed variable is not included, the standard boot process is used. The boot fails if you specify Compressed and the image isn't compressed. It also fails if you don’t specify Compressed and the image is compressed. **IMPORTANT:** The name of the compressed image must contain the suffix .gz and must be compressed with the gzip tool or by using the --gzip option at create time.
SYNC=		Specifies an optional syncfile (*syncfilename*) to download over TFTP. The syncfile indicates the number of seconds to wait before downloading the image.
	syncfilename	The name of the syncfile downloaded over TFTP.
	srv_ip	The server IP address for the TFTP download. This variable must always be included in the SYNC= parameter.
	bsize	The block size for the TFTP download. If the block size is too small according to the maximum number of data packages (32768), linuxrc automatically calculates a new block size for the download. This variable must always be indicated in the SYNC= parameter.
CONF=		Specifies the configuration files to download to the Point of Service terminal. The data is provided in a comma-separated list of *source*:*target* configuration files.
	source	The path to the source configuration file on the TFTP server.
	dest	The directory on the Point of Service terminal where you want to download the source configuration file.

Parameter	Variable	Description
	srv_ip	The server IP address for the TFTP download. This variable must always be included in the CONF= parameter.
	bsize	The block size for the TFTP download. If the block size is too small according to the maximum number of data packages (32768), linuxrc automatically calculates a new block size for the download. This variable must always be included in the CONF= parameter.
PART		Specifies partitioning data. The data is provided in a comma-separated list. The first element of the list defines the swap partition. The second element defines the root partition. Each element must include the size (*size*), the type (*id*), and the mount point (*mount*).
	size	The size of the partition. If you want the partition to take all the space left on a disk, use a lowercase letter x as the size specification.
	id	The partition type: S for swap, L for all others.
	mount	The partition mount point; for example, /home. **IMPORTANT:** The swap partition must not contain a mount point. Use a lowercase letter x instead.
JOURNAL=		Specifies a journaling file system. The value for this parameter must be set to ext3 because the only journaled file system Novell Linux Point of Service supports is ext3. If you have an existing ext2 image, you can change the file system by setting a flag in the scCashRegister or the scWorkstation objects rather than recreate the image. If ext3 is specified in either LDAP object, the Point of Service terminal extends the file system to ext3 when the image is deployed. The JOURNAL= parameter is evaluated only if the DISK= parameter is set.
DISK=		Defines the device through which the hard disk can be addressed; for example /dev/hda. This parameter is used only with PART.
RELOAD_IMAGE=		If set to a non-empty string, this parameter forces the configured image to be loaded from the server even if the image on the disk is up-to-date. The `posldap2crconfig.pl` script overwrites this optional feature of the Point of Service configuration file. This parameter is used mainly for debugging purposes. It is pertinent only on disk-based systems.

Parameter	Variable	Description
RELOAD_CONFIG=		If set to an non-empty string, this parameter forces the `config.MAC_address` file to be loaded from the server. If you run `posldap2crconfig.pl --dumpall` to regenerate the `config.MAC_address` file, it overwrites this optional parameter. This parameter is used mainly for debugging purposes. It is pertinent only on disk-based systems.

Here is a sample `config.MAC_address` file:

```
IMAGE=/dev/hda2;image/browser;2.0.21;192.168.1.1;4096;compressed
CONF=/CR/00:30:05:1D:75:D2/ntp.conf;/etc/ntp.conf;192.168.1.1;1024,
     /CR/00:30:05:1D:75:D2/XF86Config;/etc/X11/XF86Config;
     192.168.1.1;1024
PART=200;S;x,300;L;/,500;L;/opt,x;L;/home
DISK=/dev/hda
```

3.5.2 The config.image File

The `config.image` file is similar in function to `config.MAC_address`. It contains the parameters required to configure a specific Point of Service terminal during a CDBoot; that is, it indicates which client image the CDboot boot image should load and how to do it. The CD-based Point of Service configuration file must be named `config.image` and it must be located in the CD setup directory.

The format of the `config.image` file is as follows:

```
IMAGE=device;image;version;compressed
CONF=source;dest,...,source;dest
PART=size;id;Mount,...,size;id;Mount
JOURNAL=ext3
DISK=device
FEATURE=The contents of the --feature option
EXTEND=The contents of the --extend option
PARAMS=Additional options
```

Table 3-2 provides a detailed description of each parameter in config.image and its variables.

Table 3-2 *config.image configuration file parameters*

Parameter	Variable	Description
IMAGE=		Specifies the client image (*image*) and version (*version*) that will be loaded on the Point of Service terminal. When you generate the CDBoot image, ImageBuilder uses this information to generate the client image with the CDBoot image.

Parameter	Variable	Description
	device	The storage device to which the image is linked, for example, /dev/ram1 or /dev/hda2. RAM devices should not be confused with hard disk devices which use a partition table. On a Point of Service terminal, partition hda1 is used for the swap partition and hda2 defines the root file system (/). On the RAM disk device, /dev/ram0 is used for the initial RAM disk and cannot be used as storage device for the client image. It is recommended that you use /dev/ram1 for the RAM disk.
	image	The name of the client image to load on the Point of Service terminal.
	version	The version of the client image to load on the Point of Service terminal.
	compressed	Specifies a compressed image boot. If the compressed variable is not included, the standard boot process is used. The the boot fails if you specify Compressed and the image isn't compressed. It also fails if you don' specify Compressed and the image is compressed. **IMPORTANT:** The name of the compressed image must contain the suffix `.gz` and must be compressed with the gzip tool or by using the `--gzip` option at create time.
CONF=		Specifies the configuration files to download to the Point of Service terminal. The data is provided in a comma-separated list of *source*:*target* configuration files.
	source	The path to the source configuration file within the directory.
	dest	An absolute path below the client image where the configuration file is saved.
PART=		Specifies partitioning data. The data is provided in a comma-separated list. The first element of the list defines the swap partition. The second element defines the root partition. Each element must include the size (*size*), the type (*id*), and the mount point (*mount*).
	size	The size of the partition. If you want a partition to take all the space left on a disk, use a lowercase letter x as the size specification.
	id	The partition type: S for swap, L for all others.
	mount	The partition mount point; for example, /home. **IMPORTANT:** The swap partition must not contain a mount point. Use a lowercase letter x instead.

Parameter	Variable	Description
JOURNAL=		Specifies a journaling file system. The value for this parameter must be set to ext3 because the only journaled file system Novell Linux Point of Service supports is ext3. If you have an existing ext2 image, you can change the file system by setting a flag in the scCashRegister or the scWorkstation objects rather than recreate the image. If ext3 is specified in either LDAP object, the Point of Service terminal extends the file system to ext3 when the image is deployed. The JOURNAL= parameter is evaluated only if the DISK= parameter is set.
DISK=		Defines the device through which the hard disk can be addressed; for example /dev/hda. This parameter is used only with PART=.
FEATURE=		This is the value of the --feature command used when building the client image. For information on this command, refer to Section 8.1, “scr Commands,” on page 97 or Section 9.1, “xscr Commands,” on page 121. This optional parameter is only pertinent while ImageBuilder creates the client image.
EXTEND=		This is the value of the --extend option used to extend an image with an additional RPM package. For information, refer to Section 8.1, “scr Commands,” on page 97. This optional parameter is only pertinent while ImageBuilder creates the client image. **IMPORTANT:** This parameter is only relevant to standard client images generated with scr. The xscr ImageBuilder tool uses the `ImageSpecification.xml` document to extend client images.
PARAMS=		Specifies options that are used for special actions. This parameter is only pertinent while ImageBuilder creates the client image. This parameter can be used with the --gzip option to compress the image. The CDboot linuxrc recognizes a compressed image referring to the suffix .gz. A compressed CD image is uncompressed on the fly while the image is installed. For information on this command, refer to Section 8.1, “scr Commands,” on page 97 or Section 9.1, “xscr Commands,” on page 121. For POSBranch images, it is recommend that you add the following line to the `config.image` file: `PARAMS=--keep-rpm` This allows you to use the YaST2 interface to configure POSBranch Servers. However, it adds approximately 30 MB to the size of the image. If the size of the image is an issue, you can leave the RPMs out; however, you will not have YaST2 functionality.

For more information on creating a CDBoot image, see Section 10.1, “Building a CDBoot Image,” on page 171.

3.5.3 The hwtype.*MAC_address* File

When a Point of Service terminal comes online for the first time, it does not have a `config.MAC_address` file on the Branch Server. To create this file for the terminal, the system must first register the Point of Service terminal in LDAP. This is done through the Point of Service control file, `hwtype.MAC_address`. The Point of Service control file contains the information required to create the terminal's workstation object (scWorkstation) in LDAP and determine which image and configuration settings should be included in the terminal's configuration file (`config.MAC_address`).

The Point of Service control file is formatted as follows:

```
HWTYPE=hardware type
HWBIOS=bios version
CRNAME=alias name
```

NOTE: If no alias name is set, the default name of "undefined" is used.

The process used to create the `config.MAC_address` file from the `hwtype.MAC_address` file is as follows:

1. During the Point of Service boot process, the hardware type, BIOS version, and Point of Service alias name are detected.

   NOTE: The Point of Service hardware manufacturer provides a program for this function.

   ---

2. Using this information, the `posleases2ldap.pl` script creates the control file, `hwtype.MAC_address`.

   For more information, see Section A.3.8, "posleases2ldap.pl," on page 208.

3. The linuxrc program uploads `hwtype.MAC_address` to the Branch Server's upload directory, `/tftpboot/upload`.

   NOTE: The control file is uploaded to the TFTP server only when no configuration file (`config.MAC Address`) exists.

   ---

4. The hardware type identified in the `hwtype.MAC_address` file is compared to the scCashRegister objects in the LDAP directory.

   If a match is found, the information in scCashRegister and its associated objects is used to create the Point of Service terminal's scWorkstation object in LDAP and its `config.MAC_address` file in the Branch Server's `/tftpboot/CR` directory. After the `config.MAC_address` file is created, the `hwtype.MAC_address` file is deleted.

   If the hwtype is unknown, the information in the default scCashRegister object is used to create the Point of Service terminal's scWorkstation object and `config.MAC_address` file.

   IMPORTANT: This safety net feature works only if you have configured designated a default scCashRegister object in the LDAP directory. For information on defining a default scCashRegister object, see Section 6.4.1, "Adding an scCashRegister Object," on page 75.

For a detailed review of the core scripts involved in this process, see Section A.2, "Core Script Process," on page 201.

3.6 Booting the Point of Service Terminal

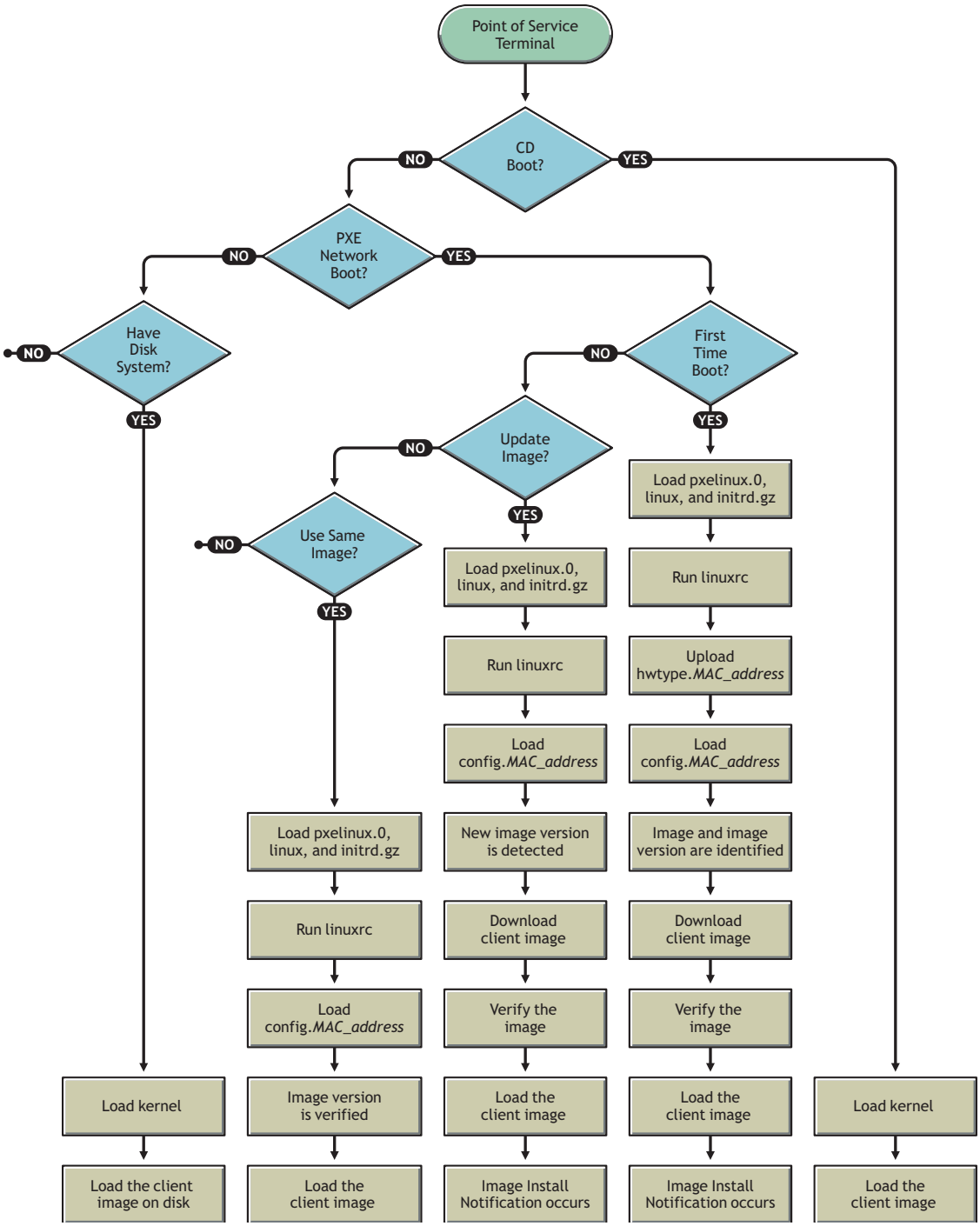
IMPORTANT: You must create scCashRegister and its associated objects before you can boot the Point of Service terminals. For more information, see Section 6.4.1, “Adding an scCashRegister Object,” on page 75.

Typically, when you boot a Point of Service terminal, it will first try to boot from CD. If a CD is not available, the terminal attempts a network PXE boot. If the network is not available, it then boots from the hard drive. (You can override this order with the BIOS settings.)

The first time you boot the Point of Service terminals, the posleases2ldap daemon automatically triggers `posldap2crconfig.pl` which then creates a Workstation object (scWorkstation) and hardware configuration files for the Point of Service terminals that register on the Branch Server. For more information on this process, see Section 3.5.3, “The hwtype.MAC_address File,” on page 34.

Figure 3-1 provides a simplified overview of the Point of Service boot process for a network PXE boot, a hard disk boot, and a CD boot.

Figure 3-1 *Point of Service terminal boot process*



Detailed information about each boot process is provided in the following sections:

3.6.1 Network PXE Boot

To boot Point of Service terminals from the network, the following conditions must be met:

- The terminal must have a network connection to the Branch Server.
- The TFTP service must be properly configured and running on the Branch Server.

  For more information on Branch Server configuration, see "Setting Up a Branch Server" in the *Novell Linux Point of Service 9 Installation Guide*.
- The terminal must have an associated scCashRegister object in the LDAP directory.

  For more information, see Section 6.4, "Defining Point of Service Terminal Objects," on page 74.
- The Point of Service boot images must be located in the `/tftpboot/boot/` directory on the Branch Server and the client images must be located in the `/tftpboot/image/` directory.

  For more information, see Section 8.5, "Distributing Images," on page 117.
- The Point of Service client images must have an associated scPosImage object in the LDAP directory and the object's scPosImageVersion attribute must be set to Active.

  For more information, see Section 6.5, "Managing Image Objects," on page 79.

If these conditions are met, the Point of Service terminal can successfully boot from the network.

The following is a detailed description of what takes place when a Point of Service terminal boots from the network:

1. The Point of Service terminal makes a DHCP request.
2. The Point of Service terminal downloads `pxelinux.0`.

   The `pxelinux.0` image is the first bootstrap image used to PXE boot the Point of Service terminals.
3. The Point of Service terminal downloads the `linux` file.

   The `linux` file is actually the `DiskNetboot-`*`version-date`*`.kernel.`*`version`*`-SLRS` image, which provides the Linux kernel used to PXE boot the Point of Service terminals.
4. Using PXE network boot or boot manager (GRUB), the Point of Service terminal boots the initrd (`initrd.gz`) that it receives from the Branch Server.

   If no PXE boot is possible, the Point of Service terminal tries to boot via hard disk, if accessible.
5. The linuxrc script begins.
6. The file systems required to receive system data are mounted; for example, the proc file system.
7. The Point of Service hardware type (*hwtype*) is detected.

   The Point of Service hardware manufacturer provides a program to do this. The first time the Point of Service terminal boots, this information is used to register the Point of Service terminal and create the terminal's `config.`*`MAC_address`* file. This information is also used to determine which configuration files the terminal should use.
8. The Point of Service BIOS version (*hwbios*) is detected.

   The Point of Service hardware manufacturer provides a program to do this.

9. Network support is activated. The required kernel module is determined from a static table by selecting the entry corresponding to the hardware type. If no known hardware type is detected, a default list of modules is used and types are tried one after the other.
10. The module is loaded using modprobe. Any dependencies to other modules are cleared at that time.
11. The network interface is set up via DHCP.
12. After the interface has been established, the DHCP variables are exported into the `/var/lib/dhcpcd/dhcpcd-eth0.info` file and the contents of DOMAIN and DNS are used to generate an `/etc/resolv.conf` file.
13. The TFTP server address is acquired.

    During this step, a check is first made to determine whether the hostname tftp.\$DOMAIN can be resolved. If not, the DHCP server is used as the TFTP server.
14. The Point of Service configuration file, `config.MAC_address`, is loaded from the Branch Server's `/tftpboot/CR` directory over TFTP.

    If this is the Point of Service terminal's first time booting, its `config.MAC_address` file does not yet exist. The Point of Service terminal must first register on the system.

    A new Point of Service terminal registers as follows:

    a. An optional alias name can be set for the new Point of Service terminal. During the creation of one of the boot images, you can enable the system alias setting using the POSSetAlias feature module. By default, there is no question for the system alias name.

    b. A Point of Service control file (`hwtype.MAC_address`) is uploaded to the TFTP server's upload directory: `/tftpboot/upload`.

       The `hwtype.MAC_address` file indicates the Point of Service hardware type, the BIOS version, and the Point of Service alias name. The system uses this information to create the terminal's `config.MAC_address` file. For more information on this process, see Section 3.5.3, "The hwtype.MAC_address File," on page 34.

    c. After the upload, the Point of Service terminal renews the DHCP lease file (`dhcpcd -n`).

    d. The Point of Service terminal attempts to load its new `config.MAC_address` file from the TFTP server.

    e. If the `config.MAC_address` file is not yet available, the Point of Service terminal waits 60 seconds before repeating steps c and d.
15. When the `config.MAC_address` file loads, the system begins an analysis of its contents.

    For more information about the content and file format of the `config.MAC_address` file, refer to Section 3.5.1, "The config.MAC_address File," on page 28.
16. The PART line in the `config.MAC_address` file is analyzed.

    If there is a PART line in the configuration file, a check is made using the image version to see whether any local system needs to be updated.

    - If no system update is required, no image download occurs and the Point of Service terminal boots from the hard drive.
    - If a system update is required, the Point of Service terminal's hard disk is partitioned according to the parameters specified in the PART line.
17. The SYNC line in the Point of Service configuration file is evaluated.

    If there is a SYNC line, the indicated file is downloaded over TFTP.

The only value the file contains is the number of seconds to wait (sleep) before the multicast download of the client image starts. If the file is not present, the boot process immediately proceeds.

18. Indicated images are downloaded with multicast TFTP.
19. If the image is compressed, it is copied then decompressed.
20. The image checksums are verified.

    If they do not match, the images are re-downloaded.
21. The CONF line in the Point of Service configuration file is evaluated.

    All the indicated files are loaded from the TFTP server and stored in a `/config/` path.
22. All the user-land processes based on the boot image (`dhcpcd -k`) are terminated.
23. The client image is mounted.
24. The configuration files stored in the `/config/` path are copied to the mounted client image.
25. If this is a new image, Image Install Notification occurs.
    a. The *`bootversion.MAC_Address`* file is created in `/tftpboot/upload`.
    b. posleases2ldap transfers the information to the scNotifiedimage attribute in the scWorkstation object in LDAP.
26. The system switches to the mounted client image.
27. The root file system is converted to the client image using pivot_root.

    All the required configuration files are now present because they had been stored in the client image or have been downloaded via TFTP.

    The file systems that are mounted read-only can be stored in cramfs-compressed RAM file systems to save Point of Service RAM resources.
28. The boot image is unmounted using an exec umount call.
29. When linuxrc or the exec call terminates, the kernel initiates the init process, which starts processing the boot scripts as specified in `/etc/inittab`.

3.6.2 CDBoot

If you are unable to electronically distribute Point of Service images over your network, you must manually distribute the images uses CDBoot images. For more information on creating a CDBoot image, see “Building a CDBoot Image”Section 10.1, “Building a CDBoot Image,” on page 171.

The behavior of Point of Service terminals booting from CD is similar to Point of Service terminals that receive the first and second stage boot images over the LAN from a Branch Server. The following is a general description of what takes place when a Point of Service terminal boots from CD:

1. The client image (for example, the Browser image) is installed to a RAM or hard disk drive on the Point of Service terminal.

   The partition information resides in the `config.image` file located on the CD.
2. The installed client image is booted from the RAM or hard disk drive on the Point of Service.

Depending on the client image that resides on the boot CD (Minimal, Java, Browser, or Desktop), you should note the following restrictions:

- The Java and Browser images should only be used for diskful Point of Service systems. Otherwise, the Point of Service system must be upgraded with enough RAM to hold the client image.
- There must be enough available RAM on diskless Point of Service terminals to load the first and second stage boot images. Otherwise the terminal returns a kernel panic error.

NOTE: Keep in mind that onboard VGA reduces the Point of Service terminal's available RAM.

Point of Service Images

4

Point of Service terminals boot two images—a boot image and a client image. To deploy Novell® Linux Point of Service, you must provide boot and client images for your Point of Service terminals. Novell Linux Point of Service provides templates for boot and client images that can be customized and used to generate new images using the ImageBuilder utilities.

NOTE: When you select the NLPOS Image Server during the Administration Server installation, the ImageBuilder utilities (scr and xscr) are installed on the Administration Server with all the files and directories required to create Point of Service images. The Image Description Trees and their associated files are written to `/opt/SLES/POS/system/image_name-version/`. The ImageBuilder utilities can also be installed on a dedicated Image Building server. For further information on installing the image building tools, see "Setting Up the Administration Server" or "Setting Up a Dedicated Image Building Server" in the *Novell Linux Point of Service 9 Installation Guide*.

The following sections provide information about images and the image templates provided with Novell Linux Point of Service:

For information on building Point of Service images, see Chapter 8, "Building Images with the scr ImageBuilder Tool," on page 97 or Chapter 9, "Building Images with the xscr ImageBuilder Tool," on page 121.

4.1 Image Building Overview

The following packages provide general product information and the Novell Linux Point of Service image building tools:

- **POS_Image** contains the README.Packages file, which describes the package structure of the client image files.
- **POS_Image-Builder** provides the standard (scr) and XML (xscr) ImageBuilder utilities.

ImageBuilder is a Perl-based tool that lets you create customized images. The necessary image building components are installed when you select the NLPOS Admin Server Image Building System in the Novell Linux Point of Server Administration Server installation.

ImageBuilder comes in two versions: scr and xscr.

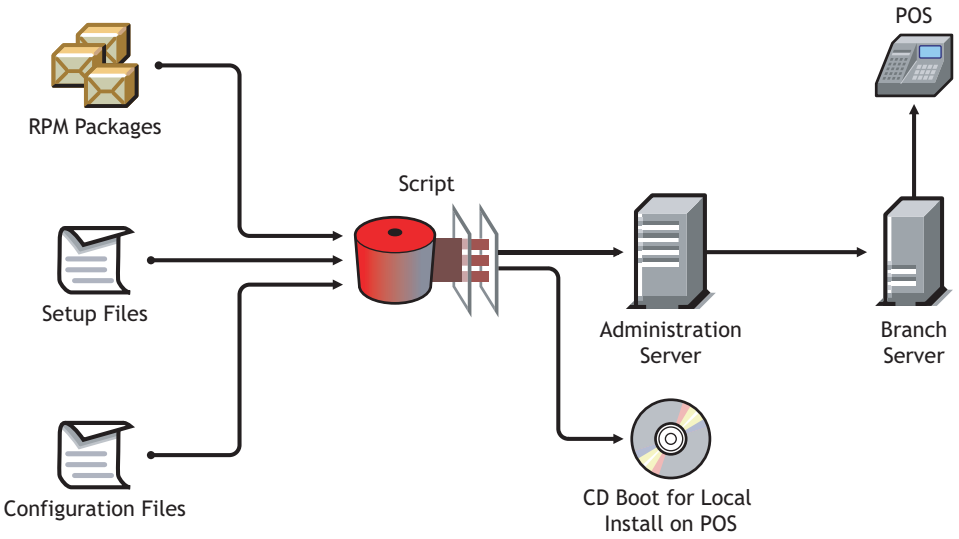
- scr builds images using the Image Description Tree and the `AdminServer.conf` file. The Image Description Tree and `AdminServer.conf` file contain files and directories that

define the structure, scripts, configuration files, and other components required to build client images for Point of Service systems. scr generates client images with Novell Linux Desktop (NLD). For more information, see Chapter 8, “Building Images with the scr ImageBuilder Tool,” on page 97.

- xscr builds images using the Image Description Tree, an Image Specification Document (`ImageSpecification.xml`), and a Distribution Source Document (`Distribution.xml`). The Image Specification and Distribution Source Documents contain XML elements that define the structure, configuration files, and other components required to build client images for Point of Service systems. xscr can generate client images with either NLD or SUSE® Linux Enterprise Server (SLES). For more information, see Chapter 9, “Building Images with the xscr ImageBuilder Tool,” on page 121.

When it builds an image, ImageBuilder compiles all the information required to run a Point of Service terminal—the operating system, application files, configuration settings, drivers, and so forth—into a single image file. This file can then be electronically distributed to Point of Service terminals over the network, or an ISO version of the image file can be burned to a CD for manual distribution, as shown in Figure 4-1.

Figure 4-1 *Client image creation and distribution process*



To build the image file, ImageBuilder requires the following information:

- **RPM Packages** are the operating system and application files that are installed on the Point of Service terminals, including packages from SLES, Novell Linux Point of Service, NLD, Service Packs and add-on packages.
- **Setup Files** include the system setup files, Branch Server setup files, and setup files for custom extensions to the image.
- **Configuration Settings** define the following parameters:
  - Image Size
  - Image Name
  - Time Zone Information
  - Driver Information (network and hardware)

- Key Table Map
- Image Type (ext3, ext2, or reiser).
- Machine Type (the machine hardware, necessary driver modules, and additional scripts)

After the image is created, it can be distributed as a CDBoot image for local installations on Point of Service terminals, or it can be distributed over the network, in which case the images are stored on the Administration Server under the `/opt/SLES/POS/rsync/` directory and are transmitted over the RSYNC server service to Branch Servers where, in turn, they can be transmitted to Point of Service terminals at boot time.

4.2 Point of Service Boot Images

Novell Linux Point of Service provides the boot image files required to boot Point of Service terminals from the network or CD. The boot images are summarized in Table 4-1.

Table 4-1 *Point of Service boot images*

Image	Description
DiskNetboot Image Description Tree: /opt/SLES/POS/system/disknetboot-*version*/ Binary Files: initrd-disknetboot-*version-date*.gz initrd-disknetboot-*version-date*.kernel.*kernel_ version* Image Specification Document: /opt/SLES/POS/system/templates/support/disknetboot.xml	DiskNetboot includes all the files and directories (including partitioning and boot loader installation) required to boot disk-based and diskless Point of Service terminals from the network. Novell Linux Point of Service includes binary versions of the first and second stage boot images used to PXE boot Point of Service terminals. **IMPORTANT:** The boot images must be copied to the Administration Server's `/opt/SLES/POS/rsync/boot` directory as `initrd.gz` and `linux` before Point of Service terminals can use the images to boot. For more information on this procedure, see "Copying Boot Images to the Administration Server's RSYNC Directory" on page 118.
CDBoot Image Description Tree: /opt/SLES/POS/system/cdboot-*version*/ Binary File: /opt/SLES/POS/image/cdboot-*version-date*.gz Image Specification Document: /opt/SLES/POS/system/templates/support/cdboot.xml	CDBoot includes all the files and directories required to boot diskless and preinstalled disk-based systems from CD. To boot diskless systems, the image loads RAM disks from a fixed CD image file. Novell Linux Point of Service includes a binary version of the CDBoot image that is used to boot Point of Service terminals from a CD. This image must be combined with a client image and the `config.image` configuration file to create CD that can be used to boot Point of Service terminals. For information on creating CDBoot images, see Section 10.1, "Building a CDBoot Image," on page 171.

The following sections provide more information on each type of boot image.

- Section 4.2.1, "DiskNetboot," on page 44
- Section 4.2.2, "CDBoot," on page 44

4.2.1 DiskNetboot

Point of Service terminals that boot from the network or hard disk require a first and second stage boot image. The first stage boot image, `initrd.gz`, is the bootstrap image used to PXE boot Point of Service terminals. The second stage boot image, `linux`, provides the Linux kernel. These images are loaded when the Point of Service terminal boots. The system then becomes network-capable and loads one of the client images over TFTP. For more information on the boot process, see Section 3.6.1, "Network PXE Boot," on page 37.

The initrd and kernel images loaded by each Point of Service terminal are determined by the Distribution Container object (scDistributionContainer) in which its associated client image object (scPosImage) is located. The scInitrdName and scKernelName attributes in the scDistributionContainer object define the initrd and kernel images for the container. All Point of Service terminals that load client images located within the Distribution Container use the designated boot images at load time.

Novell Linux Point of Service provides default versions of the first and second stage boot images. The `initrd.gz` image is provided as `/opt/SLES/POS/image/initrd-disknetboot-`*`version-date`*`.gz`. The `linux` image is provided as `/opt/SLES/POS/image/initrd-disknetboot-`*`version-date`*`.kernel.`*`kernel_version`*. You can use these default images to boot your Point of Service terminals or you can create your own boot images using the DiskNetboot Image Description Tree.

NOTE: To customize the boot images, you would clone the DiskNetboot Image Description Tree, make any required modifications, then build the images. When you use the `--build` command with the DiskNetboot Image Description Tree, ImageBuilder generates the first and second stage boot images. For more information on this procedure, see Section 8.4, "Building Images with scr," on page 109 or Section 9.4, "Building Images with xscr," on page 144.

Whether you use the boot images provided with Novell Linux Point of Service or create your own, you must copy the images to the `/opt/SLES/POS/rsync/boot/` directory on the Administration Server before running `posSynchImages.pl` on the Branch Server. The first stage boot image (`initrd-disknetboot-`*`version-date`*`.gz`) must be copied to the RSYNC directory as `initrd.gz`. The second stage boot image (`initrd-disknetboot-`*`version-date`*`.kernel.`*`kernel_version`*) must be copied to the RSYNC directory as `linux`. For specific instructions on this procedure, see "Copying Boot Images to the Administration Server's RSYNC Directory" on page 118.

The default boot images provided with Novell Linux Point of Service are ext2 images. If journaling is needed on disk-based systems, the scDiskJournal attribute of either the scCashRegister or the scWorkstation must be set to TRUE (case is important). This setting directs the Point of Service terminal to extend the file system to ext3 when the images are deployed.

4.2.2 CDBoot

In environments where no network infrastructure is available to boot Point of Service systems over the LAN, you can use boot CDs. Boot CDs are also required to deploy POSBranch Servers.

Point of Service terminals that boot from CD require a minimal Linux system image (CDBoot), a Linux system client image (Minimal, Java, Browser, or Desktop), and a `config.image` configuration file that controls whether the client image is written into a RAM disk or if it must be

placed on the hard disk of the booting node. For more information on the CDBoot process, see Section 3.6.2, “CDBoot,” on page 39

The CDBoot components must be packaged in an ISO 9660-compliant CD image and burned to CD. For detailed information on generating the CDBoot files and creating the ISO image, see Section 10.1, “Building a CDBoot Image,” on page 171.

4.3 Point of Service Client Images

Point of Service client images provide the operating system and basic software packages for Point of Service terminals. All client images (with the exception of POSBranch images) are based on the Novell Linux Desktop (NLD). This operating system provides the following components as a baseline for client images:

- Kernel modules for hardware, file system, and network support
- GLIBC and STDLIBC++ libraries
- Bash and base file handling utility
- NTP client for time synchronization
- Multicast TFTP-capable TFTP client (atftp)

Novell Linux Point of Service provides templates for the following client images:

- Section 4.3.1, “Minimal Client Image,” on page 45
- Section 4.3.2, “Java Client Image,” on page 46
- Section 4.3.3, “Browser Client Image,” on page 47
- Section 4.3.4, “Desktop Client Image,” on page 48

4.3.1 Minimal Client Image

The Minimal client image includes the runtime environment for native code (that is C and C++) and the ncurses library for user interface support. It supports only console-based applications.

The maximum size of the Minimal image is 60 MB compressed. 64 MB of RAM is required to boot the image.

Image Description Tree

The Image Description Tree for the Minimal client image is:

`/opt/SLES/POS/system/minimal-`*`version`*`/`

Image Specification Documents

Base Template. The Image Specification Document for the base template is `/opt/SLES/POS/system/templates/support/minimal-base.xml`. This file specifies the drivers and RPMs required to create the Minimal image. It is included as a child document in the `ImageSpecification.xml` document at the root of the Minimal Image Description Tree.

NLD Template. The Image Specification Document for a Minimal client image that includes the NLD RPMs is `/opt/SLES/POS/system/templates/support/minimal.xml`. This file is included as a child document for the Minimal Image Specification Document.

When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (`--dist nld|sles`). If you define the image distribution as NLD, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document.

NOTE: NLD is the default distribution type. Therefore, if you do not specify the image distribution when cloning the Image Description Tree, xscr defaults to NLD.

In general, most Point of Service images are created using the NLD distribution.

Novell Linux Point of Service includes a binary version of the Minimal NLD image that can be used for system testing. The binary file is `/opt/SLES/POS/image/minimal-`*`version-date`*`.gz`.

SLES Template. The Image Specification Document for a Minimal client image that includes the SLES RPMs is `/opt/SLES/POS/system/templates/support/minimal-sles.xml`. This file is included as a child document for the Minimal Image Specification Document.

When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (`--dist nld|sles`). If you define the image distribution as SLES, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document.

The only Point of Service images that require the SLES distribution are POSBranch images. For more information on POSBranch images, see Section 4.5, “POSBranch Images,” on page 51.

4.3.2 Java Client Image

The Java* client image contains everything in the Minimal client image and adds the X11 server and configuration. It supports console-based C/C++ applications, Java programs in a Java2 runtime environment, and X11 applications.

The maximum size of the Java image is 200 MB compressed. 128 MB of RAM is required to boot the image.

Image Description Tree

The Image Description Tree for the Java client image is:

`/opt/SLES/POS/system/java-`*`version`*`/`

Image Specification Documents

Base Template. The Image Specification Document for the base template is `/opt/SLES/POS/system/templates/support/java-base.xml`. This file specifies the drivers and RPMs required to create the Java image. It is included as a child document in the `ImageSpecification.xml` document at the root of the Java Image Description Tree.

NLD Template. The Image Specification Document for a Java client image that includes the NLD RPMs is `/opt/SLES/POS/system/templates/support/java.xml`. This file is included as a child document for the Java Image Specification Document.

When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (`--dist nld|sles`). If you define the image distribution as NLD, xscr adds this

child document to the IncludeSpecificationList element in the parent Image Specification Document.

NOTE: NLD is the default distribution type. Therefore, if you do not specify the image distribution when cloning the Image Description Tree, xscr defaults to NLD.

In general, most Point of Service images are created using the NLD distribution.

SLES Template. TheImage Specification Document for a Java client image that includes the SLES RPMs is `/opt/SLES/POS/system/templates/support/java-sles.xml`. This file is included as a child document for the Java Image Specification Document.

When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (`--dist nld|sles`). If you define the image distribution as SLES, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document.

The only Point of Service images that require the SLES distribution are POSBranch images. For more information on POSBranch images, see Section 4.5, “POSBranch Images,” on page 51.

4.3.3 Browser Client Image

The Browser client image includes all elements of the Minimal and Java images, but is also equipped with the Mozilla Web browser. The image can be extended to include other Web browsers. The Browser image supports console-based C/C++ applications, Java programs in a Java2 runtime environment, and X11 applications.

The maximum size of the Browser image is 250 MB compressed.

This image is intended for disk-based systems. To deploy the image on a disk-based system, the terminal must have 250 MB of available hard disk space and 256 MB of RAM. However, if the terminal has enough RAM, you can deploy the image in memory. To deploy the default Browser image on a diskless system, the terminal must have at least 1 GB of RAM.

Image Description Tree

The Image Description Tree for the Browser client image is:

`/opt/SLES/POS/system/browser-`*`version`*`/`

Image Specification Documents

Base Template. The Image Specification Document for the base Browser image is `/opt/SLES/POS/system/templates/support/browser-base.xml`. This file specifies the drivers and RPMs required to create the Browser image. It is included as a child document in the `ImageSpecification.xml` document at the root of the Browser Image Description Tree.

NLD Template. The Image Specification Document for a Browser client image that includes the NLD RPMs is `/opt/SLES/POS/system/templates/support/browser.xml`. This file is included as a child document for the Browser Image Specification Document.

When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (`--dist nld|sles`). If you define the image distribution as NLD, xscr adds this

child document to the IncludeSpecificationList element in the parent Image Specification Document.

NOTE: NLD is the default distribution type. Therefore, if you do not specify the image distribution when cloning the Image Description Tree, xscr defaults to NLD.

In general, most Point of Service images are created using the NLD distribution.

SLES Template. The Image Specification Document for a Browser client image that includes the SLES RPMs is `/opt/SLES/POS/system/templates/support/browser-sles.xml`. This file is included as a child document for the Browser Image Specification Document.

When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (`--dist nld|sles`). If you define the image distribution as SLES, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document.

The only Point of Service images that require the SLES distribution are POSBranch images. For more information on POSBranch, see Section 4.5, “POSBranch Images,” on page 51.

4.3.4 Desktop Client Image

The Desktop client image includes one Web browser (Mozilla) with plug-ins and a full graphical user interface (KDE 3.2 or GNOME 2.6). It supports console-based C/C++ applications, Java programs in a Java2 runtime environment, and X11 applications.

This image is intended for disk-based systems; however, if the terminal has enough RAM, you can deploy the image in memory. To deploy the default Desktop image on diskless systems, the terminal must have at least 1 GB of RAM.

Image Description Tree

The Image Description Tree for the Desktop client image is:

`/opt/SLES/POS/system/desktop-`*`version`*`/`

Image Specification Documents

Base Template. The Image Specification Document for the base Desktop image template is `/opt/SLES/POS/system/templates/support/desktop- base.xml`. This file specifies the drivers and RPMs required to create the Desktop image. It is included as a child document in the `ImageSpecification.xml` document at the root of the Desktop Image Description Tree.

NLD Template. The Image Specification Document for a Desktop client image that includes the NLD RPMs is `/opt/SLES/POS/system/templates/support/desktop.xml`. This file is included as a child document for the Desktop Image Specification Document.

When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (`--dist nld|sles`). If you define the image distribution as NLD, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document.

NOTE: NLD is the default distribution type. Therefore, if you do not specify the image distribution when cloning the Image Description Tree, xscr defaults to NLD.

In general, most Point of Service images are created using the NLD distribution.

SLES Template. The Image Specification Document for a Desktop client image that includes the SLES RPMs is `/opt/SLES/POS/system/templates/support/desktop-sles.xml`. This file is included as a child document for the Desktop Image Specification Document.

When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (`--dist nld|sles`). If you define the image distribution as SLES, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document.

The only Point of Service images that require the SLES distribution are POSBranch images. For more information on POSBranch, see Section 4.5, "POSBranch Images," on page 51.

4.4 Client Image Add-On Features

Novell Linux Point of Service includes several add-on features that can be added to client images generated with xscr. Table 4-2 describes the features that can be added to client images. For information on extending a client image to include these features, see "Adding Features to Client Images" on page 146.

IMPORTANT: Some of the add-on features have dependencies. The dependencies are noted in the table; however, you can also check the RequiredList element in the Image Description Document to verify dependencies. If the image does not have a RequiredList element, the add-on feature can be added to any client image. For more information, see "RequiredList" on page 134.

Table 4-2 *Client image add-on features*

Feature	Image Specification Document	Description
admind	/opt/SLES/POS/system/templates/addons/admind.xml	Adds the admind utility to client images. This utility allows simple commands to be executed on Point of Service terminals from a remote location. For more information, see Chapter 11, "Remotely Managing Point of Service Terminals with admind and adminc," on page 187. This feature can be added to any NLD-based client image.
Advanced Linux Sound Library	/opt/SLES/POS/system/templates/addons/alsa.xml	Adds the Advanced Linux Sound Library (ALSA) to client images. ALSA provides audio and MIDI functionality for Point of Service terminals. This feature can be added to any client image.
Debug	/opt/SLES/POS/system/templates/addons/debug.xml	Adds debugging tools to client images for troubleshooting purposes. This feature can be added to any client image.

Feature	Image Specification Document	Description
EvTouch	/opt/SLES/POS/system/templates/addons/evtouch.xml	Adds the driver for evtouch screens in ncurses mode. **NOTE:** This driver does not support evtouch screens in X11 mode. This feature can be added only to the Java, Browser, or Desktop images.
Firefox	/opt/SLES/POS/system/templates/addons/firefox.xml	Adds the Firefox browser to client images. This feature can be added only to the NLD-based Browser or Desktop images.
GNOME 2.6 for NLD	/opt/SLES/POS/system/templates/addons/gnome.xml	Adds the GNOME desktop to NLD-based client images. This feature can be added only to the NLD Desktop image.
GNOME 2.6 for SLES	/opt/SLES/POS/system/templates/addons/gnome-sles.xml	Adds the GNOME desktop to SLES-based images used for POSBranch. This feature can be added only to the SLES Desktop image.
IBM Java	/opt/SLES/POS/system/templates/addons/ibmjava.xml	Adds the current IBM Java Runtime Environment (JRE) to NLD-based client images. This feature can be added to the Java, Browser, or Desktop images.
KDE 3.2 for NLD	/opt/SLES/POS/system/templates/addons/kde.xml	Adds the KDE desktop to NLD-based client images. This feature can be added only to the NLD Desktop image.
KDE 3.2 for SLES	/opt/SLES/POS/system/templates/addons/kde-sles.xml	Adds the KDE desktop to SLES-based images used for POSBranch. This feature can be added only to the SLES Desktop image.
Mozilla	/opt/SLES/POS/system/templates/addons/mozilla.xml	Adds the Mozilla browser to client images. This feature can be added to the Browser or Desktop images.
Samba 3 Client	/opt/SLES/POS/system/templates/addons/samba.xml	Provides Common Internet File System (CIFS) file access for Windows and Linux clients. **NOTE:** The Samba 3 server is included with Novell Linux Point of Service. This feature can be added to any client image.

Feature	Image Specification Document	Description
Vim	/opt/SLES/POS/system/templates/addons/vim.xml	Adds Vim (Vi IMproved) to client images. Vim is an almost compatible version of the UNIX editor vi. Almost every possible command can be performed using only ASCII characters. Many new features have been added such as multilevel undo, command line history, filename completion, block operations, and editing of binary data. Vi is available for the AMIGA, MS-DOS, Windows NT, and various versions of UNIX. This feature can be added to any client image.
VNC 4 Remote Control Client	/opt/SLES/POS/system/templates/addons/vnc.xml	Adds the VNC 4 Remote Control client to the image so you can remotely control the Point of Service terminal over any TCP/IP connection. This feature can be added to Java, Browser or Desktop images.
YaST2	/opt/SLES/POS/system/templates/addons/yast2.xml	Adds the YaST2 console to client images. YaST2 is the system configuration console. It can configure hardware (sound cards, printers, keyboards, mice), network connections (network cards, ISDN cards, modems, DSL connections), network clients and services (NFS, NIS), as well as a general system options (language, partitioning, software, bootloader). This feature can be added only to the Desktop image.

4.5 POSBranch Images

For smaller stores where the Branch Server is running only the Point of Service infrastructure (that is, the Branch Server is running no additional applications), the Branch Server can be deployed as a control terminal running on Point of Service hardware.

NOTE: Although the POSBranch Server is intended to run only the Point of Service infrastructure, the POSBranch Server image can be extended to include some Point of Service applications, provided the terminal has adequate hardware and memory resources.

There is no Image Description Tree for the POSBranch image.

The Image Specification Document for the POSBranch image is `/opt/SLES/POS/system/templates/support/branch.xml`. This template provides the following Branch Server components:

- All the RPMs required for a functional Branch Server
- The Linux Kernel Crash Dump (LKCD) to provide a system for detecting, saving and examining system crashes
- The RPM database so YaST Online Update (YOU) can be used to update the image
- Branch Server configuration information obtained from the LDAP directory

To create a POSBranch image, the `branch.xml` template must included as a child document in one of the four client Image Specification Documents: Minimal, Java, Browser, or Desktop.

For information on creating a POSBranch image, see Section 10.2, "Building POSBranch Images," on page 176.

4.6 LDAP Image Reference Objects

Client images distributed to Point of Service terminals must have corresponding Image Reference objects (scPosImage) in the LDAP directory. Required attributes for the scPosImage object include:

- The image name (scImageName)
- The name of the image file (scImageFile)
- The image version (scPosImageVersion)

During the configuration of the Administration Server, `posInitLdap.sh` or `posInitEdir.sh` automatically create an scPosImage object for the Minimal image under the Default Distribution Container (scDistributionContainer). Other Image Reference objects must be manually created using posAdmin. For more information on this procedure, see Section 6.5.1, "Adding an scPosImage Object," on page 80. For information on scPosImage objects, see Chapter 5, "The Novell Linux Point of Service LDAP Directory," on page 55.

Unlike client images, boot images do not have reference objects in the LDAP directory. The CDBoot image obviously needs no reference in the LDAP directory because it is self contained; when you generate a CDBoot image, everything the Point of Service terminal needs to boot is provided in the CD ISO image.

Likewise, the DiskNetboot image does not require an scPosImage object in the LDAP directory; however, it is referenced within the Distribution Container object (scDistributionContainer). The initrd and kernel images loaded by each Point of Service terminal are determined by the scDistributionContainer object in which its associated client image object (scPosImage) is located. The scInitrdName and scKernelName attributes in the scDistributionContainer object define the initrd and kernel images for the container. All Point of Service terminals that load client images located within the Distribution Container use the designated boot images at load time.

4.7 Image Naming Conventions

When you build an image with ImageBuilder, the image filename is derived from the name and version of the Image Description Tree used to generate the image, plus the creation date. ImageBuilder names the file as follows:

`image_name-version-date`

- *Image_name* is derived from the name of the Image Description Tree referenced when you build the image.
- *Version* is derived from the version number of the Image Description Tree referenced when you build the image.
- *Date* is the image creation date.

4.7.1 Cloning an Image Description Tree

When you want to create a new image, the first step is to clone an existing Image Description Tree using either scr or xscr. When you type these commands, you provide the name and version of the existing tree and the name and version of the tree you are creating.

Command Syntax for scr

The basic syntax to clone an Image Description Tree in scr is as follows:

```
scr --create image_name-version --image image_name-version
```

For example, the following scr command clones the Minimal-2.0.21 Image Description Tree to create a new Image Description Tree named myImage-1.1.1:

```
scr --create myImage-1.1.1 --image minimal-2.0.21
```

The new Image Description Tree is located at `/opt/SLES/POS/system/myImage-1.1.1`. When you build an image from this tree, it will be named `myImage-1.1.1-`*`date`*.

Command Syntax for xscr

Similarly, the basic syntax to clone an Image Description Tree in xscr is as follows:

```
xscr --create image_name-version --image image_name-version --dist
nld|sles
```

The following xscr command clones the Desktop-2.0.21 Image Description Tree to create a new SLES-based Image Description Tree named `myImage-2.1.1`:

```
xscr --create myImage-2.1.1 --image desktop-2.0.21 --dist sles
```

The new Image Description Tree is located at `/opt/SLES/POS/system/myImage-2.1.1`. When you build an image from this tree, it is named `myImage-2.1.1-`*`date`*.

4.7.2 Items to Note

You cannot use the word “boot” in any image name other than the CDboot and DiskNetboot images.

When you clone an Image Description Tree, scr writes the image name to `/opt/SLES/POS/system/`*`image_name-version`*`/config` and the version to `/opt/SLES/POS/system/`*`image_name-version`*`/VERSION`. xscr writes the image name to the ImageSpecification element’s ImageName attribute and the version to the ImageSpecification element’s ImageVersion attribute within `ImageSpecification.xml`.

In both cases, the image name must correspond to the scImageName attribute and the version must correspond to the scPosImageVersion attribute within the Image Reference object (scPosImage) in the LDAP tree.

ImageBuilder maintains up to five builds of a single image in the same directory. When you generate the sixth build of an image, the oldest image version is deleted. (The utility determines the oldest image version by the image date.) If you want to maintain more than five versions of a single image, you must maintain them in separate directories.

If you plan to create an `.iso` file of an image that is larger than 650 MB in size, use the compression option so that it will fit on a standard CD.

The Novell Linux Point of Service LDAP Directory

5

All system information (system structure, the configuration and deployment method for each Branch Server, available client images, and Point of Service terminal types) is stored in an LDAP directory on the Administration Server.

The Novell® Linux Point of Service LDAP directory can run on OpenLDAP or Novell eDirectory™.

NOTE: posInitLdap provides an LDAP LDIF file that defines the directory schema and the initial records. This LDIF file can be imported into IBM Tivoli Directory Server to run the Novell Linux Point of Service LDAP directory on IBM Directory Services.

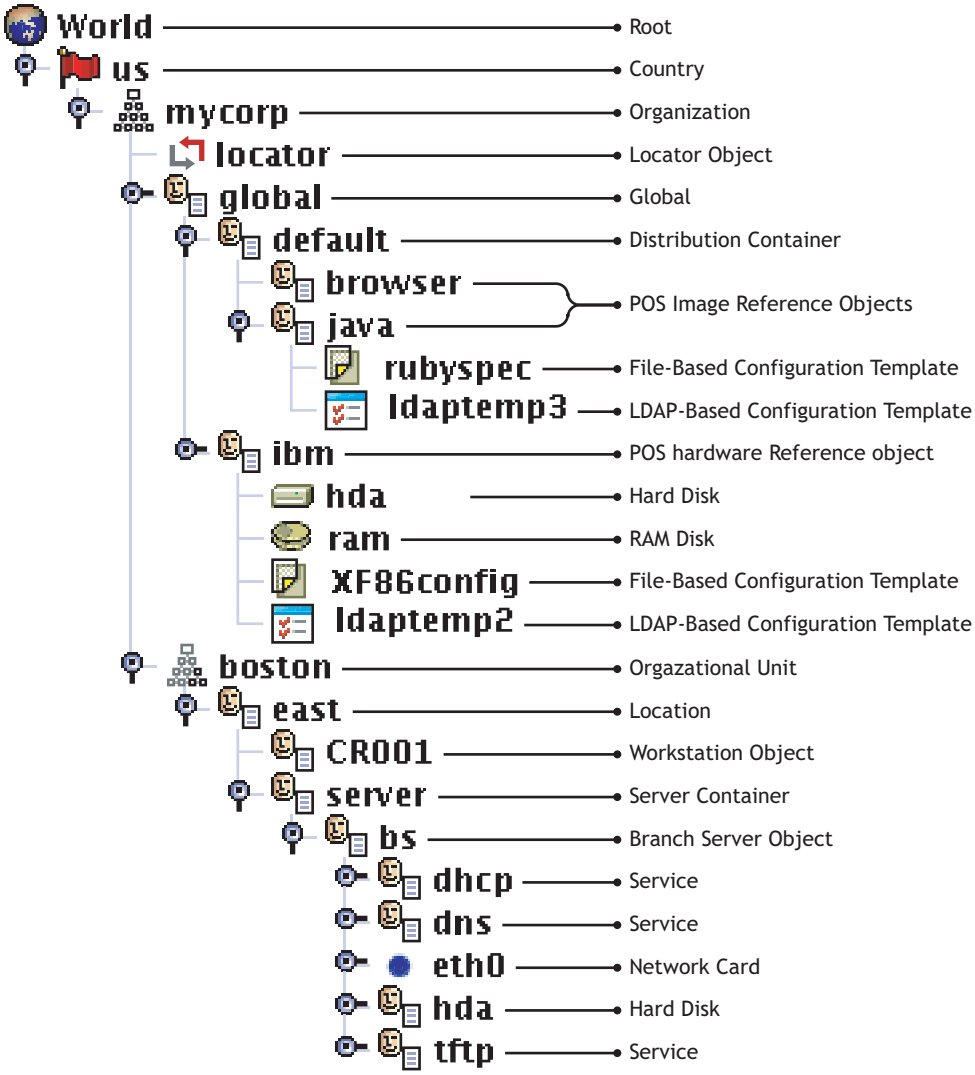
This section reviews the Novell Linux Point of Service LDAP directory.

LDAP entries are managed using the posAdmin tool. For more information on posAdmin, see Chapter 6, “Using posAdmin to Manage the LDAP Directory,” on page 65.

5.1 Logical Structure of the LDAP Directory

The LDAP directory is designed with multiple, hierarchical object classes so it can accommodate large corporate structures. Figure 5-1 shows an example of a typical LDAP directory structure for a Novell Linux Point of Service system.

Figure 5-1 *Novell Linux Point of Service LDAP directory structure*



The following is a hierarchical description of standard object classes represented in the Novell Linux Point of Service LDAP directory tree. For a complete listing of Novell Linux Point of Service object classes and their attributes, see Section 5.2, “LDAP Objects,” on page 61.

`Root`: The beginning level in the LDAP tree. The root represents the world.

`Country`: The country in which the organization is located.

`Organization` (organization): The name of the organization represented in the LDAP tree.

`Locator Object` (scHardware): Identifies where the Global container is located.

When the Branch Server queries LDAP for global configuration information, it queries this object to determine where the Global container is located.

`Global` (scRefObjectContainer, cn=global): All globally valid information for a chain or company—that is server hardware, Point of Service hardware, or client images—is stored in the Global container in the form of reference objects. These reference objects are linked to the actual entries for the Point of Service terminals and servers in the branches using distinguished names.

The initial LDAP structure after installation includes only one scRefObjectContainer named global under the directory root. Other scRefObjectContainer objects can be added as needed; however, the scRefObjectContainer container objects should always have cn=global and also appear only once per directory level. This provides great flexibility. For example, each server can be assigned its own reference objects and, therefore, its own hardware types. On the other hand, if all the servers have the same hardware, a unified standard can be defined in the global container on the regional or organizational level.

`Distribution Container` (scDistributionContainer): A container for distribution sets of images.

A distribution set is a collection of images designed for Point of Service terminals on a given version of the Linux kernel. The Default distribution container references the Linux 2.6 kernel.

The images that ship with Novell Linux Point of Service 9 are built on Novell Linux Desktop (NLD), which runs the Linux 2.6 kernel. Therefore, the reference objects for Novell Linux Point of Service 9 images must be created in the Default Distribution Container.

IMPORTANT: If you migrate from SLRS 8 to Novell Linux Point of Service 9, the migration script creates the SLRS 8 distribution container. This container references the SLRS 8 kernel and therefore, must store all the scPosImage objects for SLRS 8 images. For more information, see "Migrating from SLRS 8 to Novell Linux Point of Service 9" in the *Novell Linux Point of Service 9 Installation Guide*.

`Image Reference Object` (scPosImage): The Image Reference object stores information about an image stored on the Administration Server.

By default, a Image Reference object is created for the Minimal client image. For information on adding this object class to the LDAP directory, see Section 6.5.1, "Adding an scPosImage Object," on page 80.

IMPORTANT: If you migrate from SLRS 8 to Novell Linux Point of Service 9, the migration script moves the existing scPosImage objects to the SLRS 8 distribution container. For more information, see "Migrating from SLRS 8 to Novell Linux Point of Service 9" in the *Novell Linux Point of Service 9 Installation Guide*.

`File-Based Configuration Template` (scConfigFileSyncTemplate): scConfigFileSyncTemplate objects are used when you run services, such as X Windows, that require hardware-dependent configuration files. The scConfigFileSyncTemplate object points to the configuration file that a Point of Service terminal needs to run a given service. This object differs from scConfigFileTemplate objects because the configuration data is not stored in the object; rather, the object points to a configuration file outside the LDAP directory.

This element can also exist under scCashRegister objects.

For information on adding this object class to the LDAP directory, see Section 6.4.3, “Adding an scConfigFileSyncTemplate Object,” on page 77.

`LDAP-Based Configuration Template` (scConfigFileTemplate): scConfigFileTemplate objects are used when you run services, such as the X Window service, that require hardware-dependent configuration files. An scConfigFileTemplate object contains the configuration file data that a Point of Service terminal needs to run a given service.

This element can also exist under scCashRegister objects.

For information on adding this object class to the LDAP directory, see Section 6.4.2, “Adding an scConfigFileTemplate Object,” on page 76.

`Hardware Reference Object` (scCashRegister): The Hardware Reference object stores information about Point of Service hardware.

Typically, you should define a scCashRegister object for each type of terminal used on the Novell Linux Point of Service system; however, if a Point of Service terminal does not have an scCashRegister object for its specific hardware type, it will use the configuration defined in the default scCashRegister object. For information on adding this object class to the LDAP directory, see Section 6.4.1, “Adding an scCashRegister Object,” on page 75.

IMPORTANT: If you migrate from SLRS 8 to Novell Linux Point of Service 9, the migration script updates the existing scCashRegister objects to point to scPosImage objects in the SLRS 8 distribution container. For more information, see “Migrating from SLRS 8 to Novell Linux Point of Service 9” in the *Novell Linux Point of Service 9 Installation Guide*.

`Hard Disk` (scHardDisk): The configuration for a Point of Service terminal hard disk.

For information on adding this object class to the LDAP directory, see Section 6.4.5, “Adding an scHarddisk Object,” on page 79.

`RAM Disk` (scRamDisk): The configuration for a Point of Service terminal RAM disk.

For information on adding this object class to the LDAP directory, see Section 6.4.4, “Adding an scRAMDisk Object,” on page 78.

`File-Based Configuration Template` (scConfigFileSyncTemplate): scConfigFileSyncTemplate objects are used when you run services, such as the X Window service, that require hardware-dependent configuration files. The scConfigFileSyncTemplate object points to the configuration file that a Point of Service terminal needs to run a given service. This object differs from scConfigFileTemplate objects because the configuration data is not stored in the object; rather, the object points to a configuration file outside the LDAP directory.

This element can also exist under scPosImage objects.

For information on adding this object class to the LDAP directory, see Section 6.4.3, “Adding an scConfigFileSyncTemplate Object,” on page 77.

`LDAP-Based Configuration Template` (scConfigFileTemplate): scConfigFileTemplate objects are used when you run services, such as X Windows, that require hardware-dependent configuration files. An scConfigFileTemplate object contains the configuration file data that a Point of Service terminal needs to run a given service.

This element can also exist under scPosImage objects.

For information on adding this object class to the LDAP directory, see Section 6.4.2, “Adding an scConfigFileTemplate Object,” on page 76.

`Organizational Units` (organizationalUnit): Organization units were introduced to improve organizational coherence. They typically represent organizational structures such as regions, branches or divisions.

For information on adding this object class to the LDAP directory, see Section 6.3.1, “Adding organizationalUnit Objects,” on page 67.

`Location` (scLocation): A branch office; that is, a site where a Branch Server and Point of Service terminals are located. Location containers are used to store information about the deployed Point of Service terminals and the Branch Servers. This and all other information that can be modified at the Branch Server should be stored or referenced in the Location containers to limit the need to grant write privileges to subtrees.

For information on adding this object class to the LDAP directory, see Section 6.3.2, “Adding an scLocation Object,” on page 68.

`Workstation` (scWorkstation): The Workstation object stores information for a specific Point of Service terminal. Using information from the Hardware Reference object (scCashRegister) and Image Reference object (scPosImage), posldap2crconfig.pl automatically creates a Workstation object in the LDAP directory for every Point of Service terminal that registers on the Branch Server. For information on this process, see Section 3.5.3, “The hwtype.MAC_address File,” on page 34.

`Server Container` (scServerContainer): A container for all the Branch Server objects for a given site. The information pertaining to the Branch Servers is stored in the Server container

To provide system redundancy and failover, there can be multiple Branch Servers for each site.

For information on adding this object class to the LDAP directory, see Section 6.3.3, “Adding an scServerContainer and scBranchServer Object,” on page 69.

`Branch Server` (scBranchServer): The Branch Server object stores configuration information that is specific to each Branch Server. There must be a Branch Server object for every Branch Server in the Novell Linux Point of Service system.

IMPORTANT: The location of the scBranchServer object in the LDAP directory must correspond to the hostname defined for the Admin/Branch Server during installation. For example, if the hostname is bs.east.boston.mycorp.us, the dn of the scBranchServer object would be cn=bs,cn=server, cn=east,ou=boston,o=mycorp,c=us. You must create the scBranchServer object and its supporting organizational structure before you can run posInitBranchserver.sh and deploy the Branch Server. For more information on defining the server hostname during installation, see “Network Interfaces” on page 30. For information on creating the Branch Server objects, see Section 6.3, “Defining Branch Objects,” on page 67.

The Administration Server does not have an associated object in the LDAP tree structure.

For information on adding this object class to the LDAP directory, see Section 6.3.3, “Adding an scServerContainer and scBranchServer Object,” on page 69.

`Service` (scService): The configuration for a Branch Server service like DNS, TFTP, or DHCP.

For information on adding this object class to the LDAP directory, see Section 6.3.3, “Adding an scServerContainer and scBranchServer Object,” on page 69.

`High Availability Service` (scHAService): The configuration for a high availability Branch Server service such as DNS, TFTP, or DHCP.

For information on adding this object class to the LDAP directory, see Section 6.3.4, “Adding a Branch Server with High Availability Services (scHAService),” on page 71.

`Network Card` (scNetworkcard): The configuration for a Branch Server network interface card.

For information on adding this object class to the LDAP directory, see Section 6.3.4, “Adding a Branch Server with High Availability Services (scHAService),” on page 71.

`Hard Disk` (scHardDisk): The configuration for the Branch Server's boot hard disk.

For information on adding this object class to the LDAP directory, see Section 6.4.5, “Adding an scHarddisk Object,” on page 79.

To illustrate how the directory structure is used, here is a sample query procedure using objects from the example LDAP structure described above.

1. A search is made for an object of objectClass: scLocation with cn=eastbay.

   NOTE: The core scripts search only the names of the object classes. The common name for an entry is not used.

2. Below this scLocation, a search is made for an object of objectClass: scServerContainer (server).
3. Below this scServerContainer, a search is made for an object of objectClass: scBranchServer with cn=bs.
4. Data specific to this server is located below this scBranchServer object, such as objects of objectClass: scNetworkcard in which the IP addresses are indicated.
5. All the data that generally applies for this hardware type, such as the partitioning, is read from a reference object of objectClass: scRefServer in which this hardware is described. These reference objects are always organized as containers in an object of objectClass: scRefObjectContainer.
6. Next, the reference objects that are valid for this Branch Server are located. First, the attribute scRefServerDn in the scBranchServer object that represents this server is read. If a DN is included here, the target is used as the reference object for the Branch Server.
7. If the entry is empty, the search for an object of the objectClass: scHardware moves upward in the directory structure, one level at a time. If the attribute scRefServerDn is occupied in this type of object, this DN is taken as the target; if not, the search continues upward in the directory structure. If no appropriate object with this attribute is found all the way up to the root level, the process aborts with an error.

The procedure is similar for Point of Service terminal hardware. In this example, in addition to the referenced hardware type (through attribute scRefPcDn to a scCashRegister object), scPosImageDn points to the reference image, scPosImage object.

5.2 LDAP Objects

Table 5-1 provides an alphabetical listing of all the Novell Linux Point of Service elements represented in the LDAP directory. The Must attributes for each element are those attributes that must be defined when creating the element with posAdmin. The May attributes are optional. All of the elements are structural.

Table 5-1 *Alphabetical listing of Novell Linux Point of Service elements in the LDAP directory*

Name	Must Attributes	May Attributes	Description
scBranchServer	cn	scRefServerDn scPubKey	Server marker
scCashRegister	cn scCashRegisterName	scPosImageDn scDiskJournal	Point of Service terminal
scConfigFile SyncTemplate	cn scMust scConfigFile scBsize scConfigFileLocalPath	scConfigMd5 description	Configuration file template
scConfigFile Template	cn scMust scConfigFile scBsize	scConfigFileData scConfigFileParser scConfigMd5 description	Configuration file template

Name	Must Attributes	May Attributes	Description
scDistribution Container	cn scKernelName scInitrdName scKernelVersion scKernelMatch	scKernelExpression	Container for distribution sets of images; contains the kernel information
scHardDisk	cn scDevice scHdSize scPartitionsTable		Description of a hard disk, normally a leaf entry of a scRefServer or a scBranchServer
scHardware	cn	scPosImageDn scRefPcDn scRefMonitorDn scRefServerDn	Reference to standard PC hardware type and server hardware
scHAService	cn ipHostNumber scDnsName scServiceName scServiceStatus scServiceStartScript scPrimaryService	scDevice	High Availability service for a Branch Server Cluster
scLocation	cn ipNetworkNumber ipNetmaskNumber scDhcpRange scDhcpFixedRange scDefaultGw scDynamicIp	scLdapDn scDnsDn scWorkstationBaseName scPrinterBaseName scEnumerationMask associatedDomain	Defaults for an office
scNetworkcard	scDevice ipHostNumber	macAddress scModul scModulOption ipNetmaskNumber	Description of a network card, normally a subentry of a scBranchServer
scPosImage	cn scImageName scPosImageVersion scDhcpOptionsRemote scDhcpOptionsLocal scImageFile scBsize	scConfigFile	Image object
scRamDisk	cn scDevice		Ramdisk
scRefObject Container	cn		Reference object container
scServer Container	cn		Server container

Name	Must Attributes	May Attributes	Description
scService	cn ipHostNumber scDnsName scServiceName scServiceStartScript scServiceStatus	scServiceEmail	Server service, such as LDAP
scWorkstation	cn macAddress ipHostNumber	scSerialNumber scRefPcDn scPosImageDn scPosImageVersion scPOSRegisterBiosVersion scConfigFileDn scStandardPrinterDn userPassword scStandardPrinter scPOSGroupDn scDiskJournal scConfigUpdate scNotifiedimage	Entry for a specific, physical workstation

Using posAdmin to Manage the LDAP Directory

6

In a Novell® Linux Point of Service system, `posAdmin.pl` is a command line tool used to add, modify, and remove Branch Server and Point of Service terminal information in the LDAP directory.

This section reviews how to use posAdmin to manage objects in the LDAP directory.

6.1 Mandatory LDAP Objects

When you run the `posInitLdap.sh` or `posInitEdir.sh` script to configure the LDAP directory on the Administration Server, the following objects are automatically created:

- Country
- Organization
- Locator Object (scHardware)
- Global Container (scRefObjectContainer)
- Default Distribution Container (scDistributionContainer)
- scPosImage object for the Minimal image

With these objects in place, you must then use posAdmin to create the following mandatory objects in the LDAP tree:

- Branch Objects

  IMPORTANT: You must create the scBranchServer object and its supporting organizational structure before you can run `posInitBranchserver.sh` and deploy the Branch Server.

  - One or more organizationalUnit objects to represent your organization’s structure.
  - An scLocation object for each site where a Branch Server is located.
  - An scServerContainer to contain all the Branch Server objects for a given site.

- An scBranchServer object and its associated configuration objects for each Branch Server in your system:

- Point of Service image (scPosImage) objects for the client image files that you want the Branch Server to distribute to Point of Service terminals.

  IMPORTANT: You must create the scPosImage objects and set the scPosImageVersion attribute to Active before you boot the Point of Service terminals. The Point of Service terminals require an scPosImage object with an active scPosImageVersion attribute before they can download the corresponding physical image from the Branch Server at boot time. For more information on setting the scPosImageVersion attribute to Active, see Section 6.5.2, "Activating Images," on page 81.

- An scCashRegister object and its associated configuration objects for each type of Point of Service terminal in your system:
  - scHarddisk or scRamDisk
  - scConfigFileTemplate (optional)
  - scConfigFileSyncTemplate (optional)

When you boot the Point of Service terminals, `posldap2crconfig.pl` automatically creates a Workstation object (scWorkstation) in the LDAP directory for every Point of Service terminal that registers on the Branch Server. For information on this process, see Section 3.5.3, "The hwtype.MAC_address File," on page 34.

After the scWorkstation objects exist in the directory, you can then define attributes specific to particular workstations. For example, you can assign a specific client image (scPosImage) object to a workstation. For instructions on this procedure, see Section 6.5.3, "Assigning an Image to a Point of Service Terminal," on page 82.

The following sections provide information to help you create and manage objects in the LDAP directory.

6.2 General Command Options

Table 6-1 outlines general posAdmin command line options.

Table 6-1 *posAdmin command line options*

Option	Description
`--user` `--password`	Used primarily for authentication as a user identified by a password. For example, `posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret` If you do not authenticate using command line options, you are prompted for a user name and password.

Option	Description

`--base`	Specifies a base context in the LDAP directory. When you add a new location (branch), you specify an organization or organizational unit as a base. For example,
	`--base o=mycorp,c=us`
	`--base ou=boston,o=mycorp,c=us`
	In some cases, you can use an abbreviation or a common name for the base. This is possible only if the common name is a unique value in the directory. For example,
	`--base boston`
	If posAdmin cannot determine the base (no base or more than one base is found), it exits with an error message.

`--help`	Displays a usage message that summarizes the basic command options.

6.3 Defining Branch Objects

This section reviews the steps to add the following objects to the LDAP directory:

- Section 6.3.1, "Adding organizationalUnit Objects," on page 67
- Section 6.3.2, "Adding an scLocation Object," on page 68
- Section 6.3.3, "Adding an scServerContainer and scBranchServer Object," on page 69
- Section 6.3.4, "Adding a Branch Server with High Availability Services (scHAService)," on page 71

NOTE: Each LDAP object has two types of attributes: must and may attributes. The must attributes are the minimum requirements for an object. The may attributes are optional. This table lists only those may attributes that are relevant to Novell Linux Point of Service.

6.3.1 Adding organizationalUnit Objects

organizationalUnit objects were introduced to improve organizational coherence. They typically represent organizational structures such as regions, branches or divisions. Because they can be nested, they can be used to visually represent the structure or organization of your company.

Table 6-2 summarizes the posAdmin command options for organizationalUnit object attributes.

Table 6-2 *Command options for creating organizationalUnit objects*

Option	Type	Description
`--ou`	must	The name of the organizational unit; for example, boston. **IMPORTANT:** Use only alphanumeric characters.
`--description`	may	A human-readable description of the object.

The following command adds the boston organizational unit to the LDAP directory (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base o=mycorp,c=us --add --organizationalUnit --ou boston
```

The context of the Organizational Unit is ou=boston,o=mycorp,c=us directory.

You can add a description to the boston entry by adding the following option to the command:

```
--description 'Central Boston Headquarters'
```

6.3.2 Adding an scLocation Object

An scLocation object typically is used to represent a branch office; that is, a site where a Branch Server and Point of Service terminals are located. scLocation containers are used to store information about the deployed Branch Servers and Point of Service terminals. This and all other information that can be modified at the Branch Server should be stored or referenced in the Location containers to limit the need to grant Write privileges to subtrees.

Table 6-3 summarizes the posAdmin command options for scLocation object attributes.

Table 6-3 *Command options for creating scLocation objects*

Option	Type	Description
`--cn`	must	The common name of the location.
`--ipNetworkNumber`	must	The network address of the subnet of the branch; for example,192.168.1.0.
`--ipNetmaskNumber`	must	The netmask of the subnet of the branch; for example, 255.255.255.0.
`--scDhcpRange`	must	The dynamic IP address range of the DHCP server of the subnet. This is needed to register the Point of Service terminals. It is a comma-separated value pair; for example, 192.168.1.10, 192.168.1.50.
`--scDhcpFixedRange`	must	The fixed IP address range of the DHCP server reserved for the Point of Service terminals. It is also a comma-separated value pair, such as 192.168.1.51, 192.168.1.150.
`--scDefaultGw`	must	The default gateway for this location. This is normally a router to the corporate wide area network.
`--scDynamicIp`	must	This flag is used to enable or disable the dynamic IP address range of the DHCP server. Allowed values are TRUE to enable or FALSE to disable dynamic IP address ranges.

Option	Type	Description
`--scWorkstationBaseName`	must	The base name of the Point of Service terminals of a branch used to create a unique name for each terminal in combination with the scDhcpFixedRange attribute and the scEnumerationMask. For example, using the scWorkstationBaseName CR, an scEnumerationMask of 000, and the above-mentioned scDhcpFixedRange to build the name of the Point of Service terminals and their corresponding IP addresses, the first newly registered terminal gets the name CR001 and the IP address 192.168.1.51; the next terminal is named CR002 and gets the IP address 192.168.1.52; and so on.
`--scEnumerationMask`	must	Refer to scWorkstationBaseName.
`--associatedDomain`	may	This optional entry configures the DNS domain and the domain part of the hostnames of the Point of Service terminals to be in the stated domain. If this entry is left empty, the domain consists of the LDAP structure of the scLocation entry DN. With this entry, a different domain can be chosen.

The following command adds an scLocation named harbor to the LDAP directory (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base ou=boston,o=mycorp,c=us --add --scLocation --cn harbor
--ipNetworkNumber 192.168.1.0 --ipNetmaskNumber 255.255.255.0
--scDhcpRange 192.168.1.10,192.168.1.50
--scDhcpFixedRange 192.168.1.51,192.168.1.151
--scDefaultGw 192.168.1.254 --scDynamicIp TRUE
--scWorkstationBaseName CR --scEnumerationMask 000
```

6.3.3 Adding an scServerContainer and scBranchServer Object

There must be an scBranchServer object for every Branch Server in the Novell Linux Point of Service system. These objects store configuration information specific to each Branch Server.

An scBranchServer object contains information about hardware, at least one defined network card, and services like TFTP, DNS, and DHCP. It is located with an scLocation object in the LDAP tree.

IMPORTANT: The location of the scBranchServer object in the LDAP directory must correspond to the hostname defined for the Admin/Branch Server during installation. For example, if the hostname is bs in east.boston.mycorp.us, the dn of the scBranchServer object would be cn=bs,cn=server, cn=east,ou=boston,o=mycorp,c=us. For more information on defining the server hostname during installation, see "Network Interfaces" in the *Novell Linux Point of Service 9 Installation Guide*.

Here is the procedure to add an scBranchServer object to the LDAP directory with posAdmin.

1 Before you can add the scBranchServer to an scLocation object, you must define a scServerContainer.

  This is done with the `--scServerContainer` and common name (`--cn`) options. For example (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base cn=east,ou=boston,o=mycorp,c=us
--add --scServerContainer --cn server
```

2 In the new scServerContainer, add a Branch Server object.

This is done with the `--scBranchServer` and common name (`--cn`) options. For example (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base cn=server,cn=east,ou=boston,o=mycorp,c=us
--add --scBranchServer --cn bs
```

Optionally, you can define the reference hardware with the `--scRefServerDn` option, a pointer (Distinguished Name) to the global directory.

3 Add a network interface card with a static IP address from the defined subnet.

This is done with the `--scNetworkcard` option and the `--scDevice` and `--scIpHostNumber` attributes. For example (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base cn=bs,cn=server,cn=east,ou=boston,o=mycorp,c=us
--add --scNetworkcard --scDevice eth0 --ipHostNumber 192.168.1.1
```

Table 6-4 summarizes the posAdmin command options for scNetworkcard attributes.

Table 6-4 *Command options for creating scNetworkcard objects*

Option	Type	Description
`--scDevice`	must	The name of network device of the card; for example, eth0 or eth1.
`--ipHostNumber`	must	The IP address; for example, 192.168.1.1.
`--macAddress`	may	The MAC address of the network interface card.
`--scModul`	may	The name of the Linux kernel module for the network interface card.
`--scModulOption`	may	The module options of the Linux kernel module for the network interface card.
`--ipNetmaskNumber`	may	If the ipHostNumber is not inside the defined subnet of the location, add the netmask belonging to the IP address assigned to the network interface card.

4 Set up the Branch Server services. At a minimum, define the required DNS, TFTP and DHCP services.

The following examples demonstrate how to add the DNS, DHCP, and TFTP services.

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base cn=bs,cn=server,cn=east,ou=boston,o=mycorp,c=us
--add --scService --cn dns --ipHostNumber 192.168.1.1
--scDnsName dns --scServiceName dns --scServiceStartScript named
--scServiceStatus TRUE
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base cn=bs,cn=server,cn=east,ou=boston,o=mycorp,c=us
--add --scService --cn dhcp --ipHostNumber 192.168.1.1
```

```
--scDnsName dhcp --scServiceName dhcp
--scServiceStartScript dhcpd --scServiceStatus TRUE
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base cn=bs,cn=server,cn=east,ou=boston,o=mycorp,c=us
--add - scService --cn tftp --ipHostNumber 192.168.1.1
--scDnsName tftp --scServiceName tftp
--scServiceStartScript atftpd --scServiceStatus TRUE
```

Table 6-5 summarizes the posAdmin command options for the scService object attributes.

Table 6-5 *Command options for creating scService objects*

Option	Type	Description
`--cn`	must	The common name of the service.
`--ipHostNumber`	must	The virtual IP address of the HA Service.
`--scDnsName`	must	The DNS name of the service.
`--scServiceName`	must	The name of the service; for example, dns, dhcp, tftp.
`--scServiceStartScript`	must	The name of the init script in `/etc/init.d`; for example, atftpd for the tftp service.
`--scServiceStatus`	must	The status of the service. TRUE or FALSE are possible values.
`--scServiceEmail`	may	The email address where the service should send email notifications.

6.3.4 Adding a Branch Server with High Availability Services (scHAService)

A high availability (HA) Branch Server performs the same functions as a standard Branch Server with the following differences:

- The HA Branch Server is configured as a two-server cluster.
- It requires at least two network interface cards per server.
- Instead of scService objects, the HA Branch Server has scHAService objects.

For information on installing a HA Branch Server pair, see "Setting Up High Availability Branch Servers" in the *Novell Linux Point of Service 9 Installation Guide*.

Here is the procedure required to add a HA Branch Server object to the LDAP directory.

1. Define a scServerContainer.

   This is done with the `--scServerContainer` option and the common name (`--cn`) attribute. For example (type the command all on one line):

   ```
   posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
   --base cn=east,ou=boston,o=mycorp,c=us
   --add --scServerContainer --cn server
   ```

2. In the new scServerContainer, create two Branch Server objects.

The following commands create the BS1 and BS2 Branch Server objects in the scServerContainer.

```
#\#  bs1
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scBranchServer --cn bs1
#\#  bs2
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --basecn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scBranchServer --cn bs2
```

3 Add the network interface cards for each Branch Server.

Depending on network traffic and the desired performance, you can configure one to four network interface cards per Branch Server. For general information on how the network cards can be implemented on the network, see "Meeting System Requirements" in the *Novell Linux Point of Service 9 Installation Guide*. For specific information on the network interface card configuration, see "Network Interfaces" in the *Novell Linux Point of Service 9 Installation Guide*.

The following examples demonstrate how to add network interface cards for the Branch, DRBD, and Heartbeat interfaces to the LDAP configuration.

```
#\# eth1 on the BS1 for the Branch Server interface
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=bs1,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scNetworkcard --scDevice eth1
    --ipHostNumber 192.168.1.1
#\# eth1 on BS2 for the Branch Server interface
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=bs2,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scNetworkcard --scDevice eth1
    --ipHostNumber 192.168.1.2
#\# eth1:0 for the Branch Server interface virtual IP
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=bs1,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scNetworkcard --scDevice eth1:0
    --ipHostNumber 192.168.1.3
#\# eth2 on BS1 for the DRBD interface
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=bs1,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scNetworkcard --scDevice eth2
    --ipHostNumber 192.168.2.1
#\# eth2 on BS2 for the DRBD interface
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=bs2,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scNetworkcard --scDevice eth2
    --ipHostNumber 192.168.2.2
#\# eth3 on BS1 for the Heartbeat interface
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=bs1,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scNetworkcard --scDevice eth3
    --ipHostNumber 192.168.3.1
#\# eth3 on BS2 for the Heartbeat interface
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
```

```
    --base cn=bs2,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scNetworkcard --scDevice eth3
    --ipHostNumber 192.168.3.2
```

Table 6-6 summarizes the posAdmin command options for scNetworkcard object attributes.

Table 6-6 *Command options for creating scNetworkcard objects*

Attribute	Type	Explanation
`--scDevice`	must	The name of network device of the card. For example, eth0 or eth1.
`--ipHostNumber`	must	The IP address. For example, 192.168.1.1.
`--macAddress`	may	The MAC address of the network interface card.
`--scModul`	may	The name of the Linux kernel module for the network interface card.
`--scModulOption`	may	The module options of the Linux kernel module for the network interface card.
`--ipNetmaskNumber`	may	If the ipHostNumber is not inside the defined subnet of the location, add the netmask belonging to the IP address assigned to the network interface card.

4 Add DNS, DHCP, and TFTP as HA services.

The following commands demonstrate how to add DNS, DHCP, and TFTP as HA services.

```
#\# DNS on BS1 as primary service
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=bs1,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scHAService --cn dns --ipHostNumber 192.168.1.3
    --cDnsName dns --scServiceName dns
    --scServiceStartScript named
    --scServiceStatus TRUE --scPrimaryService TRUE
#\# DHCP on BS1 as primary service
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=bs1,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scHAService --cn dhcp --ipHostNumber 192.168.1.3
    --scDnsName dhcp --scServiceName dhcp
    --scServiceStartScript dhcpd
    --scServiceStatus TRUE --scPrimaryService TRUE
#\# TFTP on BS1 as primary service
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=bs1,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scHAService --cn tftp --ipHostNumber 192.168.1.3
    --scDnsName tftp --scServiceName tftp
    --scServiceStartScript atftpd
    --scServiceStatus TRUE --scPrimaryService TRUE
#\# DNS on BS2 as backup service
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=bs2,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scHAService --cn dns --ipHostNumber 192.168.1.3
    --scDnsName dns --scServiceName dns
```

```
    --scServiceStartScript named
    --scServiceStatus TRUE - scPrimaryService FALSE
#\# DHCP on BS2 as backup service
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=bs2,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scHAService --cn dhcp --ipHostNumber 192.168.1.3
    --scDnsName dhcp --scServiceName dhcp
    --scServiceStartScript dhcpd
    --scServiceStatus TRUE --scPrimaryService FALSE
#\# TFTP on BS2 as backup service
    posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
    --base cn=bs2,cn=server,cn=east,ou=boston,o=mycorp,c=us
    --add --scHAService --cn tftp --ipHostNumber 192.168.1.3
    --scDnsName tftp --scServiceName tftp
    --scServiceStartScript atftpd
    --scServiceStatus TRUE --scPrimaryService FALSE
```

Table 6-7 summarizes the posAdmin command options for scHAService object attributes.

Table 6-7 *Command options for creating scHAService objects*

Attribute	Type	Explanation
`--cn`	must	The common name of the service.
`--ipHostNumber`	must	The virtual IP address of the HA Service.
`--scDnsName`	must	The DNS name of the service.
`--scServiceName`	must	The name of the service; for example: dns, dhcp, tftp.
`--scServiceStartScript`	must	The name of the init script in `/etc/init.d`; for example, atftpd for the tftp service.
`--scServiceStatus`	must	The status of the service. TRUE or FALSE are possible values.
`--scPrimaryService`	must	This flag is used to describe if this a primary service or not. TRUE or FALSE are the possible values. If you define a primary server, this flag is always TRUE. On a secondary server, this flag is always FALSE.
`--scServiceEmail`	may	The email address where the service should send email notifications.

6.4 Defining Point of Service Terminal Objects

With posAdmin, you can add, remove, and modify Point of Service terminal hardware assets such as Point of Service terminals, configuration files, hard disks, network interface cards, and configuration files with the use of reference objects in the LDAP directory. Hardware reference objects are typically located in the global container in the LDAP directory.

The following sections outline how to use posAdmin to manage Point of Service terminal hardware reference objects in LDAP:

- Section 6.4.1, “Adding an scCashRegister Object,” on page 75

6.4.1 Adding an scCashRegister Object

The first step to register new Point of Service hardware is to define the name and model type of the Point of Service terminal. The scCashRegister object stores information about Point of Service hardware. Typically, you should define a scCashRegister object for each type of terminal used on the Novell Linux Point of Service system; however, if a Point of Service terminal does not have an scCashRegister object for its specific hardware type, it uses the configuration defined in the default scCashRegister object.

NOTE: To create a default scCashRegister object, define the object’s scCashRegisterName attribute as Default.

The scCashRegister objects are stored in the Global container so they can be accessed by all Branch Servers.

Table 6-8 summarizes the posAdmin command options for scCashRegister object attributes.

Table 6-8 *Command options for creating scCashRegister objects*

Option	Type	Description
`--cn`	must	The common name of the Point of Service terminal.
`--scCashRegisterName`	must	The model type of the Point of Service terminal. If this field is defined as “default,” the current scCashRegister object is used as the default Point of Service configuration. If a Point of Service terminal does not have an scCashRegister object for its specific hardware type, it will use the configuration defined in the default scCashRegister object. **IMPORTANT:** Define only one default scCashRegister object in the Global container.
`--scPosImageDn`	may	The distinguished name of the default client image defined for this Point of Service terminal type. **NOTE:** A specific client image can be defined in the scWorkstation object. The setting in the scWorkstation object overrides the default image defined in the scCashRegister object. For information on this procedure, see Section 6.5.3, “Assigning an Image to a Point of Service Terminal,” on page 82.
`--scDiskJournal`	may	This Boolean field is set to TRUE if journaling should be enabled. Journaling is only added on disk-based machines.

The following example adds a default scCashRegister object below the global container (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base  cn=global,o=mycorp,c=us  --add --scCashRegister --cn crtype3
--scCashRegisterName 1234567
--scPosImageDn cn=browser,cn=global,o=mycorp,c=us --default
```

6.4.2 Adding an scConfigFileTemplate Object

scConfigFileTemplate objects are used when you run services, such as the X Window service, that require hardware-dependent configuration files. An scConfigFileTemplate object contains the configuration file data that a Point of Service terminal needs to run a given service.

When you define the scConfigFileTemplate object, you designate a source configuration file (--scConfigFileData). posAdmin extracts the configuration data from the source file and stores it in the scConfigFileTemplate object. When a Point of Service terminal registers with a Branch Server (or when you run posAdmin.pl --updateconfig or posldap2crconfig.pl --dumpall), the Branch Server retrieves the configuration data in the scConfigFileTemplate object to create a configuration file in /tftpboot/CR/*MAC_address*/ directories on the Branch Server.

Using TFTP, the configuration file is then distributed from the Branch Server to the appropriate Point of Services terminals at boot time.

NOTE: The scCashRegister or scPosImage object under which the scConfigFileTemplate object is created determines which Point of Service terminals receive the configuration file.

If the scConfigFileTemplate object is defined under an scCashRegister object, all terminals that correspond to the type defined in the scCashRegister object receive the configuration file defined in the scConfigFileTemplate object.

If the scConfigFileTemplate object is defined under an scPosImage object, all terminals that load the client image that corresponds to the scPosImage object receive the configuration file defined in the scConfigFileTemplate object.

Table 6-9 summarizes the posAdmin command options for scConfigFileTemplate object attributes.

Table 6-9 *Command options for scConfigFileTemplate objects*

Option	Type	Description
`--cn`	must	The common name of the configuration file.
`--scMust`	must	This flag is used to enable or disable the configuration file. Allowed values are TRUE to enable or FALSE to disable the configuration file.
`--scConfigFile`	must	Specifies the path where the configuration file is installed on the Point of Service terminal. For example, `/etc/ntp.conf` or `/etc/X11/XF86Config`.
`--scBsize`	must	Specifies the block size for the TFTP download.
`--scConfigFileData`	must	The source path of the configuration file. For example, `/tmp/XF86Config.mydata`.

Option	Type	Description
`--description`	may	A description of the configuration file.
`--scConfigFileparser`	may	The name of the parserFunction to apply.
`--scConfigMd5`	may	The MD5 checksum value of the configuration file.
`--scConfigFileUpdateModel`	may	The update model for synchronizing configuration files. Allowed values are “pulled” and “changed”.

The following example adds a scConfigFileTemplate object below the Hardware Reference object, crtype3 (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base  cn=crtype3,cn=global,o=mycorp,c=us
--add --scConfigFileTemplate --cn XF86Config
--scConfigFile /etc/X11/XF86Config --scBsize 1024
--scConfigFileData /mydata/XF86Config.1234567
```

6.4.3 Adding an scConfigFileSyncTemplate Object

scConfigFileSyncTemplate objects are used when you run services, such as the X Window service, that require hardware-dependent configuration files. The scConfigFileSyncTemplate object points to the configuration file that a Point of Service terminal needs to run a given service. This object differs from scConfigFileTemplate objects because the configuration data is not stored in the object; rather, the object points to a configuration file outside the LDAP directory.

When a Point of Service terminal registers with a Branch Server (or when you run `posAdmin.pl --updateconfig` or `posldap2crconfig.pl --dumpall`), the Branch Server uses RSYNC to transfer the configuration file designated in the scConfigFileSyncTemplate object from the `/opt/SLES/POS/rsync/config/` directory on the Administration Server to `/tftpboot/CR/`*`MAC_address`*`/` directories on the Branch Server.

IMPORTANT: Any configuration files referenced in the scConfigFileSyncTemplate object must be located in the `/opt/SLES/POS/rsync/config/` directory on the Administration Server.

Using TFTP, the configuration file is then distributed from the Branch Server to the appropriate Point of Service terminals at boot time.

NOTE: The scCashRegister or scPosImage object under which the scConfigFileSyncTemplate object is created determines which Point of Service terminals receive the configuration file.

If the scConfigFileSyncTemplate object is defined under an scCashRegister object, all terminals that correspond to the type defined in the scCashRegister object receive the configuration file designated in the scConfigFileSyncTemplate object.

If the scConfigFileSyncTemplate object is defined under an scPosImage object, all terminals that load the client image that corresponds to the scPosImage object receive the configuration file designated in the scConfigFileSyncTemplate object.

Table 6-10 summarizes the posAdmin command options for scConfigFileSyncTemplate object attributes.

Table 6-10 *Command options for scConfigFileSyncTemplate objects*

Option	Type	Description
`--cn`	must	The common name of the configuration file.
`--scMust`	must	This flag is used to enable or disable the configuration file. Allowed values are TRUE to enable or FALSE to disable the configuration file.
`--scConfigFile`	must	Specifies the path where the configuration file is installed on the Point of Service terminal. For example, `/etc/ntp.conf` or `/etc/X11/XF86Config`.
`--scBsize`	must	Specifies the block size for the TFTP download.
`--scConfigFileLocalPath`	must	The local source path of the configuration file. For example, `/opt/SLES/POS/rsync/config/XF86Config.mydata`.
`--description`	may	A description of the configuration file.
`--scConfigMd5`	may	The MD5 checksum value of the configuration file.

The following example adds an scConfigFileSyncTemplate object below the Hardware Reference object, crtype3 (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base cn=crtype3,cn=global,o=mycorp,c=us
--add --scConfigFileSyncTemplate --cn XF86Config
--scConfigFile /etc/X11/XF86Config --scMust TRUE --scBsize 1024
--scConfigFileLocalPath /opt/SLES/POS/rsync/config/XF86Config.1234567
```

6.4.4 Adding an scRAMDisk Object

The scRamDisk object stores configuration information for a Point of Service terminal RAM disk. If no hard disk is available, you must configure a RAM disk for the Point of Service terminal.

Table 6-11 summarizes the posAdmin command options forscRamDisk object attributes.

Table 6-11 *Command options for scRamDisk objects*

Option	Type	Description
`--base`	must	The base distinguished name of the Hardware Reference object. For example, cn=crtype3, cn=global,o=mycorp,c=us.
`--cn`	must	The common name of the device. For example, ram.
`--scDevice`	must	The RAM disk device. **IMPORTANT:** The device /dev/ram0 cannot be used because it is used for the initial RAM disk. Therefore, we recommend using /dev/ram1. The RAM device should not be confused with the hard disk device, which uses a partition table.

The following example adds an scRamDisk object below the Hardware Reference object, crtype3 (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base cn=crtype3,cn=global,o=mycorp,c=us
--add --scRamDisk --cn ram --scDevice /dev/ram1
```

6.4.5 Adding an scHarddisk Object

The scHarddisk object stores configuration information for a Point of Service terminal hard disk.

Table 6-12 summarizes the posAdmin command options for scHarddisk object attributes.

Table 6-12 *Command options for scHarddisk objects*

Option	Type	Description
`--base`	must	The base distinguished name of the Hardware Reference object. For example, cn=crtype3, cn=global,o=mycorp,c=us.
`--cn`	must	The common name of the device. For example, hda.
`--scDevice`	must	The device of the hard disk. For example, /dev/hda.
`--scHdSize`	must	The size of the hard disk in megabytes.
`--scPartitionsTable`	must	A semicolon-separated (';') list of partition entries. Each entry has four parameters: the size in megabytes, the partition type ID (82 for swap, 83 for a Linux partition), the mount point, and the file system (swap or ext3).

The following example adds an scHarddisk object below the scCashRegister object, crtype3 (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base  cn=crtype3,cn=global,o=mycorp,c=us  --add --scHarddisk
--cn hda --scDevice /dev/hda2 --scHdSize 9000
--scPartitionsTable '1000 82 swap swap;4000 83 ext3;'
```

6.5 Managing Image Objects

Images are managed in LDAP with scPosImage objects. The scPosImage object stores information about an image stored on the Administration Server. For more information on images, see Chapter 8, "Building Images with the scr ImageBuilder Tool," on page 97.

The following sections outline how to use posAdmin to manage images in LDAP:

NOTE: Each LDAP object has two types of attributes: must and may attributes. The must attributes are the minimum requirements for an object. The may attributes are optional.

6.5.1 Adding an scPosImage Object

Every client image that you want to distribute to Point of Service terminals must have a corresponding scPosImage object in the LDAP directory. These objects are typically organized within Distribution Container objects under the Global container in the LDAP tree.

NOTE: Boot images do no have scPosImage objects; they are referenced in the scInitrdName attribute in the scDistributionContainer object.

After the installation and configuration of the Novell Linux Point of Service, an scPosImage object is automatically added to the Default Distribution Container for the Minimal image. However, this LDAP entry is only intended to serve as an example. You must manually add an scPosImage object for each client image you want to distribute to Point of Service terminals.

IMPORTANT: The images that ship with Novell Linux Point of Service 9 are built on Novell Linux Desktop (NLD) which runs the Linux 2.6 kernel. Therefore, the reference objects for Novell Linux Point of Service 9 images must be created in the Default Distribution Container.

If you migrate from SLRS 8 to Novell Linux Point of Service 9, the migration script creates the SLRS 8 distribution container. This container references the SLRS 8 kernel and therefore, must store all the scPosImage objects for SLRS 8 images. For more information, see "Migrating from SLRS 8 to Novell Linux Point of Service 9" in the *Novell Linux Point of Service 9 Installation Guide*.

Table 6-13 summarizes the posAdmin command options for scPosImage object attributes.

Table 6-13 *Command options for scPosImage objects*

Option	Type	Description
`--base`	must	The base distinguished name of the scPosImage object; for example, cn=global,o=mycorp,c=us.
`--cn`	must	The common name of the client image; for example, myjava.
`--scImageName`	must	The name of the client image; for example, myjava.
`--scPosImageVersion`	must	The version number of the client image, followed by the flag passive or active; for example, 2.0.21; active. The version number and the flag are semicolon-separated (';'). There are several combinations possible of this attribute, which are described in Table 6-14.
`--scDhcpOptionsRemote`	must	The boot option of the Point of Service terminal.The mandatory value is /boot/pxelinux.0.
`--scDhcpOptionsLocal`	reserved	This attribute is reserved for future extension of the Novell Linux Point of Service and is not used at this time.
`--scImageFile`	must	The filename of the image, which the terminal will download from the Branch Server; for example, myjava.

Option	Type	Description
--scBsize	must	Specifies the block size for the TFTP download of the client image. For example, 8192. possible values are: 4096 (4 KB) for image sizes less than 128 MB, 8192 (8 KB) for image sizes less than 256MB, 16384 (16 KB) for image sizes less than 512 MB and 32768 (32 KB) for image sizes less than 1GB. You must select a TFTP block size of 32 KB for the full-featured Desktop image, because there is a limitation of the block counter for TFTP.
`--scConfigFile`	may	Specifies the path where the configuration file is installed on the Point of Service terminal; for example, /etc/ntp.conf or /etc/X11/XF86Config.

The following example adds a scPosImage object below the Global container (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us  --password secret
--base cn=global,o=mycorp,c=us --add --scPosImage --cn myJava
--scImageName myJava --scPosImageVersion "2.0.21;active"
--scDhcpOptionsRemote /boot/pxelinux.0 --scDhcpOptionsLocal LOCALBOOT
--scImageFile myJava --scConfigFile /etc/X11/XF86Config
```

6.5.2 Activating Images

Each image can be available in several versions, as shown in Table 6-14. The scPosImageVersion attribute in each scPosImage object must be set to either active or passive. Active versions are downloaded by the Branch Server. If there are multiple active versions of a single image, the Branch Server selects the highest active version. Passive image versions are never downloaded to the Branch Server unless they are explicitly configured in the scWorkstation entry for the individual Point of Service terminal.

Table 6-14 *Possible values for the scPosImageVersion attribute*

Value	Description
1.1.2	The version number is set to 1.1.2, but this client image is disabled in LDAP and cannot be used for a new Point of Service terminal, even when the scCashRegister object that corresponds to the Point of Service terminal matches the scPosImageDn attribute entry.
1.1.2;passive	Same behavior as above.
1.1.2;active	This client image with version 1.1.2 is enabled and downloaded to the Point of Service terminals.
1.1.2;active 1.1.3;active 1.1.5;active	All image versions are enabled, but only the latest image version is downloaded to the Point of Service terminals.
1.1.2;passive 1.1.3;active 1.1.5;passive	Only image version 1.1.3 is enabled and downloaded to the Point of Service terminals.

To activate a registered image, set its scPosImageVersion attribute to active. This is done with posAdmin using the `--modify` keyword and the `--multival` (multivalue) option as follows (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--modify --scPosImage --multival
--scPosImageVersion '2.3.10;passive=>2.3.10;active'
--DN cn=browser,cn=default,cn=global,o=mycorp,c=us
```

To activate the new image version on a Branch Server, use `possyncimages.pl` and `posldap2crconfig.pl` with the `--dumpall` option.

```
possyncimages
posldap2crconfig --dumpall
```

6.5.3 Assigning an Image to a Point of Service Terminal

You can assign a specific image to a Point of Service terminal through its scWorkstation object.

The following command assigns Browser image 2.3.10 to the CR001 scWorkstation object in the boston1 container (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--modify --scWorkstation
--scPosImageDn cn=browser,cn=default,cn=global,o=mycorp,c=us
--scPosImageVersion 2.3.10
--DN cn=CR001,cn=boston1,ou=boston,o=mycorp,c=us
```

When you explicitly assign an image in the scWorkstation entry, the active or passive flag set for the scPosImage object in the global container is ignored.

NOTE: The scWorkstation object is automatically created in the LDAP directory the first time you boot a Point of Service terminal. The posleases2ldap daemon automatically triggers `posldap2crconfig.pl` which then creates an scWorkstation object and hardware configuration files for each Point of Service terminal that registers on the Branch Server. For more information on this process, see Chapter 3, “Point of Service Terminals,” on page 25.

6.5.4 Removing Images

To remove the image assigned to a workstation, run the following command (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--remove --scWorkstation --scPosImageDn  --scPosImageVersion
--DN  cn=CR001,ou=boston1,ou=boston,o=mycorp,c=us
```

6.6 Modifying LDAP Entries

The modify option enables you to modify an existing object attributes and add or delete may attributes.

To add or to modify attributes, specify the element, an attribute value pair, and a DN. The main difference between command arguments in add, remove, and modify operations is that the add operation specifies the base DN of the directory element below which the new entry should be

created with the `--base` option. The modify and remove operations identify the target element with the `--DN` option.

NOTE: If an operation is not finished successfully, posAdmin returns an error message.

Table 6-15 summarizes the posAdmin command options for modifying LDAP objects.

Table 6-15 *posAdmin modify command options*

Attribute	Type	Explanation
`--DN`	must	Distinguished name of the element to modify.
`--object`	must	Object with must or may attributes to be modified; for example, scWorkstation.
`--attribute`	must	Attribute; for example, scPosImageVersion.
`--value`	may	If a value is given the attribute is modified; otherwise, the attribute entry is deleted.

6.6.1 Adding and Removing an organizationalUnit Object Description

The following command adds a description to an organizationalUnit with the DN of ou=boston,o=mycorp,c=us:

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--DN ou=boston,o=mycorp,c=us --modify --organizationalUnit
--description 'my description of boston'
```

The following command removes the object description:

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--DN ou=boston,o=mycorp,c=us --modify --organizationalUnit
--description
```

6.6.2 Defining a Specific Image for a scWorkstation Object

The following command defines a specific client image (`--scPosImageDn`) and version (`--scPosImageVersion`) for scWorkstation object cn=pos01,cn=Lab,ou=boston, o=mycorp,c=us (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--DN cn=pos01,cn=Lab,ou=boston,o=mycorp,c=us
--modify --scWorkstation --scPosImageDn java
--scPosImageVersion 1.1.3
```

6.7 Removing LDAP Entries

To remove an object from the Novell Linux Point of Service LDAP directory, use the `--remove` option and the `--DN` attribute with the distinguished name of the object to delete. If the referred object has subentries, you must add the `--recursive` option.

Table 6-16 summarizes the posAdmin command options for deleting LDAP objects.

Table 6-16 Command options for deleting LDAP objects

Option	Type	Description
`--DN`	must	Distinguished name of the object to delete
`--recursive`	may	Option to delete an object with all sub-objects.

The following command deletes an scServerContainer with all servers and all services (type the command all on one line):

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--remove --recursive --DN cn=server,cn=east,ou=boston,o=mycorp,c=us
```

6.8 Querying LDAP Objects

To query an object, use the `--query` option, an object option such as `--scLocation` or `--scBranchServer`, and, if desired, an attribute-value pair.

Table 6-17 summarizes the posAdmin command options for querying the LDAP database.

Table 6-17 Command options for querying the LDAP database

Option	Type	Description
`--base`	must	The base option sets the base in which to search for objects. On the Administration Server, the default base is the organization (o=mycorp,c=us).
`--object`	must	Object to be queried; for example, --scLocation.
`--attribute`	may	Attribute to search within the specified object; for example, --ipNetworkNumber
`--value`	may	If an attribute value is given, only objects with matching values are searched.

The following examples illustrate possible posAdmin queries.

Example 1

List all locations with all data in boston:

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base ou=boston,o=mycorp,c=us --query --scLocation
```

Example 2

List all locations in boston that show only the ipNetworkNumber:

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base ou=boston,o=mycorp,c=us
--query --scLocation --ipNetworkNumber
```

Example 3

List all locations in boston that show only the ipNetworkNumber 192.168.1.0:

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base ou=boston,o=mycorp,c=us --query --scLocation
--ipNetworkNumber 192.168.1.0
```

6.9 Updating config.*MAC_address* and Hardware Configuration Files

If the configuration information changes for a scPosImage or scCashRegister object, you can run posAdmin with the `--updateconfig` option. This command notifies Branch Servers to update the hardware configuration and `config.`*`MAC_address`* files for specified terminals in their subnet.

When this command is executed, the Administration Server uses the RSYNC service to send an update notification file (`/opt/SLES/POS/rsync/update/update`) to all Branch Servers that service terminals designated in the `--dnList`. It also sets the scConfigUpdate attribute to TRUE for every scWorkstation object designated (either directly or indirectly) in the `--dnList`. When posleases2ldap encounters the update notification file on the Branch Server, it connects to the LDAP directory, checks the scConfigUpdate attribute in the scWorkstation objects, and refreshes the terminals' hardware configuration and `config.`*`MAC_address`* files.

IMPORTANT: For the `--updateconfig` option to work, the Branch Server object in LDAP (scBranchServer) must have an scNetworkCard object with an IP address (ipHostNumber attribute) that is visible to the Administration Server. Multiple scNetworkCard objects can exist under an scBranchServer object

---

Table 6-18 summarizes the posAdmin options for updating configuration information.

Table 6-18 *Command options for updating configuration files*

Option	Type	Description
`--base`	must	The base option sets the base in which to search for objects. On the Administration Server, the default base is the organization (o=mycorp,c=us).
`--dnList`	must	A list of distinguished names that indicates which terminals should receive updated `config.`*`MAC_address`* files. Valid object types are scPosImage, scCashRegister, scConfigFileTemplate, scConfigFileSyncTemplate and scWorkstation. This list is delimited by colons (:). For example: `--dnList cn=crtype3,cn=global,o=mycorp,c=us:cn=CR001, cn=branch,ou=boston,o=mycorp,c=us`

The following examples illustrate how the `--updateconfig` command can be used.

Example 1

Update all configuration files on clients that use the scCashRegister object cn=crtype3,cn=global, o=mycorp,c=us:

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base ou=boston,o=mycorp,c=us --updateconfig
--dnList cn=crtype3,cn=global,o=mycorp,c=us
```

Example 2

Update all configuration files on clients that use the image object cn=browser,cn=global, o=mycorp,c=us and the client cn=CR001,cn=branch,ou=boston,o=mycorp,c=us:

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base ou=boston,o=mycorp,c=us --updateconfig
--dnList cn=browser,cn=global,o=mycorp,c=us:
  cn=CR001,cn=branch,ou=boston,o=mycorp,c=us
```

Managing Image Source Files with POSCDTool and POSCopyTool

7

Before you can create images for Point of Service terminals, you must copy the image source files from the Novell® Linux Point of Service CDs to a central distribution directory and create the reference files that ImageBuilder needs to locate the image source files.

Novell Linux Point of Service provides two command line utilities to simplify the process of managing the source files required to build client images: POSCDTool and POSCopyTool. This section reviews the POSCDTool and POSCopyTool commands and procedures required to manage the image source files.

7.1 POSCDTool Command Line Options

POSCDTool is a command line utility that performs the initial system preparation required to build client images with ImageBuilder.

The POSCDTool command syntax is as follows:

```
poscdtool [options]
```

Table 7-1 summarizes the available POSCDTool options.

Table 7-1 *POSCDTool command options*

Option	Description
`--copy`	Copies the Novell Linux Point of Service CDs to the distribution directory structure. **NOTE:** The distribution directory structure is referenced in the AdminServer.conf and Distribution.xml files so ImageBuilder can locate the RPMs required to build the image. For more information, see Section 7.3.1, “Copying the Novell Linux Point of Service CDs,” on page 92.
`[--type=cd|dir|iso]`	Indicates the source media to be copied (CD, directory, or ISO). This parameter is optional. If it is not defined, POSCDTool assumes the source media is CD. **NOTE:** The iso option is not currently supported.

Option		Description
	`--source=`*`source_media`*	Indicates the path to the source media. For a CD, the path is expressed as /media/*cdrom_name*, For example, cdrom, dvdrecorder, cdrecorder, dvd. For a directory, the contents of the directory are treated as a CD. For an ISO, the path is expressed as a full path including the filename.
	`[--dest=`*`distribution_directory`*`]`	Indicates the distribution directory to which the CD media is copied. This parameter is optional. If it is not defined, POSCDTool copies to the default distribution directory, /opt/SLES/POS/dist/. **IMPORTANT:** /opt/SLES/POS/dist/ is the default path ImageBuilder uses to build client images. The CDs are copied by distribution and CD number. That is, under the destination directory (/opt/SLES/POS/dist/), there are distribution directories (NLD, SLES, SLRS). Within each of the distribution directories are revision directories (FCS, SP1, SP2, and so forth). Under each revision directory are CD directories (CD1, CD2, and so forth). A complete path might appear as follows: /opt/SLES/POS/dist/SLRS/FCS/CD1/
	`[--force]`	Forces POSCCDTool to copy the source CDs, even if they already exist in the distribution directory.
`--link`		Creates a link between the distribution directory structure required by ImageBuilder and the directories where the CD source media are currently located. This option is required only if the Novell Linux Point of Service CDs are not archived in the distribution directory structure required by ImageBuilder. For example, if the CD source media is mounted on an NFS server to provide a single point of installation for Administration and Branch Servers throughout your network, you can use the --link option to create a link between the files' current location and the distribution directory structure ImageBuilder requires to build images. **NOTE:** If you copy the CDs using POSCDTool, the CDs are automatically copied to the distribution directory structure required by ImageBuilder. For more information, see Section 7.3.2, “Linking the Novell Linux Point of Service CDs,” on page 93.

Option		Description
	`--source=`*`source_media`*	Indicates the source path for the link. This is the path to the CD source media.
	`[--dest=`*`distribution_directory`*`]`	Indicates the destination path for the link. This is the path to the distribution directory on the Administration Server. This parameter is optional. If it is not defined, POSCDTool links to the default distribution directory, /opt/SLES/POS/dist/.
`--mount`		Mounts the Novell Linux Point of Service CDs. This option is required only if the CD source media are on another server. For example, if you require multiple Administration Servers to build images, you can copy the CD source media to one of those servers and mount the other image servers to that source. This eliminates the need to copy the CD source media on every Administration Server. For more information, see Section 7.3.3, “Mounting the Novell Linux Point of Service CDs,” on page 94.
	`--source=`*`mount_source`*	Indicates the mount source.
	`[--dest=`*`distribution_directory`*`]`	Indicates the mount endpoint; that is, the distribution directory. This parameter is optional. If it is not defined, POSCDTool mounts to /opt/SLES/POS/dist/.
`--generate`		Generates the AdminServer.conf file for scr and the Distribution.xml document for xscr. For more information, see Section 7.3.4, “Generating AdminServer.conf or Distribution.xml,” on page 94.
	`[--type=conf|xml]`	Designates what type of file to create. This parameter is optional. If it is not defined,POSCDTool creates both the AdminServer.conf file and the Distribution.xml document. conf generates the AdminServer.conf file. xml generates the Distribution.xml document.
	`[--source=`*`distribution_directory`*`]`	Indicates the distribution directory where the CD source media is located. This parameter is optional. If it is not defined. POSCDTool uses the default distribution directory, /opt/SLES/POS/dist/.

Option		Description
	[--dest=*output_path*]	Indicates the destination path for the output file. This parameter is optional. If it is not defined, POSCDTool creates the files as follows: ◆ The AdminServer.conf file is created in /etc/opt/SLES/POS/. ◆ The Distribution.xml document is created in /opt/SLES/POS/system/template/.
	[--imageclass=NLD|SLES]	Defines the ImageClass element in the Distribution.xml document. The parameter is optional. If it is not defined, POSCDTool creates the Distribution.xml file with both the SLES and NLD image classes. POSCDTool verifies you have the CD source that corresponds to each image class before it generates the distribution.xml document. **IMPORTANT:** The ImageClass element in the Distribution Source Document must match the ImageClass element in the Image Specification Document.
--verify		Verifies the availability of the CD source files that correspond to the image classes defined in the Distribution.xml document.
	[--source=*distribution_directory*]	Indicates the distribution directory where the CDs have been copied, linked, or mounted. This parameter is optional. If it is not defined, POSCDTool references the default distribution directory, /opt/SLES/POS/dist/.
	[--imageclass=NLD|SLES]	Indicates which CD source set should be verified. NLD or SLES. This parameter is optional. If it is not defined, POSCDTool verifies both image classes. If you want to restrict the verification process, designate a specific image class.

7.2 POSCopyTool Command Line Options

POSCopyTool is a simplified version of POSCDTool that performs only the following tasks:

- It copies the NLD and SLES source CDs to the default distribution directory, /opt/SLES/POS/dist/. As POSCopyTool copies the source CDs, it prompts you for the CDs it needs to complete the process. You can provide the CDs in any order; POSCopyTool tracks which CDs it has copied.

- It creates the `AdminServer.conf` file and `Distribution.xml` document in their default directories. It defines the `Distribution.xml` document with both the NLD and SLES image classes.
- It verifies the availability of the both the NLD and SLES CD source files.

The POSCopyTool command syntax is as follows:

`poscopytool [options]`

Table 7-2 summarizes the available POSCopyTool command options.

Table 7-2 *POSCopyTool command options*

Option	Description
`--source=source_media`	Indicates the path to the source media. For a CD, the path is expressed as /media/*cdrom_name*, For example, cdrom, dvdrecorder, cdrecorder, dvd. For a directory, the contents of the directory are treated as a CD. For an ISO, the path is expressed as a full path including the filename.
`[--dest=distribution_directory]`	Indicates the distribution directory to which the CD media is copied. This parameter is optional. If it is not defined, POSCDTool copies to the default distribution directory, /opt/SLES/POS/dist/. **IMPORTANT:** /opt/SLES/POS/dist/ is the default path ImageBuilder uses to build client images. The CDs are copied by distribution and CD number. That is, under the destination directory (/opt/SLES/POS/dist/), there are distribution directories (NLD, SLES, SLRS). Within each of the distribution directories are revision directories (FCS, SP1, SP2, and so forth). Under each revision directory are CD directories (CD1, CD2, and so forth). A complete path might appear as follows: /opt/SLES/POS/dist/SLRS/FCS/CD1/
--list	Lists the CDs the POSCopyTool will copy to the distribution directory.
`[--force]`	Forces POSCopyTool to copy the source CDs, even if they already exist in the distribution directory.

7.3 Managing the Image Source Files

This section reviews tasks required to prepare the Administration Server tasks to build client images with ImageBuilder.

- Section 7.3.1, “Copying the Novell Linux Point of Service CDs,” on page 92

This task is required for all Administration Servers where you want to build images. You can copy the Novell Linux Point of Service CDs using either POSCDTool or POSCopyTool.

- Section 7.3.2, “Linking the Novell Linux Point of Service CDs,” on page 93

  This task is required only if you are maintaining the product CDs in another directory structure-for example, if you store the product CDs on an NFS server to provide a single point of installation for Administration and Branch Servers throughout your network. You must use POSCDTool to link the Novell Linux Point of Service CDs.

- Section 7.3.3, “Mounting the Novell Linux Point of Service CDs,” on page 94

  This task is required only if you have multiple Administration Servers where you want to build images and you want to mount the servers to a single distribution directory rather than copy the Novell Linux Point of Service CDs to each server. You must use POSCDTool to mount the Novell Linux Point of Service CDs.

- Section 7.3.4, “Generating AdminServer.conf or Distribution.xml,” on page 94

  This task is required for all Administration Servers where you want to build images. You can manually generate the AdminServer.conf or Distribution.xml files with POSCDTool. POSCopyTool automatically generates these files after completing the copy procedure.

- Section 7.3.5, “Verifying CD Availability,” on page 95

  We recommend you verify the Novell Linux Point of Service CDs are available in the distribution directories before you try to build an image. You can manually verify the CD availability with POSCDTool. POSCopyTool automatically verifies CD availability after completing the copy procedure.

7.3.1 Copying the Novell Linux Point of Service CDs

POSCDTool and POSCopyTool copy the RPM software packages used to build NLD-based client images to the following distribution directory structure:

NOTE: The following bullet list shows the NLD CDs in the default distribution directory structure, `/opt/SLES/POS/dist`.

- SLESCD0=/opt/SLES/POS/dist/NLD9/SP1/CD1
- SLESCD1=/opt/SLES/POS/dist/NLD9/SP1/CD2
- SLESCD2=/opt/SLES/POS/dist/NLD9/FCS/CD1
- SLESCD3=/opt/SLES/POS/dist/NLD9/FCS/CD2
- SLESCD4=/opt/SLES/POS/dist/NLD9/FCS/CD3
- SLESCD5=/opt/SLES/POS/dist/NLPOS9/FCS/CD4

POSCDTool and POSCopyTool copy the RPM software packages used to build SLES-based POSBranch images to the following distribution directory structure:

NOTE: The following bullet list shows the SLES CDs in the default distribution directory structure, `/opt/SLES/POS/dist`.

- SLESCD0=/opt/SLES/POS/dist/NLPOS9/FCS/CD1
- SLESCD1=/opt/SLES/POS/dist/NLPOS9/FCS/CD2
- SLESCD2=/opt/SLES/POS/dist/NLPOS9/FCS/CD4

- SLESCD3=/opt/SLES/POS/dist/SLES9/FCS/CD1
- SLESCD4=/opt/SLES/POS/dist/SLES9/FCS/CD2
- SLESCD5=/opt/SLES/POS/dist/SLES9/FCS/CD3

NOTE: POSBranch images can only be generated with the xscr tool. For more information, see Section 10.2, "Building POSBranch Images," on page 176.

After you copy the CDs using POSCDTool or POSCopyTool, ImageBuilder can use the copied files to build images.

POSCDTool Command

The copy command syntax for POSCDTool is as follows:

```
poscdtool.pl --copy [--type=cd|dir] --source=source_media [--
dest=distribution_directory]
```

For example, the following command copies the Novell Linux Point of Service CDs from a CD source to the default distribution directory, opt/SLES/POS/dist/:

```
poscdtool.pl --copy --source=/media/cdrom
```

POSCopyTool Command

The copy command syntax for POSCopyTool is as follows:

```
poscopytool.pl --source=path
```

For example:

```
poscopytool.pl --source=/media/dvd
```

When you use POSCopyTool, it performs the following functions:

- Copies the Novell Linux Point of Service CDs from the designated source media to the default distribution directory, `opt/SLES/POS/dist/`.
- Creates `AdminServer.conf` in `/opt/SLES/POS/`.
- Creates Distribution.xml with both the NLD and SLES image classes in `/opt/SLES/POS/system/templates/`.
- Verifies the source CDs were correctly copied to the distribution directory.

7.3.2 Linking the Novell Linux Point of Service CDs

ImageBuilder cannot access the RPMs on the Novell Linux Point of Service CDs unless they are stored in the distribution file structure.

Under the destination directory, ImageBuilder requires that the Novell Linux Point of Service CDs be archived in distribution directories (NLD, SLES, SLRS). Within each of the distribution directories are revision directories (FCS, SP1, SP2, and so forth). Under each revision directory are CD directories (CD1, CD2, and so forth).

If you are maintaining the product CDs in another directory structure—for example, if you store the product CDs on an NFS server to provide a single point of installation for Administration and Branch Servers throughout your network—you must link the source CDs to the distribution file structure.

IMPORTANT: This option is required only if the Novell Linux Point of Service CDs are not archived in the distribution file structure. If you copy the CDs using POSCDTool, the CDs are automatically copied to the distribution file structure.

The link command syntax is as follows:

```
poscdtool.pl --link --source=source_media
[--dest=distribution_directory]
```

For example, the following command links the Novell Linux Point of Service CDs from a CD source on an NFS server to the default distribution directory, `/opt/SLES/POS/dist/`:

```
poscdtool.pl --link --source=/nfs/cd
```

This command links the CDs on the NFS server to the default distribution directory, `opt/SLES/POS/dist/`.

7.3.3 Mounting the Novell Linux Point of Service CDs

If you have multiple Administration Servers where you want to build images, you can mount a single distribution directory on each server rather than copying the Novell Linux Point of Service CDs to each server.

The mount command syntax is as follows:

```
poscdtool.pl --mount --source=mount_source
[--dest=distribution_directory]
```

For example, the following command mounts the default distribution directory on an Administration Server to one on another Administration Server:

```
poscdtool.pl --mount --source=adminserver1:/hd1
```

7.3.4 Generating AdminServer.conf or Distribution.xml

The `AdminServer.conf` and `Distribution.xml` files define the paths to the distribution directories where you have copied the Novell Linux Point of Service CDs. ImageBuilder searches these paths to locate the RPM packages required to build images.

`AdminServer.conf` is used by scr. This ASCII, line-based file is located at `etc/opt/SLES/POS/`. For more information on the `AdminServer.conf` file structure and contents, see Section 8.2.2, “AdminServer.conf,” on page 107.

The `Distribution.xml` document is used by xscr. By default, the `Distribution.xml` document is located in `/opt/SLES/POS/system/templates/`. For information on the `Distribution.xml` elements and attributes, see Section 9.2.3, “Distribution Source Document (Distribution.xml),” on page 140.

Upon completion of the copy procedure, POSCopyTool automatically generates the `AdminServer.conf` and `Distribution.xml` files.

If necessary, you can use the following syntax to manually generate `AdminServer.conf` and `Distribution.xml` with POSCDTool (type the command all on one line):

```
poscdtool.pl --generate [--type=conf|xml]
[--source=distribution_directory] [--dest=output_path]
[--imageclass=NLD|SLES]
```

For example, the following command generates both `AdminServer.conf` and `Distribution.xml` using the default distribution directory. `Distribution.xml` is defined with both the NLD and SLES image classes and the document is saved to the default output path, `/opt/SLES/POS/system/template/`. The final `AdminServer.conf` file is saved to the default output path, `/opt/SLES/POS/`.

```
poscdtool.pl --generate
```

The following command uses the default distribution directory structure to create only the `AdminServer.conf` file in the default output path, `/etc/opt/SLES/POS/`.

```
poscdtool.pl --generate --type=conf
```

The following command uses the default distribution directory structure to create only the `Distribution.xml` document with the SLES image class. The document is saved to the default output path, `/opt/SLES/POS/system/template/`.

```
poscdtool.pl --generate --type=xml --imageclass=SLES
```

7.3.5 Verifying CD Availability

After all the initial configuration is complete, it is recommended that you verify the Novell Linux Point of Service CDs are available in the distribution directories before you try to build an image.

Upon completion of the copy procedure, POSCopyTool automatically verifies the source CDs were correctly copied to the distribution directory.

Use the following syntax to manually verify the source CDs with POSCDTool:

```
poscdtool.pl --verify [--source=distribution_directory]
[--imageclass=NLD|SLES]
```

For example, the following command verifies the both the SLES and NLD source files are available in the default distribution directory, `/opt/SLES/POS/dist/`:

```
poscdtool.pl --verify
```

The following command verifies only the NLD source files are available in `/opt/SLES/POS/dist/`:

```
poscdtool.pl --verify --imageclass=NLD
```

Building Images with the scr ImageBuilder Tool

8

This section reviews the commands, image components, and processes required to build Novell® Linux Point of Service client images with the scr ImageBuilder tool.

8.1 scr Commands

The basic command line that provides all features required to build client and boot images is:

`scr [options]`

Table 8-1 summarizes the available scr options. For examples of how these options are applied, see “Building Images with scr” on page 109.

NOTE: If an option has an abbreviated form, the abbreviation is indicated below the option.

Table 8-1 *scr command options*

Option	Description
`--build` `-b`	Used in conjunction with `--destdir`, this option builds an image. This process assumes the Image Description Tree has been previously prepared. For more information, see the `--prepare` option. Images are created with a time stamp in the filename. Old images are kept on the server. **IMPORTANT:** scr only maintains five builds of a single image in the same directory. When you generate the sixth build of an image, scr deletes the oldest image version. (scr determines the oldest image version by the image date.) If you want to maintain more than five versions of a single image, you must maintain them in separate directories. For sample usage, see “Building Images with scr” on page 109.

Option	Description
`--create` *image_name-version* `-c` *image_name-version*	Used in conjunction with `--image`, this option clones an existing Image Description Tree. The new Image Description Tree is created at /opt/SLES/POS/system/*image_name-version/*. The name of the new Image Description Tree designated with the `--create` option must include the *image_name* and *version*. If you want to change the version number of your cloned Image Description Tree, you must edit the VERSION file located in the root of the Image Description Tree. The scr tool does not list the correct version number if you only modify the version included in the directory name. For sample usage, see “Cloning the Image Description Tree” on page 109.
`--create-data-image` *directory*	Used in conjunction with `--image` and `--destdir`, this option creates a data-only image. A data-only image is an ext2 image file containing only a copy of the Image Description Tree starting at the given directory. This kind of image cannot be used as operating system or boot image. If a disk-based system is booting and the IMAGE variable in the `config.`*MAC_address* file includes an additional data image that will be downloaded to a /dev/ram*x* device, the data contents are automatically included into the system. If a data image is downloaded into a partition on the disk, the data is available at the mount point referring to the contents of the PART variable. An advantage of this feature compared to the normal CONF workflow is that the data image is controlled in the same way as the client image, which means that any changes to the data image are detected automatically and the image is updated if necessary. Images are created with a time stamp in the filename. Old images are kept on the server. For sample usage, see “Using Data Images to Manage External Configuration Files” on page 113.
`--create-iso`	Used in conjunction with `--destdir`, this option creates an ISO image from a previously prepared root image tree. For sample usage, see Section 10.1.5, “Creating the CD ISO Image,” on page 175.
`--destdir` *directory* `-d` *directory*	Designates the destination directory for the image and the checksum file. For sample usage, see Section 8.4.4, “Building the Image,” on page 116.
`--export-config`	Exports the tarball included in the image with the `--import-config` option. The tarball contains the Image Description Tree and command line used to build the image.

Option		Description
`--extend` *`setup_file`*		Used in conjunction with `--prepare`, this option extends the image. It uses the *setup_file* to install additional RPM packages that are not part of any distribution. The *setup_file* indicates additional RPMs with the following specifications: ◆ **Package description:** A line in the *setup_file* that indicates what the package is called, which RPM options must be used to install it, which version of the package should be used, and in which directory the package is located. If no directory is indicated, the system searches for the package in the package directories designated in AdminServer.conf. ◆ **Config:** Following the optional keyword config, the name of an RPM appears. The package is unpacked to the files-user directory with cpio. For sample usage, see "Adding a Package to a Custom File" on page 111.
`--feature` *`list`* `-f` *`list`*		Used in conjunction with `--prepare`, this option defines features to include in the image after it has been prepared. You can list one or more of the following features in a comma-separated list.

	`adduser:`*`username`*
	`[+`*`group_name`*`]`
	`[+nohome]=[`*`password`*`]:`
		Includes a user with a password in the image.
		
		◆ If the password is not provided, scr prompts for the user password during image preparation.
		
		◆ If an empty string is used, no password is set for the user.
		
		◆ If a group name is provided, the user is assigned to the group.
		
		◆ If the nohome flag is set, the user does not have a home directory.

	`addgroup:`*`group_name`*	Includes a group with a group password in the image. ◆ If no password is provided, scr prompts for the group password during image preparation. ◆ If an empty string is used, no password is set for the group.
	`auth`	Includes root authentication in the image. It requests a password for the root user during the generation of the file system image. The encrypted password is then entered in the existing /etc/shadow.

Option		Description
	`boot_cd:config=`*`CD_setup_directory`*	Creates a CD bootable image. It requires the CD setup directory as a parameter. Use this option when generating the CDBoot image. For sample usage, see Section 10.1.4, “Generating the CDBoot Image,” on page 174.
	`serial_console`	Includes serial console support in the image. This option generates the corresponding files, inittab and security, and stores them in the files-user tree. For sample usage, see Section 8.4.4, “Building the Image,” on page 116.
	`set_serial`	Includes a run-level script called setserial in the image. This script enables a service to configure all available serial interfaces for raw access during boot. This is needed for Point of Service systems providing more than the standard /dev/ttyS0 and /dev/ttyS1 serial interfaces.
`--gzip` `-z`		Used in conjunction with `--build`, this option compresses the created image file using gzip.
`--help` `-h`		Lists all the scr command line options and their syntax.
`--image` *`image_name-version`* `-i` *`image_name-version`*		Defines the name of the Image Description Tree you want to prepare, build, or clone. The tree name consists of the image name and the version number, separated by a dash. For example, browser-2.0.21. For sample usage, see Section 8.4.4, “Building the Image,” on page 116.
`--import-config`		Includes a tarball in the image that contains the Image Description Tree and command line used to build the image.
`--keep-root`		Used in conjunction with `--prepare` or `--build`, this option maintains the root image tree. The root image tree is normally removed after an error or after the image is created using the `--prepare` option. This option prevents the root image tree from being deleted.
`--keep-rpm`		Used in conjunction with `--build`, this option maintains the RPM database. The RPM database is normally removed from the image to save space. This option prevents the RPM database from being deleted.

Option	Description
`--list`	Shows a list of all available image descriptions and versions from which an image can be built. To create an image from one of the listed items, use one of the scr build commands and specify the image name in the --image option. For sample usage, see Section 8.4.4, “Building the Image,” on page 116.
`--logfile`	Used in conjunction with `--prepare`, this option creates a log file of the image build process.

`--no-stripping` *file*	Maintains symbols in the image.
	
	Executables and libraries are normally stripped out to discard symbols and save space. If symbols are needed, this option can be used.
	
	◆ If you specify a filename, only the matching files are not stripped.
	◆ If you do not specify a filename, nothing is stripped.
	◆ The `--no-stripping --preserve-dates` option preserves the date and time stamp while stripping.
	
	NOTE: You can use this command to prevent stripping of symbols in JRE components.
	
	The syntax of the *file* is based on glob patterns. Each line of the file specifies a glob pattern that can match exactly one file or multiple files. For example:
	`/usr/X11R6/bin/XFree86`
	`/lib/*`
	
	This command prevents the file /usr/X11R6/bin/XFree86 and all files within the directory /lib from being stripped.
	
	NOTE: Glob patterns don’t work recursively.

`--setenv` *environment_variable=value* `-s` *environment_variable=value*	Used in conjunction with the `--prepare` command, this option sets the environment variable. For example, scr --prepare --image image-2.0.3 --setenv SCR_BUILD_DIR=/tmp **NOTE:** The value of the SCR_BUILD_DIR environment variable creates an image root tree in the /tmp directory.

Option	Description
`--prepare` `-p`	Used in conjunction with `--image`, this option generates only the Image Description Tree; the file system image is not created. The resulting structure, the root image tree, can be manually modified. The root directory of the image is named root-*image_name-version* and is located in the current directory. For sample usage, see Section 8.4.4, “Building the Image,” on page 116.
`--unsetenv` *`environment_variable=value`* `-u` *`environment_variable=value`*	Used in conjunction with `--prepare`, this option unsets the environment variable. This option takes precedence over the `--setenv` option.
`--verify` `-V`	Used in conjunction with `--prepare`, this option verifies all RPM packages after they are installed. When finished, ImageBuilder displays the verification results. For sample usage, see Section 8.4.4, “Building the Image,” on page 116.
`--version`	Returns the ImageBuilder version number.

8.2 scr Image Building Components

scr builds images using the Image Description Tree and the `AdminServer.conf` file. The Image Description Tree and `AdminServer.conf` file contain files and directories that define the structure, scripts, configuration files, and other components required to build client images for Point of Service systems.

The following sections review the image components required to build images with scr:

- Section 8.2.1, “Image Description Tree,” on page 102
- Section 8.2.2, “AdminServer.conf,” on page 107

8.2.1 Image Description Tree

scr builds images using a specific file system directory structure known as the Image Description Tree. The Image Description Tree provides the structure, scripts, configuration files, and other components required to build client images for Point of Service systems. These components are stored under `/opt/SLES/POS/system/`*`image_name-version/`*.

Table 8-2 summarizes the Image Description Tree components required to build images with scr.

Table 8-2 *Required components of an Image Description Tree*

Component	Description
/opt/SLES/POS/system/*image_name-version*/IMAGE	An unformatted file that contains a brief description of the image and its function.

Component	Description
/opt/SLES/POS/system/*image_name-version*/VERSION	A file that contains the version number of the Image Description Tree, such as 1.1.2. If you want to change the version number of your Image Description Tree, you must edit the VERSION and the name of the Image Description Tree directory. If you only modify the version included in the directory, the scr tool does not list the correct version number. scr replaces the version specified in the VERSION file with the version specified on the command line when you clone the Image Description Tree.
/opt/SLES/POS/system/*image_name-version*/files/	A subdirectory that contains special files, directories, and scripts, This function of this directory is to ensure that the RPM is used as the package manager before any packages are installed in the image. The entire directory is copied to the root of the image tree using cp -a. This directory cannot contain any libraries or binary files. Any binaries and libraries required before the first RPM call must be extracted from the corresponding packages in advance.
/opt/SLES/POS/system/*image_name-version*/files-user/	A subdirectory that contains special files, directories, and scripts for adapting the image environment after the installation of all the image packages.
/opt/SLES/POS/system/*image_name-version*/package/	A subdirectory where ImageBuilder searches for RPM packages. The directory is automatically initialized depending on the entries in the ImageBuilder configuration file, /etc/opt/SLES/POS/AdminServer.conf. For more information, see Section 8.2.2, “AdminServer.conf,” on page 107. If there is no package directory, ImageBuilder creates a link to the global package directory (/opt/SLES/POS/pac/) and the links designated in the AdminServer.conf file.
/opt/SLES/POS/system/*image_name-version*/script/	A subdirectory that contains Bash scripts that are called after a package is installed, primarily to remove the parts of a package that are not needed for the Point of Service system. **IMPORTANT:** For these scripts to run, the script name must match the name of the RPM (without the version).

Component	Description
/opt/SLES/POS/system/*image_name-version*/config	A configuration file that indicates the image size, type, and base name. The structure of the file corresponds to the format Key: Value. The configuration file defines the following keys: ◆ **size:** *image_size* *Image_size* is defined as a number followed by M (megabyte) or G (gigabyte). The scr ImageBuilder tool automatically extends the image size if the specified configuration size value is too small. However, if the designated *image_size* value plus the additional space required to build the image is more than 100 MB, scr aborts with an error message. If the designated *image_size* is larger than the space required to build the image, scr does not reduce the image size. This is because, in some instances, the additional space might be required to run custom scripts included with the image. ◆ **type:** ext2|ext3 The image type is currently restricted to ext2 or ext3, although, if necessary, different formats are possible. If you have an existing ext2 image, you can change the file system by setting a flag in the scCashRegister or the scWorkstation objects rather than recreate the image. If ext3 is specified in either LDAP object, the Point of Service terminal extends the file system to ext3 when the image is deployed. ◆ **name:** *image_name* *Image_name* indicates the base name of the image. When the image is generated, the *image_name* is automatically expanded to include the version number and the date. The version number is extracted from the VERSION file. scr replaces the *image_name* with the name specified on the command line when you clone the Image Description Tree. ◆ **timezone:** *relative_path_to_time_zone* All time zone definitions are located in the /usr/share/zoneinfo directory. To specify which time zone you want to use in the image, enter the relative path to a specific time zone definition. For example, timezone:US/Mountain. The ImageBuilder uses this information to extract the corresponding time zone from the timezone package. The time zone value is then stored as /etc/localtime in the image. ◆ **imagetype:** diskful|diskless The value for this optional parameter is either *diskful* or *diskless*. If imagetype is not specified, the image is built with the original setup description. If diskful is set, all the packages required to handle the image on a hard disk are included in the setup description. If diskless is set, all unnecessary packages are removed from the setup description.

Component	Description

/opt/SLES/POS/system/*image_name-version*/config (*continued*)	In addition to the keys described above, it is also possible to specify script variables in the Key:Value format. All values entered in the configuration file are stored in the .profile file before executing the scripts in the script directory. Using the following command, the .profile file is created at the root of the installed image so it can be sourced from any script:
	`test -f /.profile \&\& . /.profile`
	The parameters of the configuration file are then available as variables in the script and can be processed appropriately.
	With the `test -f /.profile \&\& . /.profile` command, the following script variables may also be stored in the configuration file:
	◆ **usbdrivers:** *file_names*
	Contains a comma-separated list of filenames. The filenames are interpreted as USB driver names and correspondingly captured if they are contained in the kernel tree.
	◆ **netdrivers:** *file_names*
	Contains a comma-separated list of filenames. Every file is indicated relative to the directory /lib/modules/*version*/kernel/drivers/net. The names are interpreted as network drivers and captured if they are contained in the kernel tree.
	◆ **drivers:** *file_names*
	Contains a comma-separated list of filenames. Every file is indicated relative to the directory /lib/modules/*version*/kernel/. The names are interpreted as general driver names and captured if they are contained in the kernel tree.
	◆ **locale:** *locale_names*
	Contains a comma-separated list of valid locale names. The image only contains support for the given locales. This includes the glibc part as well as the X11 library. A list of valid locales can be obtained with the locale -a command.
	◆ **keytable:** ***console_keymap***
	Contains the name of the console keymap to use. The name corresponds to a map file stored below the path /usr/share/kbd/keymaps. In addition, the KEYTABLE variable within the /etc/sysconfig/keyboard file is set according to the keyboard mapping.
	A representation of the configuration file for the disknetboot-2.0.21 image description is shown below:
	`name:initrd-disknetboot`
	`size:15M`
	`type:ext2`
	`netdrivers:pcnet32.o,mii.o,natsemi.o,tulip/tulip.o`

Component	Description
/opt/SLES/POS/system.*image_name-version*/config.cleanup	An optional configuration script for the image. This script is called at the end of the installation and after all the installation scripts have run. It is designed to clean up the image system. The target programs and files are those needed only while the installation scripts are running.
/opt/SLES/POS/system.*image_name-version*/config.system	An optional configuration script for the image. This script is called at the end of the installation but before the installation scripts have run. It is designed to configure the image system, such as the activation or deactivation of certain services (insserv). The call is not made until after the switch to the image has been made with chroot. **IMPORTANT:** This file provides the scripts required to install the image. In most instances, we recommend that you do not modify this file.
/opt/SLES/POS/system/*image_name-version*/setup	A configuration file that indicates which packages make up the image and which RPM options must be used to install them. Each package can also be accompanied by a specific version of the package. The structure of the file is as follows: `package_basename : RPM_option : package_version` Multiple RPM options are separated from each other by commas. If an executable shell script with the same name as the package base name is present in the script directory, it is executed after the installation of all the packages. For an example of a sample setup file, see Section C.1, “Sample setup File,” on page 235.
/opt/SLES/POS/system/*image_name-version*/setup.user	An optional configuration file that can be present in addition to setup. The file has the same format as the setup file, but a path to the package can be indicated after the package version. The structure of the file is as follows: `Package Basename : RPM Option : Package Version : Path` For an example of a sample setup.user file, see Section C.2, “Sample setup.user File,” on page 237.

Component	Description
/opt/SLES/POS/system/*image_name-version*/setup.txt	An optional information file for the LDAP system. This file contains information regarding which configuration files are required by the image and whether they are hardware or system--dependent. The structure of the file is as follows: *flag* : *configuration_file_name* The *configuration_file_name* includes the full file path. The following values can be set for the *flag* value: ◆ **SYS** specifies that the configuration file is a hardware-independent, such as /etc/ntp.conf. ◆ **HWD** specifies that the configuration file is hardware-dependent, such as /etc/X11/XF86Config.

8.2.2 AdminServer.conf

When scr generates an image, it searches the paths listed in the `AdminServer.conf` file to find the RPM packages required to create the image. This ASCII, line-based file is located at `etc/opt/SLES/POS/`. It provides the paths to the distribution directories where you have copied the Novell Linux Point of Service CDs.

The `AdminServer.conf` file also references the maintenance directory. The maintenance directory is essentially an "override" directory. RPMs located in this directory take precedence over RPMs located in the distribution directories. You can add any RPM to this directory that you want scr to use in lieu of the default RPMs in the distribution directories. By default, the maintenance directory contains the glibc and devs RPMs. For a detailed breakdown of the maintenance directory structure, see Section B.1, "Administration Server Directory Structure," on page 211.

Information is organized in simple key=value format where the value after the key indicates the path to an Novell Linux Point of Service CD. For example, the following is the `AdminServer.conf` file for an NLD distribution:

```
SLESCD0=/opt/SLES/POS/maintenance/nld
SLESCD1=/opt/SLES/POS/dist/NLD9/SP1/CD1
SLESCD2=/opt/SLES/POS/dist/NLD9/SP1/CD2
SLESCD3=/opt/SLES/POS/dist/NLD9/FCS/CD1
SLESCD4=/opt/SLES/POS/dist/NLD9/FCS/CD2
SLESCD5=/opt/SLES/POS/dist/NLD9/FCS/CD3
SLESCD6=/opt/SLES/POS/dist/NLPOS9/FCS/CD4
```

The `AdminServer.conf` file for a SLES distribution appears as follows:

```
SLESCD0=/opt/SLES/POS/maintenance/sles
SLESCD1=/opt/SLES/POS/dist/NLPOS9/FCS/CD1
SLESCD2=/opt/SLES/POS/dist/NLPOS9/FCS/CD2
SLESCD3=/opt/SLES/POS/dist/NLPOS9/FCS/CD4
SLESCD4=/opt/SLES/POS/dist/SLES9/FCS/CD1
SLESCD5=/opt/SLES/POS/dist/SLES9/FCS/CD2
SLESCD6=/opt/SLES/POS/dist/SLES9/FCS/CD3
```

The order in which the CDs are listed is important because when scr generates the image, it searches each CD in the designated order for RPMs listed in the setup file. For example, given the preceding AdminServer.conf file, ImageBuilder searches for RPM packages beginning with `/opt/SLES/POS/maintenance/nld` and ending with `/opt/SLES/POS/dist/NLPOS9/FCS/CD4`. Note that the maintenance directory is listed first so scr uses any RPM it finds in this directory before it searches the distribution directories.

The POSCDTool is used to generate the `AdminServer.conf` file. For more information on this process, see Section 7.3.4, "Generating AdminServer.conf or Distribution.xml," on page 94.

NOTE: If you receive a newer SLES9 Service Pack CD, run the poscdtool utility to regenerate the `AdminServer.conf` file.

8.3 Getting Ready to Build Images with scr

Before you can build client images with scr, you must complete the following tasks:

1. Install ImageBuilder and the Image Description Trees.
2. Copy the image source files from the Novell Linux Point of Service CDs to a central distribution directory.
3. Define the location of the image source files.

These steps are explained in the following sections.

8.3.1 Installing ImageBuilder and Image Templates

ImageBuilder and the corresponding Image Description Trees are installed when you select the image building utilities during the Administration Server installation. For further information on creating an image server, see "Setting Up the Administration Server" or "Setting Up a Dedicated Image Building Server" in the *Novell Linux Point of Service 9 Installation Guide*.

During installation of the image server, the following image building components are installed:

- All Administration and Branch Server packages are installed to the server.
- The ImageBuilder packages (scr and xscr) are installed to the `/usr/bin/` directory.
- The Image Description Trees for each image are installed to `/opt/SLES/POS/system/image_name-version/`. For information on scr Image Description Trees, see Section 8.2.1, "Image Description Tree," on page 102.
- The default configuration information for all kernel drivers are installed to `/opt/SLES/POS/system/templates/drivers/`.

8.3.2 Copying the Novell Linux Point of Service CDs to a Central Distribution Directory

To build the client images, ImageBuilder must have access to the source RPMs. Therefore, before building client images, you must copy the source files on the Novell Linux Point of Service CDs to a central distribution directory.

The POSCDTool and POSCopyTool utilities included with Novell Linux Point of Service copy the RPMs required to build client images. For information on this procedure, see Section 7.3.1, “Copying the Novell Linux Point of Service CDs,” on page 92.

8.3.3 Defining the Location of the Image Source Files

When ImageBuilder builds an image, it must know where it can locate the RPMs required to build the image.

For scr, the path to the RPM packages is defined in the ImageBuilder configuration file, `/etc/opt/SLES/POS/AdminServer.conf`. This ASCII file provides the CD paths to the installation source tree where you have copied the Novell Linux Point of Service CDs.

When scr generates an image, it searches the paths listed in the AdminServer.conf file to find the packages required to create the requested image. The order of the single CD entries is important because scr looks for the requested package in the same order as the CD specification in `AdminServer.conf`.

For more information, see Section 8.2.2, “AdminServer.conf,” on page 107. For information on creating this file, see Section 7.3.4, “Generating AdminServer.conf or Distribution.xml,” on page 94.

8.4 Building Images with scr

After you have installed ImageBuilder and the image templates, copied the image source files to a distribution directory, and defined the image source location files, you can start building Point of Services images.

The process required to build an image with scr is as follows:

1. Clone the Image Description Tree.
2. Add software packages or add-on options to an image.
3. Configure the image.
4. Build the image.

These steps are explained in the following sections.

8.4.1 Cloning the Image Description Tree

scr builds images using a specific file system directory structure known as the Image Description Tree. The Image Description Tree provides the structure, scripts, configuration files, and other components required to build images for Point of Service systems.

You can use the default Image Description Trees provided with Novell Linux Point of Service to generate the DiskNetBoot, CDBoot, Minimal, Browser, Java, and Desktop images. However, to maintain a standardized source tree and simplify the upgrade process, it is recommended that you maintain the default Image Description Trees provided with Novell Linux Point of Service as master copies. To build your own images, you can clone the default Image Description Trees, then modify the cloned tree.

NOTE: To view a list of available Image Description Trees, execute the `scr --list` command.

The basic syntax to clone an Image Description Tree is as follows:

```
scr --create image_name-version --image image_name-version
```

For example, the following command clones the Minimal-2.0.21 Image Description Tree to create a new Image Description Tree named myImage-1.1.1:

```
scr --create myImage-1.1.1 --image minimal-2.0.21
```

IMPORTANT: You cannot use the word “boot” in any image name other than the cdboot and disknetboot images.

The new Image Description Tree is located at `/opt/SLES/POS/system/myImage-1.1.1`. You can then modify the cloned Image Description Tree as required to create your new image. For a description of the individual Image Description Tree components, see Section 8.2.1, “Image Description Tree,” on page 102 and Appendix B, “Novell Linux Point of Service Files and Directory Structure,” on page 211.

8.4.2 Adding Software Packages or Add-on Options to an Image

Extending an image is the process used to add software packages or add-on options to an image. You can extend images with software packages included within the Novell Linux Point of Service CD set as well as packages that are not included within the Novell Linux Point of Service CD set. You can also extend images with unpackaged software.

To extend client images with software packages that are included within the Novell Linux Point of Service CD set such as the client image add-on options, you simply add the package to the list of packages marked for installation. For example, to extend the Minimal image to provide the Samba 3 client, it is only necessary to add the package to the list of packages marked for installation. This can be done in two ways:

- Add the package to the setup file, which can be found in the Image Description Tree. After this, adapt the size parameter of configuration file, which can be found in the description tree. For information on this procedure, see “Adding a Package to the Setup File” on page 111.
- Create a custom setup file using the same syntax as the setup file and add the package to it. The size parameter can also be part of this file. Specify the file as an argument of the scr `--extend` option. For information on this procedure, see “Adding a Package to a Custom File” on page 111.

To extend client images with software packages not included within the Novell Linux Point of Service CD set, you must add those packages to directory ImageBuilder uses to build the image, list the package in the setup file, and adapt the size parameter of configuration file. For information on this procedure, see “Extending Images with Non-Standard Packages” on page 111

To extend client images with unpackaged software, you must prepare the Image Description Tree, install the unpackaged software within the image, then build the image. For information on this procedure, see “Extending an Image with Unpackaged Software” on page 112.

Adding a Package to the Setup File

The following instructions illustrate how to add the vim package file to the myImage-1.1.1 setup file:

1. Create a copy of the standard Minimal Image Description Tree:
   ```
   scr --create myImage-1.1.1 --image minimal-2.0.21
   ```
2. Add the following line to `/opt/SLES/POS/system/myImage-1.1.1/setup`:
   ```
   vim : x : x
   ```
3. Adapt the size parameter of the `/opt/SLES/POS/system/myImage-1.1.1/config` file.
   ```
   size:42M
   ```

   IMPORTANT: You must modify the size parameter in the image config file to reflect the new image size because the scr ImageBuilder tool requires accurate image size information to generate the image.
4. You can then build the new image.

   For information on this procedure, see Section 8.4.4, “Building the Image,” on page 116.

Adding a Package to a Custom File

The following instructions illustrate how to add the vim package to a file with the same syntax as the setup file:

1. Create a copy of the standard Minimal Image Description Tree:
   ```
   scr --create myImage-1.1.1 --image minimal-2.0.21
   ```
2. Create the `/tmp/setup.with.vim` file and add the following lines:
   ```
   size:42M
   vim : x : x
   ```
   The specification of packages in this file requires the single packages to exist at `/opt/SLES/POS/pac`. If your package resides elsewhere, specify the path at the end of the line, for example, `vim : x : x : /tmp/editors`.
3. You can then build the new image.

   For information on this procedure, see Section 8.4.4, “Building the Image,” on page 116.

Extending Images with Non-Standard Packages

The following instructions illustrate how to add an alternate vim package (that is, a vim package not included with Novell Linux Point of Service CD set) to myImage-1.1.1:

1. Create a copy of the standard Minimal image:
   ```
   scr --create myImage-1.1.1 --image minimal-2.0.21
   ```
2. Copy the vim package to the global package directory, `/opt/SLES/POS/pac`.
   ```
   cp vim-other.rpm /opt/SLES/POS/pac
   ```
3. Add the following line to `/opt/SLES/POS/system/myImage-1.1.1/setup.user`:
   ```
   vim-other : x : x
   ```

The example uses the package name vim-nstandard.rpm. If you use package names with version numbers, for example, `vim-other-1.3-471.i586.rpm`, you must add the following line in `setup.user`:

```
vim-other : x : 1.3-471.
```

4 Adapt the size parameter of the `/opt/SLES/POS/system/myImage-1.1.1/config` file.

```
size:42M
```

IMPORTANT: You must modify the size parameter in the image config file to reflect the new image size because the scr ImageBuilder tool requires accurate image size information to generate the image.

5 You can then build the new image.

For information on this procedure, see Section 8.4.4, “Building the Image,” on page 116.

Extending an Image with Unpackaged Software

The following instructions illustrate how to add software not packaged into an RPM package to myImage-1.1.1:

1 Create a copy of the standard Minimal image:

```
scr --create myImage-1.1.1 --image minimal-2.0.21
```

2 Prepare the image:

```
scr --prepare --image myImage-1.1.1
```

3 After the image is prepared, find the root system of the image below the directory root-myImage-1.1.1.

4 Copy the non-RPM software to a directory within the image. For example:

```
cp software root-myImage-1.1.1/tmp
```

5 Change to the image system with the command:

```
chroot root-myImage-1.1.1 bash
```

6 Perform all the steps needed to install the software.

7 Exit the image system with the exit command.

8 Adapt the size parameter of the `/opt/SLES/POS/system/myImage-1.1.1/config` file.

```
size:42M
```

IMPORTANT: You must modify the size parameter in the image config file to reflect the new image size because the scr ImageBuilder tool requires accurate image size information to generate the image.

9 You can then build the new image.

For information on this procedure, see Section 8.4.4, “Building the Image,” on page 116.

8.4.3 Configuring the Image

Configuring an image means adapting it for a specific hardware environment. This includes activating and deactivating services, setting up special POStinstall scripts, adding standard configuration files and setting the time zone.

The following sections review these image configuration options.

Setting the Time Zone

Time zones are set in the config file.

`timezone:`*`relative_path_to_time_zone`*

All time zone definitions are located in the `/usr/share/zoneinfo` directory. To specify which time zone you want to use in the image, enter the relative path to a specific time zone definition. For example, `timezone:US/Mountain`. The ImageBuilder uses this information to extract the corresponding time zone from the timezone package. The time zone value is then stored as `/etc/localtime` in the image.

For more information on the config file, see Section 8.2.1, "Image Description Tree," on page 102.

Including Fixed Configuration Files

A fixed configuration is a configuration file that provides information for a service that is hardware independent. Fixed configuration files are stored in the Image Description Tree under the files-user subdirectory.

The following instructions illustrate how to add the fixed configuration file, `/etc/sysconfig/hotplug`, to the Image Description Tree, `/opt/SLES/POS/system/myImage-1.1.1/`:

1. Go to the `/opt/SLES/POS/system/`*`image_name-version`*`/files-user` directory:

   ```
   cd /opt/SLES/POS/system/myImage-1.1.1/files-user
   ```

2. Within the `files-user` directory, create a directory structure that parallels the original system location of the configuration file:

   ```
   mkdir -p etc/sysconfig
   ```

3. Copy the configuration file to the appropriate directory within the `files-user` tree.

   In this case, simply copy the hotplug file to the `/opt/SLES/POS/system/`*`image_name-version`*`/files-user/etc/sysconfig/` directory:

   ```
   cp /etc/sysconfig/hotplug etc/sysconfig
   ```

The file tree within files-user is completely copied to the image when it is generated. For more information on the `files-user` directory, see Section 8.2.1, "Image Description Tree," on page 102

Using Data Images to Manage External Configuration Files

A data-only image is an ext2 image file that contains only a copy of the Image Description Tree starting at the given directory. This kind of image cannot be used as operating system or boot image. However, it can be used to add external configuration files to a Point of Service terminal.

If a disk-based system is booting and the IMAGE parameter in the `config.MAC_address` file includes an additional data image that will be downloaded to a /dev/ram*x* device, the data contents are automatically included into the system. If a data image is downloaded into a partition on the disk, the data is available at the mount point referring to the contents of the PART variable.

The advantage of using data images to add external configuration files to a Point of Service terminal is that the data image is controlled in the same way as the client image. This means you can manage the configuration files independent of the client image.

IMPORTANT: To implement this functionality, you must manually modify the `config.MAC_address` file for each Point of Service terminal that you want to load the data image. However, when you run `posAdmin --updateconfig` or `posldap2crconfig.pl --dumpall` to refresh the `config.MAC_address` files on the Branch Server, these modifications are overwritten. Therefore, to maintain the functionality, you must manually reconfigure the `config.MAC_address` files each time you regenerate the files.

For more information on the `posldap2crconfig.pl` command, see Section A.3.5, "posldap2crconfig.pl," on page 206.

For more information on the `posAdmin --updateconfig` command, see Section 6.9, "Updating config.MAC_address and Hardware Configuration Files," on page 85.

The following instructions illustrate how to manage external configuration files with a data image:

1. Create a temporary directory that contains the data.

   ```
   mkdir /tmp/mydata
   ```

2. Create the directory structure according to the original system location of the configuration file below this data directory and apply your configurations.

   ```
   mkdir -p /tmp/mydata/etc/X11
   vi /tmp/mydata/etc/X11/XF86Config
   ```

3. Create a data image.

   ```
   xscr --create-data-image /tmp/mydata \
       --image mydata-2.0.21 --destdir /tmp/myDataDirectory
   ```

   This call creates the data image, mydata-2.0.21, and the referring MD5 sum in `/tmp/myDataDirectory/`.

4. Copy the image to the `/opt/SLES/POS/rsync/image/` directory on the Administration Server.

   IMPORTANT: The data image must be copied to the `/opt/SLES/POS/rsync/image` directory on the Administration Server before the Branch Server can distribute it to Point of Service terminals.

   ---

5. To activate the data image, add the data image to the IMAGE parameter in the `config.MAC_Address` file.

   The IMAGE entry might appear as follows:

   ```
   IMAGE=/dev/hda2;minimal;1.1.8;192.168.100.1;1024,
         /dev/ram2;mydata;2.0.21;192.168.100.1;1024
   ```

   IMPORTANT: To ensure the contents of the data image are copied to the system, the image must be downloaded to a /dev/ram*x* device.

With the data image listed as an IMAGE entry in the `config.MAC_address` file, the data image contents are copied to the Point of Service terminal after the data image has been downloaded to /dev/ram2.

To update the data image on the Point of Service terminal, you must perform the following:

1. Generate a new version of the data image.
2. Copy the new data image version to the `/opt/SLES/POS/rsync/image/` directory on the Administration Server.
3. Run `possyncimages.pl` to download the image to the Branch Server.
4. Modify the IMAGE entry in the `config.MAC_Address` file to reflect the data image's new version number.

Enabling DMA on Point of Service Terminal CD Drives

Setting up a Direct Memory Access (DMA) channel for the CD drive on your Point of Service terminals speeds up the process of booting and loading an image from CD. The CDBoot image template provided with Novell Linux Point of Service includes the RPM package (hdparm) required to enable DMA so that the DMA channel is configured when the terminal boots from CD. However, if you would like DMA to be enabled beyond the initial install, you must add the DMA feature to the client image.

To add DMA functionality to a client image:

1. Include the hdparm package in the image's `opt/SLES/POS/system/image_name-version/setup` file as follows:

   `hdparm : RPM_Option : Package_Version`

   NOTE: The CDBoot Image Specification Document includes the hdparm RPM package by default.

   ---

2. Add the CD device (usually /dev/hdc) to the DEVICES_FORCE_IDE_DMAflag in the `/etc/sysconfig/ide` file. For example:

   `DEVICES_FORCE_IDE_DMA="/dev/hdc:on"`

3. Provide a way for the `/etc/sysconfig/ide` file to be deployed on the Point of Service terminal.

   This can be accomplished in one of two ways:

   - Add the `/etc/sysconfig/ide` file to the `/opt/SLES/POS/system/image_name-version/files-user/` directory in the Image Description Tree.
   - Create an scConfigFileTemplate or scConfigFileSyncTemplate object under the scPosImage object associated with this image or under the scCashRegister object associated with the Point of Service terminals that use this image. For more information on this procedure, see Section 6.4.2, "Adding an scConfigFileTemplate Object," on page 76 or Section 6.4.3, "Adding an scConfigFileSyncTemplate Object," on page 77.

4. Build the image.

   For more information on this procedure, see Section 8.4.4, "Building the Image," on page 116.

Activating and Deactivating System Services

System services are activated or deactivated in the config.system file by using the insserv command to set or remove links.

To activate a service, add the following call to the `config.system` file:

```
sbin/insserv /etc/init.d/service
```

To deactivate a service, add the following call to the `config.system` file:

```
sbin/insserv -r /etc/init.d/service
```

For more information on the `config.system` file, see Section 8.2.1, “Image Description Tree,” on page 102.

Writing Post-Install Scripts

A Post-install script is always bound to a package from the setup file and is usually used to remove items from the package that are not needed for the image. This type of script must have the same name as the corresponding package and is stored in the script directory of the Image Description Tree (`opt/SLES/POS/system/image_name-version/script/`). The script itself is called within the image environment, which means it is not possible to damage the host system with your script even if you are using absolute paths.

A Post-install script uses the following format:

```
#!/bin/sh
echo -n "Image [image_name_version]..."
test -f /.profile \&\& . /.profile

... script code

echo done
```

image_name-version is the name of the image to which this script belongs.

For more information on the script directory, see Section 8.2.1, “Image Description Tree,” on page 102.

8.4.4 Building the Image

To get a list of available Image Description Trees, execute the following command:

```
scr --list
```

The output appears as follows:

```
28-Jul 17:52:52 <1> : Image: browser          Version: 2.0.21
28-Jul 17:52:52 <1> : Image: minimal          Version: 2.0.21
28-Jul 17:52:52 <1> : Image: disknetboot      Version: 2.0.21
28-Jul 17:52:52 <1> : Image: java             Version: 2.0.21
28-Jul 17:52:52 <1> : Image: cdboot           Version: 2.0.21
28-Jul 17:52:52 <1> : Image: desktop          Version: 2.0.21
```

NOTE: The `--list` option does not validate the Image Description Tree. You can build an image from the listed description trees only if they are complete.

To generate an image from one of the available Image Description Trees, execute the scr command with the `--prepare` and `--build` options. For example, the following scr command creates the standard Minimal image with the version 2.0.21 in the working directory myImages and verifies the RPM packages:

```
scr --prepare --image minimal-2.0.21
    --build --destdir myImages --verify
```

The following command illustrates how to enable support for the serial console using the --feature option for the Minimal image:

```
scr --prepare --image minimal-2.0.21
    --build --destdir compiled --feature serial_console
```

IMPORTANT: scr only maintains five builds of a single image in the same directory. When you generate the sixth build of an image, scr deletes the oldest image version. (xscr determines the oldest image version by the image date.) If you want to maintain more than five versions of a single image, you must maintain them in separate directories.

8.5 Distributing Images

To electronically distribute new or updated client images, you must first copy the images into the central RSYNC directory of the Administration Server and then transfer the images to the Branch Servers.

This section reviews each step in the electronic distribution process.

- Section 8.5.1, “Copying Images to the Administration Server RSYNC Directory,” on page 117
- Section 8.5.2, “Distributing Images to the Branch Server,” on page 118
- Section 8.5.3, “Distributing Images to Point of Service Terminals,” on page 119
- Section 8.5.4, “Image Install Notification,” on page 119

NOTE: If you are unable to electronically distribute Point of Service images over your network, you must manually distribute the images uses CDBoot images. For more information on creating a CDBoot image, see Section 10.1, “Building a CDBoot Image,” on page 171.

8.5.1 Copying Images to the Administration Server RSYNC Directory

The first step to distribute new client images is to copy the images from the `/opt/SLES/POS/image` directory to the RSYNC directory, `/opt/SLES/POS/rsync/`.Client images must be located in the `/opt/SLES/POS/rsync/image/` directory on the Administration Server before the RSYNC service can transmit the images to the Branch Server. The boot image must be located in `/opt/SLES/POS/rsync/boot/`.

NOTE: Copying the client images to the RSYNC directory is done manually to control which client image types and versions are distributed to the Branch Servers.

Copying Client Images to the Administration Server's RSYNC Directory

The following example demonstrates how to put a previously extended Browser client image in the Administration Server's RSYNC directory so it can be received, on request, by the Branch Server:

1. Copy the extended Browser client image:

```
cp /opt/SLES/POS/image/myBrowser-2.0.21-2004-12-05 \
   /opt/SLES/POS/rsync/image/browser-2.0.21
```

2. Copy the corresponding Browser image MD5 checksum file:

```
cp /opt/SLES/POS/image/myBrowser-2.0.21-2004-12-05.md5 \
   /opt/SLES/POS/rsync/image/myBrowser-2.0.21.md5
```

Copying Boot Images to the Administration Server's RSYNC Directory

The following example demonstrates how to copy the first and second stage boot images to the Administrations Server's RSYNC directory so they can be received, on request, by the Branch Server:

NOTE: Point of Service terminals boot two images, a first stage image (`initrd.gz`) and a second stage image (`linux`). For more information, see Section 3.6, "Booting the Point of Service Terminal," on page 35.

---

1. Copy the initrd-disknetboot image as `initrd.gz`:

```
cp /opt/SLES/POS/image/initrd-disknetboot-version-date.gz
   /opt/SLES/POS/rsync/boot/initrd.gz
```

2. Copy the kernel image as `linux`:

```
cp /opt/SLES/POS/image/initrd-disknetboot-version-
date.kernel.kernel_version /opt/SLES/POS/rsync/boot/linux
```

8.5.2 Distributing Images to the Branch Server

If you create a new image or change an image version, you can run the `possyncimages.pl` script at the Branch Server to transfer new or updated images to the Branch Server after the images are in the Administration Server's RSYNC directory.

IMPORTANT: The RSYNC service must be properly configured and running on the Administration Server for the possynimages.pl script to run. For more information, see Section 6.3.3, "Adding an scServerContainer and scBranchServer Object," on page 69.

Additionally, each client image must have an associated scPosImage object in LDAP and the object's scPosImageVersion attribute must be set to active before possyncimages.pl will transfer the images to the Branch Server. For more information, see Section 6.5.2, "Activating Images," on page 81.

The basic process is as follows:

1. The `possyncimages.pl` script initially checks via PID file to determine if an instance is already running.
2. The image files are then copied from the Administration Server to the Branch Server. Boot images are copied from the `/opt/SLES/POS/rsync/boot/` directory on the Administration Server to the `/tftpboot/boot/` directory on the Branch Server. Client

images and their associated MD5 checksum files are copied from /opt/SLES/POS/rsync/image/ to /tftpboot/image/.

During this process, the TFTP server must be stopped or otherwise prevented from transmitting the image files to clients.

For more information on the possyncimages.pl script, see Section A.3.10, "possyncimages.pl," on page 208.

IMPORTANT: Remember that distributing client images from the Administration Server to the Branch Servers is only one part of the process required to deploy new versions of a client image. You must also update the scPosImageVersion attribute within the Image Reference object (scPosImage) in the LDAP tree. Otherwise Point of Service terminals already registered in LDAP cannot boot the new client image version. For more details, refer to Section 6.5, "Managing Image Objects," on page 79 and Section A.3.5, "posldap2crconfig.pl," on page 206.For an illustration of Novell Linux Point of Service system dependences, see Section 1.2, "Dependencies Between LDAP, Branch Server, and Point of Service Terminal," on page 15.

After executing the possyncimages.pl script, verify the result by checking the contents of the following directories:

- /tftpboot/image
- /tftpboot/boot

8.5.3 Distributing Images to Point of Service Terminals

New or updated images are distributed to Point of Service terminals at boot time. For information on this process, see Section 3.6, "Booting the Point of Service Terminal," on page 35.

8.5.4 Image Install Notification

When the Branch Server distributes a new image to a Point of Service terminal, the system provides notification that the image was successfully installed on the Point of Service terminal. The notification is stored in the scWorkstation object in the LDAP directory on the Administration Server.

When the image is successfully installed on the Point of Service terminal, the linuxrc script running on the Point of Service terminal creates the *bootversion.MAC_Address* file in the /tftpboot/upload directory on the Branch Server. posleases2ldap then transfers the information to the scNotifiedimage attribute in the scWorkstation object in LDAP and deletes the *bootversion.MAC_address* file.

Building Images with the xscr ImageBuilder Tool

9

This section reviews the commands, image components, and processes required to build Novell® Linux Point of Service client images with the xscr ImageBuilder tool.

9.1 xscr Commands

The basic command line that provides all features required to build client and boot images is:

`xscr [options]`

Table 9-1 summarizes the available xscr command line options. For examples of how these options are applied, see “Building Images with xscr” on page 144.

NOTE: If an option has an abbreviated form, the abbreviation is indicated below the option.

Table 9-1 *xscr command options*

Option	Description
`--build` `-b`	Used in conjunction with `--destdir`, this option builds an image. This process assumes the Image Description Tree has been previously prepared. For more information, see the `--prepare` option. Images are created with a time stamp in the filename. Old images are kept on the server. **IMPORTANT:** xscr maintains up to five builds of a single image in the same directory. When you generate the sixth build of an image, xscr deletes the oldest image version. (xscr determines the oldest image version by the image date.) If you want to maintain more than five versions of a single image, you must maintain them in separate directories. For sample usage, see “Building the Image” on page 162.

Option	Description
`--config` *filename*	Specifies an admin file to use instead of the default file.
`--create` *image_name-version* `-c` *image_name-version*	Used in conjunction with `--image` and `--dist nld|sles`, this option clones an existing Image Description Tree. The new Image Description Tree is created at /opt/SLES/POS/system/*image_name-version/*. The name of the new Image Description Tree must include the *image_name* and *version*. If you want to change the version number of your cloned Image Description Tree, you must edit the Version attribute in the ImageSpecification element within the ImageSpecification.xml file located in the root of the Image Description Tree. The xscr tool does not list the correct version number if you only modify the version included in the directory name. For sample usage, see Section 9.4.1, “Cloning the Image Description Tree,” on page 144.
`--create-data-image` *directory*	Used in conjunction with `--image` and `--destdir`, this option creates a data-only image. A data-only image is an ext2 image file containing only a copy of the Image Description Tree starting at the given directory. This kind of image cannot be used as operating system or boot image. If a diskful system is booting and its IMAGE variable in the config.*MAC_address* file includes an additional data image that will be downloaded to a /dev/ram*x* device, the data contents are automatically included into the system. If a data image is downloaded into a partition on the disk, the data is available at the mount point referring to the contents of the PART variable. An advantage of this feature compared to the normal CONF workflow is that the data image is controlled in the same way as the client image, which means that any changes to the data image are detected automatically and the image is updated if necessary. Images are created with a time stamp in the filename. Old images are kept on the server. For sample usage, see “Using Data Images to Manage External Configuration Files” on page 159.
`--create-iso` *name*	Used in conjunction with `--destdir`, this option creates an ISO image from a previously prepared root image tree. For sample usage, see Section 10.1.5, “Creating the CD ISO Image,” on page 175.
`--delta`	Creates a delta image by comparing two images and stores the compressed delta image and checksum file in the specified destination directory. For sample usage, see Section 9.6.1, “Creating the Delta Image File,” on page 165.

Option	Description
`--destdir` *directory* `-d` *directory*	Designates the destination directory for the image and the checksum file. For sample usage, see Section 9.4.4, “Building the Image,” on page 162.
`--dist nld|sles`	Used in conjunction with `--create`, this option defines what type of image the ImageBuilder tool generates. If the distribution is NLD, ImageBuilder generates the image with NLD RPMs. If the distribution is SLES, ImageBuilder generates the image with SLES RPMs. For more information, see Section 9.4.1, “Cloning the Image Description Tree,” on page 144. **NOTE:** NLD is the default distribution type. Therefore, if you do not specify the image distribution when cloning the Image Description Tree, xscr defaults to NLD. In general, most Point of Service images are created using the NLD distribution. The only images that require the SLES distribution are POSBranch images. For more information, see Section 10.2, “Building POSBranch Images,” on page 176.
`--export-config` *image_name*	Exports the tarball included in the image with the `--import-config` command. The tarball contains the Image Description Tree and command line used to build the client image.
`--feature` *list* `-f` *list*	Used in conjunction with `--prepare`, this option defines features to include in the image after it has been prepared. You can list one or more of the following features in a comma-separated list: ◆ `boot_cd:config=`*CD_setup_directory* This option creates a CD bootable image. It requires the CD setup directory as a parameter. Use this option when generating the CDBoot image. For sample usage, see Section 10.1.4, “Generating the CDBoot Image,” on page 174. ◆ `serial_console` This option includes serial console support in the image and generates the corresponding files, inittab and security, and stores them in the files-user tree. For sample usage, see Section 9.4.4, “Building the Image,” on page 162. ◆ `set_serial` This option Includes a run-level script called setserial in the image. This script enables a service to configure all available serial interfaces for raw access during boot. This is needed for Point of Service systems providing more than the standard /dev/ttyS0 and /dev/ttyS1 serial interfaces.

Option	Description
`--gzip` `-z`	Used in conjunction with `--build`, this option compresses the created image file using gzip.
`--help` `-h`	Lists the xscr command line options and their syntax.
`--image` *`image_name-version`* `-i` *`image_name-version`*	Defines the name of the Image Description Tree you want to prepare, build, or clone. The name of the Image Description Tree consists of the image name and version number, separated by a dash; for example, browser-2.0.21. For sample usage, see Section 9.4.4, “Building the Image,” on page 162.
`--import-config`	Adds a tarball to the image that contains the Image Description Tree and command line used to build the client image.
`--keep-root`	Used in conjunction with `--prepare`, this option maintains the root image tree. The root image tree is normally removed after an error or after the image is created using the --build option. This option prevents the root image tree from being deleted.
`--keep-rpm`	Used in conjunction with `--build`, this option maintains the RPM database. The RPM database is normally removed from the image to save space. This option prevents the RPM database from being deleted.
`--list`	Lists the existing images on the server.
`--logfile` *`filename`*	Used in conjunction with `--prepare`, this option creates a log file of the image build process.
`--nostrict`	Checks the RPM package signature before installation. This command can be used in conjunction with the `--prepare` option. For sample usage, see Section 9.4.4, “Building the Image,” on page 162.

Option	Description

`--no-stripping` *`filename`*	Prevents xscr from stripping symbols from executables and libraries in the image.
	
	Executables and libraries are normally stripped out to discard symbols and save space. If symbols are needed, this option can be used.
	
	◆ If you specify a filename, only the matching files are not stripped.
	◆ If you do not specify a filename, nothing is stripped.
	◆ The `--no-stripping --preserve-dates` option preserves the date and time stamp while stripping.
	
	NOTE: You can use this command to prevent stripping of symbols in JRE components.
	
	The syntax of *filename* is based on glob patterns. Each line of the file specifies a glob pattern that can match exactly one file or multiple files.
	
	For example, the following pattern prevents the file `/usr/X11R6/bin/XFree86` and all files within the directory `/lib` from being stripped:
	`/usr/X11R6/bin/XFree86`
	`/lib/*`
	
	NOTE: Glob patterns don't work recursively.

`--prepare` `-p`	Used in conjunction with `--image`, this option generates only the Image Description Tree; the file system image is not created. The resulting structure, the root image tree, can be manually modified. The root directory of the image is named root-*image_name-version* and is located in the current directory.
`--setenv` *`env_var=value`* `-s` *`env_var=value`*	Used in conjunction with the `--prepare` command, this option sets the environment variable. For example: `xscr --prepare --image image-2.0.3 --setenv SCR_BUILD_DIR=/tmp` **NOTE:** The value of the SCR_BUILD_DIR environment variable creates an image root tree in the `/tmp` directory.
`--unsetenv` *`env_var=value`* `-u` *`env_var=value`*	Used in conjunction with `--prepare`, this option unsets the environment variable. This option takes precedence over the `--setenv` option.
`--update-product-file`	Lets you update the product file within a DiskNetboot image without rebuilding the image. For sample usage, see Section 9.7, "Updating the Product File in a Boot Image," on page 168.

Option	Description
`--verify` `-V`	Used in conjunction with `--prepare`, this option verifies all RPM packages after they are installed. When finished, ImageBuilder displays the verification results.
`--version`	Returns the ImageBuilder version number.

9.2 xscr Image Building Components

xscr builds images using the Image Description Tree, an Image Specification Document (`ImageSpecification.xml`), and a Distribution Source Document (`Distribution.xml`). The Image Specification and Distribution Source Documents contain XML elements that define the structure, configuration files, and other components required to build client images for Point of Service systems.

The XML-based Image Specification and Distribution Source Documents provide significant advantages in image design and manageability:

- You can manage image subcomponents as discrete elements. Image drivers, RPMs, and even features can be separately managed within the Image Specification Document. This allows you to easily add features, RPMs, and drivers to an image.
- You can define global settings in the parent Image Specification Document to customize the implementation of image subcomponents. For example, in the parent document, you can choose to include or exclude specific drivers or RPMs.

This granular level of control is made possible by the structure of the Novell Linux Point of Service XML schema. The Novell Linux Point of Service XML schema organizes the template components in discrete elements so they can be individually managed.

The following sections review the image components required to build images with xscr:

- Section 9.2.1, “Image Description Tree,” on page 126
- Section 9.2.2, “Image Specification Documents,” on page 128
- Section 9.2.3, “Distribution Source Document (Distribution.xml),” on page 140

9.2.1 Image Description Tree

xscr builds images using a specific file system directory structure known as the Image Description Tree. The Image Description Tree provides the XML documents, scripts, configuration files, and other components required to build client images for Point of Service systems.

NOTE: xscr uses the Image Specification (`ImageSpecification.xml`) and Distribution Source (`Distribution.xml`) documents in place of the IMAGE, VERSION, config, setup, and setup.user files used by scr to generate a client image.

Table 9-2 summarizes the Image Description Tree components required to build images with xscr.

Table 9-2 *Image Description Tree components for xscr*

Component	Description
`/opt/SLES/POS/system/`*`image_name-version`*`/files/`	A subdirectory that contains special files, directories, and scripts, This function of this directory is to ensure that the RPM is used as the package manager before any packages are installed in the image. The entire directory is copied to the root of the image tree using `cp -a`. This directory cannot contain any libraries or binary files. Any binaries and libraries required before the first RPM call must be extracted from the corresponding packages in advance.
`/opt/SLES/POS/system/`*`image_name-version`*`/files-user/`	A subdirectory that contains special files, directories, and scripts for adapting the image environment after the installation of all the image packages.
`/opt/SLES/POS/system/`*`image_name-version`*`/package/`	A subdirectory where ImageBuilder searches for RPM packages. The directory is automatically initialized depending on the entries in the Distribution Source file, Distribution.xml. For more information, see Section 9.2.3, “Distribution Source Document (Distribution.xml),” on page 140. If there is no package directory, ImageBuilder creates a link to the global package directory (`/opt/SLES/POS/pac/`) and the links designated in the `AdminServer.conf` file.
`/opt/SLES/POS/system/`*`image_name-version`*`/script/`	A subdirectory that contains Bash scripts that are called after a package is installed, primarily to remove the parts of a package that are not needed for the Point of Service system. **IMPORTANT:** For these scripts to run, the script name must match the name of the RPM (without the version).
`/opt/SLES/POS/system.`*`image_name-version`*`/config.cleanup`	An optional configuration script for the image. This script is called at the end of the installation and after all the installation scripts have run. It is designed to clean up the image system. The target programs and files are those needed only while the installation scripts are running.
`/opt/SLES/POS/system.`*`image_name-version`*`/config.system`	An optional configuration script for the image. This script is called at the end of the installation but before the installation scripts have run. It is designed to configure the image system, such as the activation or deactivation of certain services (insserv). The call is not made until after the switch to the image has been made with chroot. **IMPORTANT:** This file is pre-defined to provide the scripts required to install the image. Do not modify this file.

Component	Description

`/opt/SLES/POS/system/`
`image_name-version``/`
`setup.txt`
	An optional information file for the LDAP system. This file contains information regarding which configuration files are required by the image and whether they are hardware or system-dependent. The structure of the file is as follows:
	`flag : configuration_file_name`
	The following values can be set for the *flag* value:
	◆ **SYS** specifies that the configuration file is hardware-independent, such as `/etc/ntp.conf`.
	◆ **HWD** specifies that the configuration file is hardware-dependent, such as `/etc/X11/XF86Config`.
	The *configuration_file_name* includes the full file path.

9.2.2 Image Specification Documents

Image Specification Documents contain XML elements that define the structure, configuration, and other components required to build client images for Point of Service systems. In general, a master Image Specification Document (or parent document) defines general image parameters and individual image subcomponents such as add-on features, custom applications, and so forth are defined in sub-documents referred to as child documents.

Novell Linux Point of Services allows you to nest multiple child documents within a parent Image Specification Document. These child documents can be located anywhere and can be given any filename. The parent Image Specification Document must be named `ImageSpecification.xml` and must be located at the root of the Image Description Tree (`/opt/SLES/POS/system/`*`image_name-version`*).

NOTE: Client Image Specification Documents can be defined in an XML editor or in a standard text editor. XML editors provide the advantage of a graphical user interface. Typically, XML elements are presented as graphical objects and are visually organized in the schema hierarchy. Element attributes are defined as fields within the element objects. After the XML template is defined, the template can be saved as a standard XML document. The graphics in the following sections were taken in an XML editor. They show XML schema in a graphical format.

Novell Linux client Image Specification Documents can also be defined in a standard text editor. Text-based XML documents are more complicated because the schema hierarchy and element attributes are defined through the document syntax and organization. The sample XML documents in Appendix C, “Sample Files,” on page 235 are presented in text format.

Figure 9-1 represents the root elements in the Novell Linux Point of Service XML schema.

Figure 9-1 *Novell Linux Point of Service XML root elements*

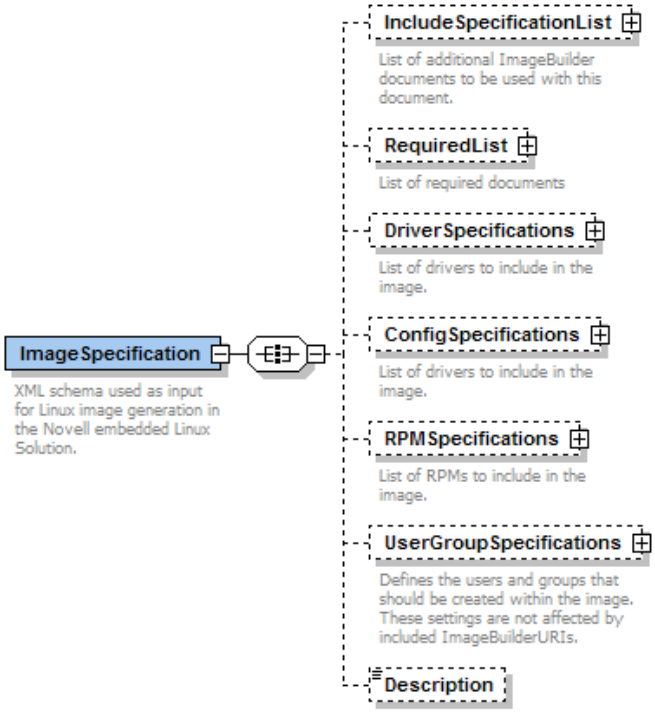


Table 9-3 summarizes the root elements in the XML Image Specification schema.

Table 9-3 *Image Specification root elements*

Root Element	Description
ImageSpecification	Defines general image settings such as image name, version, time zone, and so forth. The ImageSpecification settings in parent documents take precedence over settings in child documents.
IncludeSpecificationList	Identifies the child Image Specification Documents to include when generating the parent document in ImageBuilder.
Required List	Identifies the Image Specification Documents required to build the image.
DriverSpecifications	Identifies the USB, network, and general drivers to include or exclude from the image. DriverSpecifications settings in parent documents take precedence over child documents. That means the parent document can exclude any item that is in the include list of a child document or conversely, the parent document can include any item that is in the exclude list of a child document.

Root Element	Description
ConfigSpecifications	This element is not currently implemented. These configuration specifications must be defined through the config.system, and setup.txt files in the Image Description Tree. For more information, see Section 9.2.1, “Image Description Tree,” on page 126.
RPMSpecifications	Identifies the RPMs to include or exclude from the image. RPMSpecifications settings in parent documents take precedence over child documents. That means the parent document can exclude any item that is in the include list of a child document or conversely, the parent document can include any item that is in the exclude list of a child document.
UserGroupSpecifications	Defines the users and groups to create within the image. The UserGroupSpecifications settings in the parent document take precedence over settings in child documents.
Description	Provides a general description of the client image. This element is only read in the root Image Specification Document (ImageSpecification.xml); it is ignored in all child documents.

The following sections provide detailed information on the Novell Linux Point of Service XML root elements and their sub-elements.

- “ImageSpecification” on page 130
- “IncludeSpecificationList” on page 133
- “DriverSpecifications” on page 134
- “RPMSpecifications” on page 136
- “UserGroupSpecifications” on page 138
- Section 9.2.3, “Distribution Source Document (Distribution.xml),” on page 140

ImageSpecification

The ImageSpecification element defines general image settings such as image name, version and time zone. The ImageSpecification settings in parent documents take precedence over settings in child documents.

Table 9-4 summarizes the attributes in the ImageSpecification element.

Table 9-4 *ImageSpecification element attributes*

Attribute	Description
ImageName=*"image_name"*	*Image_name* indicates the name of the image. When the image is generated, the *image_name* is automatically expanded to include the version number and the date. The version number is extracted from the ImageVersion attribute. xscr replaces the *image_name* with the name specified on the command line when you clone the Image Description Tree. **IMPORTANT:** You cannot use the word "boot" in any image name other than the cdboot and disknetboot images. This attribute is required in the parent ImageSpecification.xml document; it is optional in a child ImageSpecification.xml document.
SchemaVersion="*version*"	The XML schema version for the current Image Specification Document. The current version is "1." This attribute is required.
SchemaRevision="*revision*"	The XML schema revision number for the current Image Specification Document.The current revision is "1." This attribute is required.
ImageType="diskful|diskless"	The value for this parameter is either *diskful* or *diskless*. If diskful is set, all the packages required to handle the image on a hard disk are included in the setup description. If diskless is set, all unnecessary packages are removed from the setup description. This attribute is optional for the DiskNetboot image. However, for all other images, you must define the ImageType as diskful or diskless.
ImageVersion=*"version"*	The version number of the image generated from the current Image Specification Document. xscr replaces the *version* with the version specified on the command line when you clone the Image Description Tree. This attribute is required in the parent ImageSpecification.xml document; it is optional in a child ImageSpecification.xml document.
AddOnSize=*"Image_size* "	The size that is added to the final image size. *Image_size* is defined as a number followed by M (megabyte) or G (gigabyte). If you specify a value of "1 M," xscr adds 1 MB to the final image size. This attribute is optional.

Attribute	Description
Type=”ext2|ext3”	The image type is currently restricted to ext2 or ext3, although, if necessary, different formats are possible. **NOTE:** If you have an existing ext2 image, you can change the file system by setting a flag in the scCashRegister or the scWorkstation objects rather than recreate the image. If ext3 is specified in either LDAP object, the Point of Service terminal extends the file system to ext3 when the image is deployed. This attribute is required in the parent ImageSpecification.xml document; it is optional in a child ImageSpecification.xml document.
Timezone=”time_zone”	All time zone definitions are located in the /usr/share/zoneinfo directory. To specify which time zone you want to use in the image, enter the relative path to a specific time zone definition. For example, timezone-”US/Mountain.” The ImageBuilder uses this information to extract the corresponding time zone from the timezone package. The time zone value is then stored as /etc/localtime in the image. This attribute is optional for the DiskNetboot image. However, we recommend that you define the Timezone attribute for all other images.
Locale=”locale_names”	Contains a comma-separated list of valid locale names. Novell Linux Point of Service provides support for the following locales: ◆ de_DE (German) ◆ es_ES (Spanish) ◆ fr_FR (French) ◆ it_IT (Italian) ◆ ja_JP (Japanese) ◆ ko_KR (Korean) ◆ pt_PT (Portuguese) ◆ zh_CN (Simplified Chinese) ◆ zh_TW (Traditional Chinese) **IMPORTANT:** In addition to designating the image locale, you must add the child Image Specification Documents required to provide the language files for each image feature. For more information, see “Changing the Image Language” on page 152. This attribute is optional for the DiskNetboot image. However, it is recommended that you define the Locale attribute for all other images.

Attribute	Description
Keytable=”console_keymap”	Contains the name of the console keymap to use. The name corresponds to a map file stored below the path /usr/share/kbd/keymaps; for example, keytable=”us.map.gz”. Additionally, the KEYTABLE variable within the /etc/sysconfig/keyboard file is set according to the keyboard mapping. This attribute is optional for the DiskNetboot image. However, we recommend that you define the Keytable attribute for all other images.

IncludeSpecificationList

Novell Linux Point of Services allows you to nest multiple child Image Specification Documents within a parent document. These child documents can be located anywhere and can be given any filename.

The ImageSpecificationList element identifies the child Image Specification Documents to include when generating the parent document in ImageBuilder.

Figure 9-2 represents the IncludeSpecificationList elements.

Figure 9-2 *IncludeSpecificationList elements*

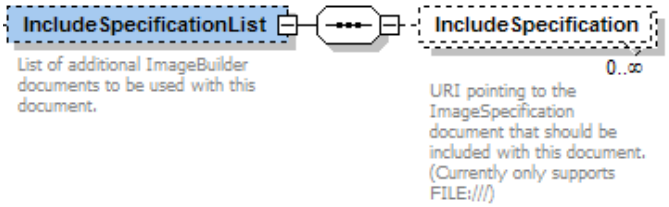


Table 9-5 summarizes the sub-elements and attributes in the IncludeSpecificationList element.

Table 9-5 *IncludeSpecificationList element attributes*

Sub-Element	Attribute	Description
IncludeSpecification	URI	The URI of a child document to include with the current Image Specification Document. **NOTE:** Currently, xscr only supports local URIs (FILE:///). When ImageBuilder generates `ImageSpecification.xml`, it also processes the child Image Specification Documents and includes them as part of the image. This attribute is required.

For information on using the IncludeSpecificationList element to add features to a client image, see “Adding Features to Client Images” on page 146.

RequiredList

The RequiredList element identifies image dependencies for client add-on features. If an add-on image requires a specific parent document, the dependency is expressed in the RequiredList element. For example, the Firefox add-on feature requires the `browser.xml` Image Specification document; therefore, this feature can be added only to the Browser or Desktop client images.

Figure 9-3 represents the sub-elements and attributes in the RequiredList element:

Figure 9-3 *RequiredList elements*

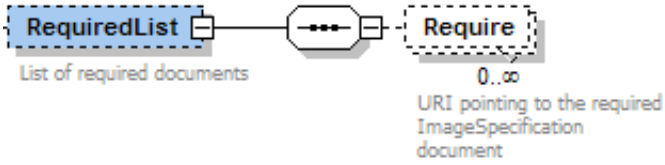


Table 9-6 summarizes the sub-elements and attributes in the RequiredList element.

Table 9-6 *RequiredList sub-elements and attributes*

Sub-Element	Attribute	Description
Require	URI	A URI that identifies where the required Image Specification Documents are located. **NOTE:** Currently, xscr only supports FILE:///. **IMPORTANT:** All Image Specification Documents listed in each child document's RequiredList element must exist in the parent Image Specification Document for xscr to build the image. If one of the required documents is not present, xscr returns the following error: `Dependency check failed on image_name.xml.`

DriverSpecifications

The DriverSpecifications element identifies the USB, network, and general drivers to include or exclude from the image. DriverSpecifications settings in parent documents take precedence over child documents. That means the parent document can exclude any item that is in the include list of a child document or conversely, the parent document can include any item that is in the exclude list of a child document.

NOTE: Include settings in parent documents take precedence over child documents; that is, the parent document can include any item that is in the exclude list of a child document.

Similarly, exclude settings in parent documents take precedence over child documents; that is, the parent document can exclude any item that is in the include list of a child document.

Figure 9-4 represents the sub-elements and attributes in the DriverSpecifications element:

Figure 9-4 *DriverSpecifications elements*

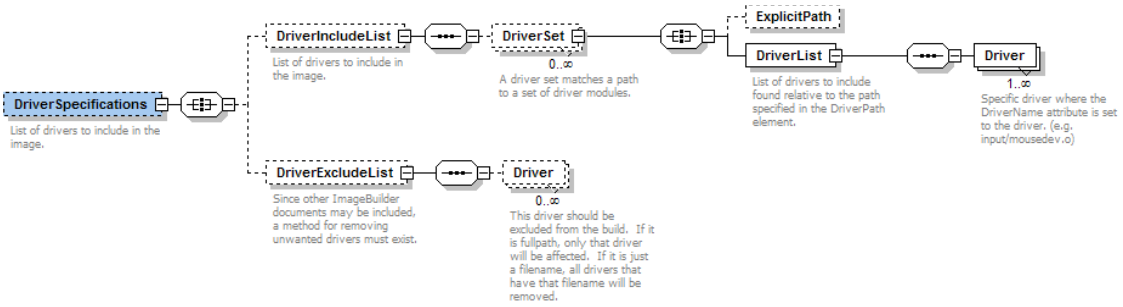


Table 9-7 summarizes the sub-elements and attributes in the DriverSpecifications element.

Table 9-7 *DriverSpecifications element attributes*

Sub-Element	Attribute	Description
DriverIncludeList		The list of USB, network, and general drivers to include in the image.
DriverSet		A set of drivers that are located in a single path. Ideally, you should define separate DriverSets for your USB, network, and general drivers. These must be in the kernel RPM.
ExplicitPath	URI	A URI that identifies where the drivers in the DriverList element are located. **NOTE:** Currently, xscr only supports FILE:///. This attribute is required.
DriverList		The list of drivers to include in the image.
Driver	Name	A text string that identifies the driver filename to include in the image. The driver name is indicated relative to the ExplicitPath. For example, `input/mousedev.ko` This attribute is required.
DriverExcludeList		The list of USB, network, and general drivers to exclude from the image. This element allows you to exclude drivers included in child documents.
Driver	Name	A text string that identifies the driver filename to exclude from the image. If the name includes a full path, ImageBuilder excludes only that particular driver. If the name is just a filename, ImageBuilder excludes all drivers with that filename from the image. This attribute is required.

For information on adding or excluding drivers from a client image, see “Adding Drivers” on page 148.

RPMSpecifications

The RPMSpecifications element identifies the RPMs to include or exclude from the image.RPMSpecifications settings in parent documents take precedence over child documents. That means the parent document can exclude any item that is in the include list of a child document or conversely, the parent document can include any item that is in the exclude list of a child document.

NOTE: Include settings in parent documents take precedence over child documents; that is, the parent document can include any item that is in the exclude list of a child document.

Similarly, exclude settings in parent documents take precedence over child documents; that is, the parent document can exclude any item that is in the include list of a child document.

Figure 9-5 represents the sub-elements and attributes in the RPMSpecifications element.

Figure 9-5 *RPMSpecifications elements*

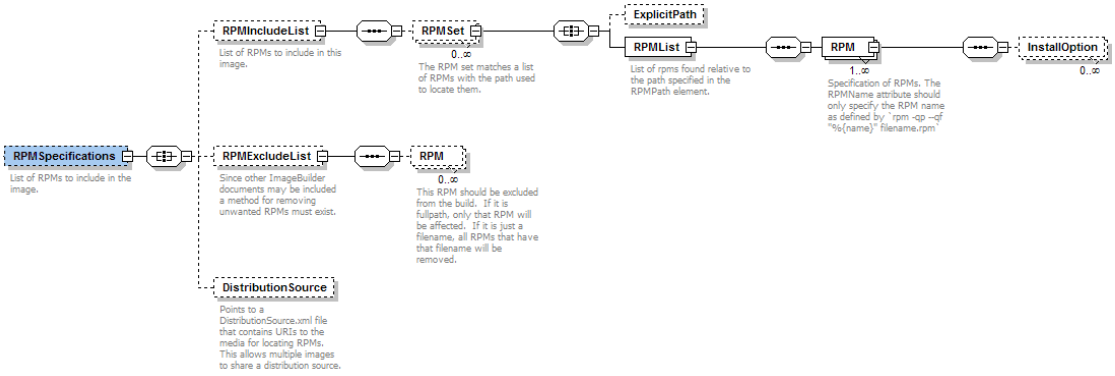


Table 9-8 summarizes the sub-elements and attributes in the RPMSpecifications element.

Table 9-8 *RPMSpecifications element attributes*

Sub-Element	Attribute	Description
RPMIncludeList		The list of the RPMs to include in the image.
RPMSet		A set of RPMs that are located in a single path. A separate RPMSet element must be defined for each ExplicitPath.
ExplicitPath	URI	A URI that identifies where the RPMs in the RPMList element are located. If the value is “x,” ImageBuilder searches all the paths specified in the Distribution.xml file. This attribute is required.
RPM List		The list of RPMs to include in the image.

Sub-Element		Attribute	Description
	RPM	Name	A text string that identifies the RPM filename to include in the image. The RPM name is indicated relative to the ExplicitPath. The Name attribute should specify the RPM name as defined by the following command: `rpm -qp --qf "%{name}" filename.rpm` This attribute is required.
		Version	The version of the RPM to include in the image. If the value is "x," ImageBuilder includes the latest RPM in the image. This attribute is optional.
	InstallOption	Option	A command option that defines installation parameters for RPMs in the current RPM list. The available install options are as follows: ◆ `--excludedocs` Do not install documentation. ◆ `--includedocs` Install documentation. ◆ `--replacepkgs` Replace a package with a new copy of itself. ◆ `--replacefiles` Replace files owned by another package. ◆ `--force` : Ignore package and file conflicts. ◆ `--noscripts` Do not execute pre- and postinstall scripts. ◆ `--ignorearch` Do not verify package architecture. ◆ `--ignoreos` : Do not verify package operating system. ◆ `--nodeps` Do not check dependencies. ◆ `--nosignature` Do not verify package or header signature. This attribute is required.
RPMExcludeList			The list of RPMs to exclude from the image. This element allows you to exclude RPMs included in child documents.

Sub-Element	Attribute	Description
RPM	Name	A text string that identifies the RPM filename to exclude from the image. If the name includes a full path, ImageBuilder excludes only that particular RPM. If the name is just a filename, ImageBuilder excludes all RPMs with that filename from the image. This attribute is required.
	Version	The version of the RPM to exclude from the image. If the value is “all,” ImageBuilder excludes all versions of the RPM from the image. This attribute is optional.
DistributionSource	URI	A URI that identifies where the Distribution.xml file is located. The Distribution.xml template is located in /opt/SLES/POS/system/templates/. For more information, see Section 9.2.3, “Distribution Source Document (Distribution.xml),” on page 140. This attribute is required.
	ImageClass	The type of image generated from the current Image Specification Document. The available image classes are NLD and SLES. In general, most Point of Service images are created using the NLD image class. The only images that require the SLES image class are POSBranch images. **IMPORTANT:** The ImageClass element must match the ImageClass definition in the Distribution Source Document.

For information on adding or excluding RPMs from a client image, see “Adding RPMs” on page 150.

UserGroupSpecifications

The UserGroupSpecifications element defines the users and groups that are created within the image. (For more information on standard templates setup.user files, see Section 8.2.1, “Image Description Tree,” on page 102.) All users and groups that will be logging into the Point of Service terminals configured by the current image can be pre-defined within the image.

The UserGroupSpecifications settings in the parent document take precedence over settings in child documents.

Figure 9-6 represents the sub-elements and attributes in the UserGroupSpecifications element.

Figure 9-6 *UserGroupSpecifications elements*

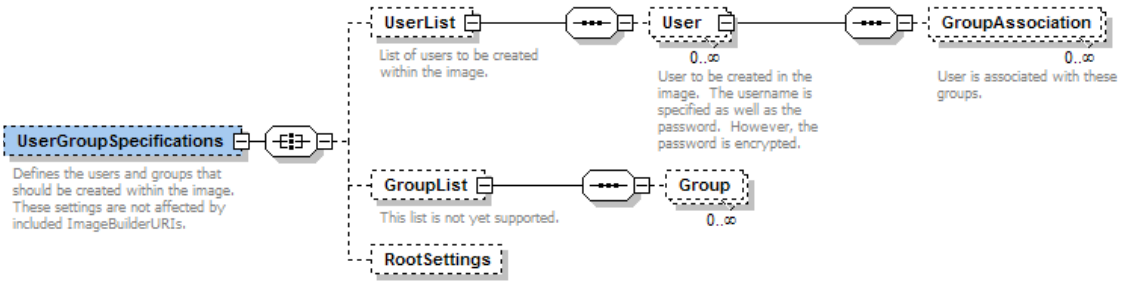


Table 9-9 summarizes the sub-elements and attributes in the UserGroupSpecifications element.

Table 9-9 *UserGroupSpecifications element attributes*

Element	Attribute	Description
UserList		The user accounts created on Point of Service terminals configured by images generated from the current Image Specification Document.
User	Name	A text string that specifies the user account to create in the image. This attribute is required.
	EncryptedPassword	A text string that specifies the encrypted password for the user account. **IMPORTANT:** This value must be the encrypted password. You must use an encryption utility to encrypt the user account password before it can be entered as the value for this attribute. This attribute is optional.
	HasPassword	A Boolean value (true/false) that indicates if the user has a password. This attribute is required.
	UserID	A text string that specifies a unique user ID. This value must be unique within the root Image Specification Document and its children. This attribute is required.
	HomeDirectory	A text string that identifies the user's home directory. This attribute is optional.
	Main Group	A text string that identifies the main group the user belongs to. This attribute is optional.

Element	Attribute	Description
	Disable	A Boolean value (true/false) that disables access to the user account. The default value is false.
GroupAssociation	Name	A text string that specifies the names of groups this user belongs to. This attribute is required.
GroupList		The Linux user groups created on Point of Service terminals configured by images generated from the current Image Specification Document.
Group	Name	A text string that specifies the user account to create in the image. This attribute is required.
	GroupID	A text string that specifies a unique group ID. This value must be unique within the root Image Specification Document and its children. This attribute is required.
RootSettings	DisableRootAccess	A Boolean value (true/false) that disables the Root account. This attribute is optional.
	EncryptedRootPassword	A text string that specifies the encrypted password for the Root account. **IMPORTANT:** This value must be the encrypted password. You must use an encryption utility to encrypt the Root account password before it can be entered as the value for this attribute. This attribute is only valid if the Root account is enabled. This attribute is optional.

For information on using the UserGroupSpecifications element to manage users, groups, and the Root account in a client image, see "Managing Users and Groups within an Image" on page 151.

Sample ImageSpecification.xml Documents

To view examples of Image Specification Documents, see the following:

- Section C.3.1, "ImageSpecification.xml Template," on page 238
- Section C.3.2, "Defined ImageSpecification.xml Document," on page 240

9.2.3 Distribution Source Document (Distribution.xml)

When xscr generates an image, it searches the paths listed in the Distribution Source Document to find the RPM packages required to create the image. The Distribution Source Document,

`Distribution.xml`, defines the paths to the distribution directories where you have copied the Novell Linux Point of Service CDs.

The `Distribution.xml` document also references the maintenance directory. The maintenance directory is essentially an “override” directory. RPMs located in this directory take precedence over RPMs located in the distribution directories. You can add any RPM to this directory that you want xscr to use in lieu of the default RPMs in the distribution directories. By default, the maintenance directory contains the glibc and devs RPMs. For a detailed breakdown of the maintenance directory structure, see Section B.2, “Branch Server Directory Structure,” on page 231.

By default, the Distribution.xml document is located in `/opt/SLES/POS/system/templates/` . The location of the Distribution.xml document must be defined in the DistributionSource attribute in the Image Specification Document.

The posCDTool is used to generate the Distribution Source Document. For more information, see Section 7.3.4, “Generating AdminServer.conf or Distribution.xml,” on page 94.

The Distribution Source Document defined in the parent document takes precedence over Distribution Source Documents referenced in child documents.

NOTE: Novell Linux Point of Service Distribution Source Documents can be defined in an XML editor or in a standard text editor. XML editors provide the advantage of a graphical user interface. Typically, XML elements are presented as graphical objects and are visually organized in the schema hierarchy. Element attributes are defined as fields within the element objects. After the XML template is defined, the template can be saved as a standard XML document. The graphics in this section were taken in an XML editor. They show XML schema in a graphical format.

Distribution Source Documents can also be defined as XML documents in standard text format. These documents are more complicated because the schema hierarchy and element attributes are defined through the document syntax and organization. The XML documents in Appendix C, “Sample Files,” on page 235 are presented in text format.

Figure 9-7 represents the sub-elements and attributes in the `Distribution.xml` document.

Figure 9-7 *Distribution.xml document elements and attributes*

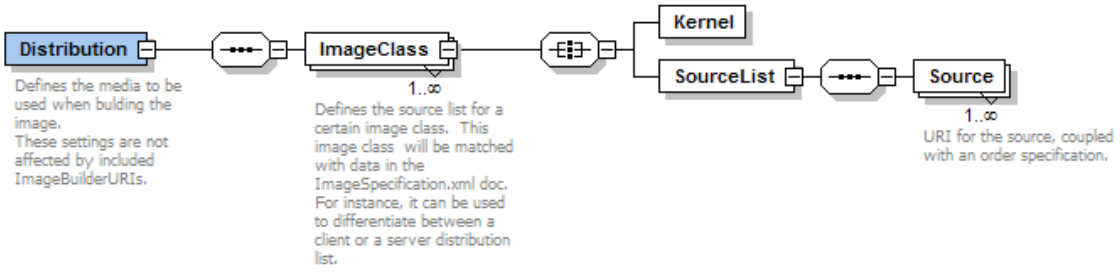


Table 9-10 summarizes the sub-elements and attributes in the `Distribution.xml` document.

Table 9-10 *Distribution.xml element attributes*

Element	Attribute	Description
Distribution	SchemaVersion="*version*"	The XML schema version for the current Image Specification Document. The current version is "1." This attribute is required.
	SchemaRevision="*revision*"	The XML schema revision number for the current Image Specification Document. The current revision is "1." This attribute is required.
ImageClass	Name	The type of image generated from Image Specification Documents that reference this Distribution Source Document. The default image classes are NLD and SLES. The NLD ImageClass generates client images. The SLES ImageClass generates POSBranch images. **IMPORTANT:** This ImageClass element must match the ImageClass definition in the RPMSpecifications element in all associated Image Specification Documents.
Kernel	Name	The name of the kernel required for the image. The kernel name for Novell Linux Point of Service 9 images is kernel-SLRS.
	Version	The version of the kernel required for the image.
	Path	The absolute path to the kernel required for the image.
SourceList		A list of URIs to the media where RPMs required to generate the image are located. This element enables multiple images to share a single Distribution Source Document.
Source	URI	A URI to the media where RPMs required to generate the image are located.
	Order	Any positive number, zero or higher. This value indicates the search order for the current URI. When ImageBuilder generates the image, it searches each URI in their designated order for RPMs listed in the RPMIncludeList element.

Sample Distribution.xml Documents

To view examples of Distribution Source Documents, see the following:

- Section C.4.1, "Distribution.xml Template," on page 244
- Section C.4.2, "Defined Distribution.xml Document," on page 245

9.3 Getting Ready to Build Images with xscr

Before you can build client images with xscr, you must complete the following tasks:

1. Install ImageBuilder and the image templates.
2. Copy the image source files from the Novell Linux Point of Service CDs to a central distribution directory.
3. Define the location of the image source files.

These tasks are explained in the following sections.

9.3.1 Installing ImageBuilder and the Image Templates

ImageBuilder and the corresponding image templates are installed when you select the NLPOS Image Server Software Selection option during the Administration Server installation. For further information on creating an image server, refer to the *Novell Linux Point of Service 9 Installation Guide*.

During installation of the image server, the following image building components are installed:

- The ImageBuilder packages (scr and xscr) are installed to the `/usr/bin` directory.
- The Image Description Trees for each image are installed to `/opt/SLES/POS/system/image_name-version`. For information on xscr Image Description Trees, see Section 9.2.1, “Image Description Tree,” on page 126.
- The template Image Specification Documents for CDBoot, DiskNetboot, Minimal, Java, Browser, Desktop, and POSBranch images are installed to `/opt/SLES/POS/system/templates/support`. These documents specify the RPMs and drivers required to build their respective images and are included as child documents in their parent Image Specification Document (`ImageSpecification.xml`) at the root of the Image Description Tree. For more information, see Section 4.2, “Point of Service Boot Images,” on page 43 and Section 4.3, “Point of Service Client Images,” on page 45.
- The `Distribution.xml` template is installed to `/opt/SLES/POS/system/templates`. For more information, see Section 9.2.3, “Distribution Source Document (Distribution.xml),” on page 140.
- The default configuration information for all kernel drivers are installed to `/opt/SLES/POS/system/templates/drivers`.
- The child Image Specification Documents for the client image add-on options such as Samba3 client, GNOME, KDE, and VNC 4 Remote Control Client are installed to `/opt/SLES/POS/system/templates/addons`. For more information, see Section 4.4, “Client Image Add-On Features,” on page 49.

9.3.2 Copying the Novell Linux Point of Service CDs to a Central Distribution Directory

To build the client images, ImageBuilder must have access to the source RPMs. Therefore, before building client images, you must copy the Novell Linux Point of Service source CDs to a distribution directory on the Administration Server.

The POSCDTool and POSCopyTool utilities included with Novell Linux Point of Service copy the RPMs required to build client images. For information on this procedure, see Section 7.3.1, “Copying the Novell Linux Point of Service CDs,” on page 92.

9.3.3 Defining the Location of the Image Source Files

When ImageBuilder builds an image, it must know where it can locate the RPMs required to build the image.

For xscr, the location of the RPM packages is defined in the Distribution Source Document (`Distribution.xml`). This XML document defines the paths to the distribution directories where you have copied the Novell Linux Point of Service CDs.

When xscr generates an image, it searches the URIs listed in the `Distribution.xml` document to find the RPM packages required to create the image. The order in which xscr searches the URIs is determined by the Order attribute for each URI. For more information on the Distribution Source Document, see Section 9.2.3, “Distribution Source Document (Distribution.xml),” on page 140. For information on creating the document, see Section 7.3.4, “Generating AdminServer.conf or Distribution.xml,” on page 94.

9.4 Building Images with xscr

After you have installed ImageBuilder and the image templates, copied the image source files to a distribution directory, and defined the image source location files, you can start building Point of Services images.

The process required to build an image with xscr is as follows:

1. Clone the Image Description Tree.
2. Customize the Image Specification Document (ImageSpecification.xml).
3. Configure the image.
4. Build the image.

These steps are discussed in the following sections.

9.4.1 Cloning the Image Description Tree

xscr builds images using a specific file system directory structure known as the Image Description Tree. The Image Description Tree provides the XML documents, configuration files, and other components required to build images for Point of Service systems.

You can use the default Image Description Trees provided with Novell Linux Point of Service to generate the DiskNetboot, CDBoot, Minimal, Browser, Java, and Desktop images. However, to maintain a standardized source tree and simplify the upgrade process, we recommend you maintain the default Image Description Trees provided with Novell Linux Point of Service as master copies. To build your own images, you can clone the default Image Description Trees, then modify the cloned tree.

When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (`--dist nld|sles`). If you define the image distribution as NLD, xscr adds a child document (*`image_name`*`.xml`) to the parent Image Specification Document that includes the NLD RPMs in the image. Conversely, if you define the image distribution as SLES, xscr adds a

child document (*image_name*-sles.xml) to the parent Image Specification Document that includes the SLES RPMs in the image.

NOTE: NLD is the default distribution type. Therefore, if you do not specify the image distribution when cloning the Image Description Tree, xscr defaults to NLD.

In general, most Point of Service images are created using the NLD distribution. The only images that require the SLES distribution are POSBranch images. For more information, see Section 10.2, "Building POSBranch Images," on page 176.

The basic syntax to clone an Image Description Tree is as follows:

```
xscr --create image_name-version --image image_name-version --dist
nld|sles
```

For example, the following command clones the Desktop-2.0.21 Image Description Tree to create a new NLD-based Image Description Tree named myImage-1.1.1:

```
xscr --create myImage-1.1.1 --image desktop-2.0.21
```

IMPORTANT: You cannot use the word "boot" in any image name other than the cdboot and disknetboot images.

The new Image Description Tree is located at /opt/SLES/POS/system/myImage-1.1.1. You can then modify the cloned Image Description Tree as required to create your new image. For a description of the individual Image Description Tree components, see Section 9.2.1, "Image Description Tree," on page 126 and Appendix B, "Novell Linux Point of Service Files and Directory Structure," on page 211.

9.4.2 Customizing the Image Specification Document

Image Specification Documents contain XML elements that define the structure, configuration, and other components required to build images for Point of Service systems. In general, a master Image Specification Document (or parent document) defines general image parameters and individual image subcomponents such as add-on features, custom applications, and so forth are defined in sub-documents referred to as child documents.

Novell Linux Point of Services allows you to nest multiple child documents within a parent Image Specification Document. These child documents can be located anywhere and can be given any filename. The parent Image Specification Document must be named ImageSpecification.xml and must be located at the root of the Image Description Tree (/opt/SLES/POS/system/*image_name-version*).

The default Image Description Trees provided with Novell Linux Point of Service have a parent Image Specification Document at the root of the tree. After you clone the tree, you can customize the parent Image Specification Document as needed to build your image.

The following sections outline how to customize elements in Image Specification Documents:

- "Adding Features to Client Images" on page 146
- "Adding Drivers" on page 148
- "Adding RPMs" on page 150
- "Managing Users and Groups within an Image" on page 151

- "Changing the Image Language" on page 152
- "Setting Image Configuration Settings" on page 158

Adding Features to Client Images

Novell Linux Point of Service provides add-on features that can be added to client images generated with xscr. To add a feature to a client image, simply reference the add-on feature's Image Specification Document (that is, child document) within the client Image Specification Document (that is, parent document). The parent document's IncludeSpecificationList element identifies the child documents that you want ImageBuilder to include when it generates the image.

For example, to include the VNC 4 Remote Control add-on option in a client image, you must provide the URI to the `vnc.xml` child document in the parent document's IncludeSpecificationList element, as follows:

```
<IncludeSpecificationList>
   <IncludeSpecification URI="/opt/SLES/POS/system/templates/addons/
   vnc.xml"/>
</IncludeSpecificationList>
```

Table 9-11 lists the Image Specification Documents for add-on features that can be added to client images.

IMPORTANT: Some of the add-on features have dependencies. The dependencies are noted in Table 9-11; however, you can also check the RequiredList element in the Image Description Document to verify dependencies. If the image does not have a RequiredList element, the add-on feature can be added to any client image. For more information, see "RequiredList" on page 134.

Table 9-11 *Image Specification Documents for client add-on features*

Feature	Image Specification Document	Description
admind	/opt/SLES/POS/system/templates/addons/admind.xml	Adds the admind utility to client images. admind allows simple commands to be executed on Point of Service terminals from a remote location. For more information, see Chapter 11, "Remotely Managing Point of Service Terminals with admind and adminc," on page 187. This feature can be added to any NLD-based client image.
Advanced Linux Sound Library	/opt/SLES/POS/system/templates/addons/alsa.xml	Adds the Advanced Linux Sound Library (ALSA) to client images. ALSA provides audio and MIDI functionality for Point of Service terminals. This feature can be added to any client image.
Debug	/opt/SLES/POS/system/templates/addons/debug.xml	Adds debugging tools to client images for troubleshooting purposes.This feature can be added to any client image.

Feature	Image Specification Document	Description
EvTouch	/opt/SLES/POS/system/templates/addons/evtouch.xml	Adds the driver for evtouch screens in ncurses mode. **NOTE:** This driver does not support evtouch screens in X11 mode. This feature can be added only to the Java, Browser, or Desktop images.
Firefox	/opt/SLES/POS/system/templates/addons/firefox.xml	Adds the Firefox browser to client images. This feature can be added only to the NLD-based Browser or Desktop images.
GNOME 2.6 for NLD	/opt/SLES/POS/system/templates/addons/gnome.xml	Adds the GNOME desktop to NLD-based client images. This feature can be added only to the NLD Desktop image.
GNOME 2.6 for SLES	/opt/SLES/POS/system/templates/addons/gnome-sles.xml	Adds the GNOME desktop to SLES-based images used for POSBranch. This feature can be added only to the SLES Desktop image.
IBM Java	/opt/SLES/POS/system/templates/addons/ibmjava.xml	Adds the current IBM Java Runtime Environment (JRE) to NLD-based client images. This feature can be added to the Java, Browser, or Desktop images.
KDE 3.2 for NLD	/opt/SLES/POS/system/templates/addons/kde.xml	Adds the KDE desktop to NLD-based client images. This feature can be added only to the NLD Desktop image.
KDE 3.2 for SLES	/opt/SLES/POS/system/templates/addons/kde-sles.xml	Adds the KDE desktop to SLES-based images used for POSBranch. This feature can be added only to the SLES Desktop image.
Mozilla	/opt/SLES/POS/system/templates/addons/mozilla.xml	Adds the Mozilla browser to client images. This feature can be added to the Browser or Desktop images.
Samba 3 Client	/opt/SLES/POS/system/templates/addons/samba.xml	Provides Common Internet File System (CIFS) file access for Windows and Linux clients. **NOTE:** The Samba 3 server is included with Novell Linux Point of Service. This feature can be added to any client image.

Feature	Image Specification Document	Description
Vim	/opt/SLES/POS/system/templates/addons/vim.xml	Adds Vim (Vi IMproved) to client images. Vim is an almost compatible version of the UNIX editor vi. Almost every possible command can be performed using only ASCII characters. Many new features have been added such as multilevel undo, command line history, filename completion, block operations, and editing of binary data. Vi is available for the AMIGA, MS-DOS, Windows NT, and various versions of UNIX. This feature can be added to any client image.
VNC 4 Remote Control Client	/opt/SLES/POS/system/templates/addons/vnc.xml	Adds the VNC 4 Remote Control client to the image so you can remotely control the Point of Service terminal over any TCP/IP connection. This feature can be added to Java, Browser or Desktop images.
YaST2	/opt/SLES/POS/system/templates/addons/yast2.xml	Adds the YaST2 console to client images. YaST2 is the system configuration console. It can configure hardware (sound cards, printers, keyboards, mice), network connections (network cards, ISDN cards, modems, DSL connections), network clients and services (NFS, NIS), as well as a general system options (language, partitioning, software, bootloader). This feature can be added only to the Desktop image.

For more information on the IncludeSpecificationList element, see “IncludeSpecificationList” on page 133. For a sample Image Specification Document, see Section C.3.2, “Defined ImageSpecification.xml Document,” on page 240.

Adding Drivers

The Novell Linux Point of Service XML schema allows you to include or exclude specific drivers from a client image. To include or exclude drivers, you must reference the driver within a parent or child Image Specification Document. The DriverSpecifications element identifies the drivers that you want to include or exclude from the image.

Novell Linux Point of Service includes the driver packages listed in Table 9-12.

Table 9-12 *Driver packages included with Novell Linux Point of Service*

Driver Set	URI	Driver Name
General Drivers	/kernel/	◆ net/packet/* ◆ fs/ext3/* ◆ fs/jbd/* ◆ drivers/ide/* ◆ drivers/dump/* ◆ drivers/char/dcs/* ◆ drivers/cdrom/* ◆ drivers/input/keyboard/ikbps.ko ◆ drivers/pci/*
Network Drivers	/kernel/drivers/net/	◆ pcnet32.ko ◆ mii.ko ◆ natsemi.ko ◆ tulip/tulip.ko ◆ eepro100.ko ◆ e100.ko ◆ e1000/e1000.ko
USB Drivers	/kernel/drivers/usb/	◆ host/uhci-hcd.ko ◆ host/ohci-hcd.ko ◆ core/usbcore.ko ◆ host/ehci-hcd.ko ◆ storage/usb-storage.ko ◆ nput/hid.ko
SCSI Drivers	/kernel/drivers/scsi/	◆ scsi_mod.ko ◆ sg.ko ◆ sd_mod.ko ◆ st.ko ◆ sr_mod.ko

To include these drivers in a client image, you must list the driver name and URI in the DriverIncludeList element. Conversely, to exclude a driver, you must list the driver in the DriverExcludeList element.

For example, to include network drivers but exclude the `st.ko` SCSI driver from a client image, you would define the DriverSpecifications element as follows:

```
<DriverSpecifications>
   <DriverIncludeList>
      <DriverSet Name="netdrivers">
         <ExplicitPath URI="/kernel/drivers/net/"/>
            <DriverList>
               <Driver Name="pcnet32.ko"/>
               <Driver Name="mii.ko"/>
```

```
                <Driver Name="natsemi.ko"/>
                <Driver Name="tulip/tulip.ko"/>
                <Driver Name="eepro100.ko"/>
                <Driver Name="e100.ko"/>
                <Driver Name="e1000/e1000.ko"/>
             </DriverList>
          </DriverSet>
       </DriverIncludeList>
    <DriverExcludeList>
       <Driver Name="st.ko"/>
    </DriverExcludeList>
</DriverSpecifications>
```

When working with parent and child documents, remember that DriverSpecifications settings in parent documents take precedence over child documents. That means the parent document can exclude any item that is in the include list of a child document or conversely, the parent document can include any item that is in the exclude list of a child document.

For more information on the DriverSpecifications element, see “DriverSpecifications” on page 134. For a sample Image Specification Document with a complete driver definition, see Section C.3.2, “Defined ImageSpecification.xml Document,” on page 240.

Adding RPMs

The Novell Linux Point of Service XML schema allows you to include or exclude RPM packages from a client image. To include or exclude an RPM package, you must reference the RPM within a parent or child Image Specification Document. The RPMSpecifications element identifies the RPMs that you want to include or exclude from the image.

For example, Novell Linux Point of Service includes an RPM package for the VT100 terminal emulator. To include the VT100 terminal emulator in a client image, you must list the name of the RPM package in the RPMIncludeList element. You must also include the URI for a DistributionSource document that defines where the VT100 RPM is located and define the ImageClass.

NOTE: The ImageClass determines the type of image generated from the current Image Specification Document. The available image classes are NLD and SLES. In general, most Point of Service images are created using the NLD image class. The only images that require the SLES image class are POSBranch images.

To exclude an RPM package from a client image, you must list the name of the RPM package in the RPMExcludeList element.

The following example includes the VT100 terminal emulator package and excludes the SCSI package from a NLD-based client image:

```
<RPMSpecifications>
   <RPMIncludeList>
      <RPMSet>
         <RPMList>
            <RPM Name="vt100"/>
         </RPMList>
      </RPMSet>
   </RPMIncludeList>
```

```
   <RPMExcludeList>
      <RPM Name="scsi"/>
   </RPMExcludeList>
   <DistributionSource URI="/opt/SLES/POS/system/templates
      /Distribution.xml" ImageClass="NLD"/>
</RPMSpecifications>
```

When working with parent and child documents, remember that RPMSpecifications settings in parent documents take precedence over child documents. That means the parent document can exclude any item that is in the include list of a child document or conversely, the parent document can include any item that is in the exclude list of a child document.

For more information on the RPMSpecifications element, see "RPMSpecifications" on page 136. For information on the Distribution Source Document, see Section 9.2.3, "Distribution Source Document (Distribution.xml)," on page 140. For a sample Image Specification Document, see Section C.3.2, "Defined ImageSpecification.xml Document," on page 240. For a sample Distribution Source Document, see Section C.4.2, "Defined Distribution.xml Document," on page 245.

Managing Users and Groups within an Image

The UserGroupSpecifications element within an Image Specification Document defines the users and groups that are created within the image. All users and groups that will be logging into the Point of Service terminals configured by the current image can be pre-defined within the image.

The UserGroupSpecifications settings in the parent document take precedence over settings in child documents.

This section outlines how to manage users and groups within the UserGroupSpecifications element.

- "Adding Users to an Image" on page 151
- "Adding Groups to an Image" on page 152
- "Managing the Root Account" on page 152

For more information on the UserGroupSpecifications element, see "UserGroupSpecifications" on page 138.

Adding Users to an Image

To add a user to a client image, you must define the user account within the UserList element in the UserGroupSpecifications element. User attributes include the user name, encrypted password, userID, home directory, main group, and group associations.

NOTE: This EncrptedPassword value in the UserList element must be the encrypted password. You must use an encryption utility to encrypt the user account password before it can be used as the value for this attribute.

The following example creates a user account for Mandy Post in the client image:

```
<UserGroupSpecifications>
   <UserList>
      <User Name="Mandy Post"
       EncryptedPassword="d41d8cd98f00b204e9800998ecf8427e"
       HasPassword="true" UserId="mpost" HomeDirectory="/home/mpost"
```

```
        MainGroup="Admin">
       <GroupAssociation Name="All"/>
       <GroupAssociation Name="Admin"/>
       </User>
   </UserList>
</UserGroupSpecifications>
```

Adding Groups to an Image

To add a group to a client image, you must define the group within the GroupList element in the UserGroupSpecifications element. Group attributes include the group name and ID.

The following example creates the All and Admin groups in the client image:

```
<UserGroupSpecifications>
   <GroupList>
      <Group Name="All" GroupId="001"/>
      <Group Name="Admin" GroupId="002"/>
   </GroupList>
</UserGroupSpecifications>
```

Managing the Root Account

The RootSettings element within the UseGroupSpecifications element allows you to manage the Root account for all Point of Service terminals configured with the current client image. Using the DisableRootAccess element, you can disable the Root account. Using the EncryptedRootPassword element, you can specify the encrypted password for the Root account.

NOTE: This EncrptedPassword value in the RootSettings element must be the encrypted password.

The following syntax in the Image Specification Document disables the Root user account:

```
<UserGroupSpecifications>
   <RootSettings DisableRootAccess="true"/>
</UserGroupSpecifications>
```

The following syntax in the Image Specification Document defines a password for the Root user account:

```
<UserGroupSpecifications>
   <RootSettings DisableRootAccess="false"
    EncryptedRootPassword="encrypted_password"/>
</UserGroupSpecifications>
```

IMPORTANT: The EncryptedRootPassword element is only valid if the Root account is enabled.

Changing the Image Language

To define the language deployed on the Point of Service terminal:

1. Define the language in the Locale attribute. The Locale attribute contains a comma-separated list of valid locale names and is stored in the ImageSpecification element within the Parent Image Specification Document.

   For example, the following syntax sets the client image locale, time zone, and keytable to German:

```
<ImageSpecification xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="/opt/SLES/POS/
ImageSpecification.xsd" SchemaVersion="1" SchemaRevision="1"
ImageName="java" ImageVersion="2.0.21" Type="ext2"
ImageType="diskful" Timezone="Europe/Berlin" Locale="en_US, de_DE"
Keytable="de.map.gz">
```

Novell Linux Point of Service provides support for the following locales:

- de_DE (German)
- en_US (US English)
- es_ES (Spanish)
- fr_FR (French)
- it_IT (Italian)
- ja_JP (Japanese)
- ko_KR (Korean)
- pt_PT (Portuguese)
- zh_CN (Simplified Chinese)
- zh_TW (Traditional Chinese)

For more information on the ImageSpecification element, see "ImageSpecification" on page 130.

2 Add the child Image Specification Documents required to support the client image add-on features for each language (other than English) designated in the Locale attribute. The child documents must be added to the parent Image Specification Document's IncludeSpecificationList element.

NOTE: The English language files are included in the add-on feature image; therefore, they do not need to be separately added to the client image.

For example, if de_DE is specified in the Locale attribute and the client image includes the Mozilla browser and GNOME desktop add-on features, you must include the German versions of `gnome.xml` and `mozilla.xml` as child documents in the parent document's IncludeSpecifications element as follows:

```
<IncludeSpecificationList>
   <IncludeSpecification URI="/opt/SLES/POS/system/templates
    /addons/gnome.xml"/>
   <IncludeSpecification URI="/opt/SLES/POS/system/templates
    /addons/mozilla.xml"/>
   <IncludeSpecification URI="/opt/SLES/POS/system/templates
    /locale/de_DE/gnome.xml"/>
   <IncludeSpecification URI="/opt/SLES/POS/system/templates
    /locale/de_DE/mozilla.xml"/>
</IncludeSpecificationList>
```

The locale Image Specification Documents are located in the `/opt/SLES/POS/system/templates/locale/`*`locale/`* directories.

Table 9-13 lists the child Image Specification Documents provided for each supported locale.

Table 9-13 *Image Specification Documents for supported locales*

/opt/SLES/POS/system/templates/locale/	
de_DE	German Locale Documents
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
mozilla.xml	This Image Specification Document provides the language files required to support the Mozilla browser in Desktop or Browser client images.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
es_ES	Spanish Locale Documents
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
fr_FR	French Locale Documents
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.

kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
it_IT	Italian Locale Documents
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
ja_JP	Japanese Locale Documents
browser.xml	This Image Specification Document provides the language files required to support the NLD-based Browser image.
browser-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Browser image.
desktop.xml	This Image Specification Document provides the language files required to support the NLD-based Desktop image.
desktop-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Desktop image.
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
java.xml	This Image Specification Document provides the language files required to support the NLD-based Java image.
java-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Java image.

kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
mozilla.xml	This Image Specification Document provides the language files required to support the Mozilla browser in Desktop or Browser client images.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
ko_KR	Korean Locale Documents
browser.xml	This Image Specification Document provides the language files required to support the NLD-based Browser image.
browser-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Browser image.
desktop.xml	This Image Specification Document provides the language files required to support the NLD-based Desktop image.
desktop-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Desktop image.
java.xml	This Image Specification Document provides the language files required to support the NLD-based Java image.
java-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Java image.
mozilla.xml	This Image Specification Document provides the language files required to support the Mozilla browser in Desktop or Browser client images.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
pt_PT	Portuguese Locale Documents
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
zh_CN	Simplified Chinese Locale Documents

browser.xml	This Image Specification Document provides the language files required to support the NLD-based Browser image.
browser-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Browser image.
desktop.xml	This Image Specification Document provides the language files required to support the NLD-based Desktop image.
desktop-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Desktop image.
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
java.xml	This Image Specification Document provides the language files required to support the NLD-based Java image.
java-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Java image.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
zh_TW	Traditional Chinese Locale Documents
browser.xml	This Image Specification Document provides the language files required to support the NLD-based Browser image.
browser-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Browser image.
desktop.xml	This Image Specification Document provides the language files required to support the NLD-based Desktop image.
desktop-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Desktop image.
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
java.xml	This Image Specification Document provides the language files required to support the NLD-based Java image.

java-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Java image.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.

Setting Image Configuration Settings

The current Novell Linux Point of Service XML schema allows you to configure the following image settings within the ImageSpecification element in the `ImageSpecification.xml` document:

- Image Name
- Schema Version
- Schema Revision
- Image Type (diskful or diskless)
- Image Version
- Add-on Size
- File System Type (ext2 or ext3)
- Time Zone
- Console Keymap

For an explanation of each setting, see “ImageSpecification” on page 130.

The following syntax in the Image Specification Document creates the Java 2.0.21 diskful image with an ext2 file system. The image time zone is Mountain Standard time, the locale is US English, and the console keymap is US.

NOTE: xscr replaces the ImageName and ImageVersion attributes with the name specified on the command line when the Image Description Tree is generated.

```
<ImageSpecification xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="/opt/SLES/POS/
ImageSpecification.xsd" SchemaVersion="1" SchemaRevision="1"
ImageName="java" ImageVersion="2.0.21" Type="ext2" ImageType="diskful"
Timezone="US/Mountain" Locale="en_US" Keytable="us.map.gz">
```

9.4.3 Configuring the Image

Configuring an image means adapting it for a specific hardware environment. This includes activating and deactivating services, setting up special Post-install scripts, adding standard configuration files and setting the time zone.

The following sections review these image configuration options:

- “Including Fixed Configuration Files” on page 113
- “Using Data Images to Manage External Configuration Files” on page 113
- “Activating and Deactivating System Services” on page 116
- “Writing Post-Install Scripts” on page 116

Including Fixed Configuration Files

A fixed configuration is a configuration file that provides information for a service that is hardware independent. Fixed configuration files are stored in the Image Description Tree under the files-user subdirectory.

The following instructions illustrate how to add the fixed configuration file, `/etc/sysconfig/hotplug`, to the Image Description Tree, `/opt/SLES/POS/system/myImage-1.1.1`:

1. Go to the `source/files-user` directory of the image:
   ```
   cd /opt/SLES/POS/system/myImage-1.1.1/files-user
   ```
2. Create the directory structure according to the original system location of the configuration file:
   ```
   mkdir -p etc/sysconfig
   ```
3. Create the configuration file within the `files-user` tree.

   In this case, simply copy the file from the real system to the image tree:
   ```
   cp /etc/sysconfig/hotplug etc/sysconfig
   ```

The file tree within `files-user` is completely copied to the image when it is generated. For more information on the `files-user` directory, see Section 9.2.1, “Image Description Tree,” on page 126

Using Data Images to Manage External Configuration Files

A data-only image is an ext2 image file that contains only a copy of the Image Description Tree starting at the given directory. This kind of image cannot be used as operating system or boot image. However, it can be used to add external configuration files to a Point of Service terminal.

If a disk-based system is booting and the IMAGE parameter in the `config.`*`MAC_address`* file includes an additional data image that will be downloaded to a /dev/ram*x* device, the data contents are automatically included into the system. If a data image is downloaded into a partition on the disk, the data is available at the mount point referring to the contents of the PART variable.

The advantage of using data images to add external configuration files to a Point of Service terminal is that the data image is controlled in the same way as the client image. This means you can manage, modify and refresh the configuration files independent of the client image.

IMPORTANT: To implement this functionality, you must manually modify the `config.`*`MAC_address`* file for each Point of Service terminal that you want to load the data image. However, when you run `posAdmin --updateconfig` or `posldap2crconfig.pl --dumpall` to refresh the `config.`*`MAC_address`* files on the Branch Server, these modifications are overwritten. Therefore, to maintain the functionality, you must manually reconfigure the `config.`*`MAC_address`* files each time you regenerate them. For more information on the `posldap2crconfig.pl` command, see Section A.3.5, “posldap2crconfig.pl,” on page 206. For more information on the `posAdmin --updateconfig`

command, see Section 6.9, “Updating config.MAC_address and Hardware Configuration Files,” on page 85.

The following instructions illustrate how to manage external configuration files with a data image:

1. Create a temporary directory that contains the data.

   ```
   mkdir /tmp/mydata
   ```

2. Create the directory structure according to the original system location of the configuration file below this data directory and apply your configurations.

   ```
   mkdir -p /tmp/mydata/etc/X11
   vi /tmp/mydata/etc/X11/XF86Config
   ```

3. Create a data image.

   ```
   xscr --create-data-image /tmp/mydata \
       --image mydata-2.0.21 --destdir /tmp/myDataDirectory
   ```

   This call creates the data image, `mydata-2.0.21`, and the referring MD5 sum in `/tmp/myDataDirectory/`.

4. Copy the image to the `/opt/SLES/POS/rsync/image/` directory on the Administration Server.

   IMPORTANT: The data image must be copied to the `/opt/SLES/POS/rsync/image` directory on the Administration Server before the Branch Server can distribute it to Point of Service terminals.

5. To activate the data image, add the data image to the IMAGE parameter in the `config.MAC_Address` file.

   The IMAGE entry might appear as follows:

   ```
   IMAGE=/dev/hda2;minimal;1.1.8;192.168.100.1;1024,
         /dev/ram2;mydata;2.0.21;192.168.100.1;1024
   ```

   IMPORTANT: To ensure the contents of the data image are copied to the system, the image must be downloaded to a /dev/ram*x* device.

With the data image listed as an IMAGE entry in the `config.MAC_address` file, the data image contents are copied to the Point of Service terminal after the data image has been downloaded to /dev/ram2.

To update the data image on the Point of Service terminal, do the following:

1. Generate a new version of the data image.
2. Copy the new data image version to the `/opt/SLES/POS/rsync/image/` directory on the Administration Server.
3. Run `possyncimages.pl` to download the image to the Branch Server.
4. Modify the IMAGE entry in the `config.MAC_Address` file to reflect the data image’s new version number.

Enabling DMA on Point of Service Terminal CD Drives

Setting up a Direct Memory Access (DMA) channel for the CD drive on your Point of Service terminals speeds up the process of booting and loading an image from CD. The CDBoot image template provided with Novell Linux Point of Service includes the RPM package (hdparm) required

to enable DMA so that the DMA channel is configured when the terminal boots from CD. However, if you would like DMA to be enabled beyond the initial install, you must add the DMA feature to the client image.

To add DMA functionality to a client image:

1. Add the hdparm package to the RPMIncludeList element in the Image Specification Document as follows:

```
<RPMSpecifications>
   <RPMIncludeList>
      <RPMSet>
         <RPMList>
            <RPM Name="hdparm"/>
         </RPMList>
      </RPMSet>
   </RPMIncludeList>
   <DistributionSource URI="/opt/SLES/POS/system/templates
    /Distribution.xml" ImageClass="NLD"/>
</RPMSpecifications>
```

   NOTE: The CDBoot Image Specification Document includes the hdparm RPM package by default.

2. Add the CD device (usually /dev/hdc) to the DEVICES_FORCE_IDE_DMAflag in the `/etc/sysconfig/ide` file. For example:

   `DEVICES_FORCE_IDE_DMA="/dev/hdc:on"`

3. Provide a way for the `/etc/sysconfig/ide` file to be deployed on the Point of Service terminal.

   This can be accomplished in one of two ways:

   - Add the `/etc/sysconfig/ide` file to the `/opt/SLES/POS/system/`*`image_name-version`*`/files-user/` directory in the Image Description Tree.
   - Create an scConfigFileTemplate or scConfigFileSyncTemplate object under the scPosImage object associated with this image or under the scCashRegister object associated with the Point of Service terminals that use this image. For more information on this procedure, see Section 6.4.2, “Adding an scConfigFileTemplate Object,” on page 76 or Section 6.4.3, “Adding an scConfigFileSyncTemplate Object,” on page 77.

4. Build the image.

   For more information on this procedure, see Section 9.4.4, “Building the Image,” on page 162.

Activating and Deactivating System Services

System services are activated or deactivated in the `config.system` file by using the `insserv` command to set or remove links.

To activate a service, add the following line to the `config.system` file:

`sbin/insserv /etc/init.d/`*`service`*

To deactivate a service, add the following line to the `config.system` file:

`sbin/insserv -r /etc/init.d/`*`service`*

For more information on the `config.system` file, see Section 9.2.1, “Image Description Tree,” on page 126.

Writing Post-Install Scripts

A Post-install script is always bound to a package from the setup file and is usually used to remove items from the package that are not needed for the image. This type of script must have the same name as the corresponding package and is stored in the script directory of the Image Description Tree (`opt/SLES/POS/system/image_name-version/script/`). The script itself is called within the image environment, which means it is not possible to damage the host system with your script even if you are using absolute paths.

A Post-install script uses the following format:

```
\#!/bin/sh
echo -n "Image [image_name_version]..."
test -f /.profile \&\& . /.profile

... script code

echo done
```

image_name-version is the name of the image to which this script belongs.

For more information on the script directory, see Section 9.2.1, “Image Description Tree,” on page 126.

9.4.4 Building the Image

xscr builds images using an Image Specification Document (`ImageSpecification.xml`) and a Distribution Source Document (`Distribution.xml`). These documents perform the same function as the IMAGE, VERSION, config, setup, and setup.user files in the Image Description Tree. They contain XML elements that define the structure, configuration files, and other components required to build client images.

If the Image Specification file is complete and located at the root of the Image Description Tree (`/opt/SLES/POS/system/image_name-version/`), you can generate the corresponding client image.

For example, the following xscr command creates the XML Desktop image with the version 2.0.6 in the working directory myImages and verifies the RPMs (type the command all on one line):

```
xscr --prepare --image desktop-2.0.6 --nostrict --build
--destdir myImages --verify
```

IMPORTANT: xscr only maintains five builds of a single image in the same directory. When you generate the sixth build of an image, xscr deletes the oldest image version. If you want to maintain more than five versions of a single image, you must maintain them in separate directories.

9.5 Distributing Images

To electronically distribute new or updated client images, you must first copy the images into the central RSYNC directory of the Administration Server and then transfer the images to the Branch Servers.

This section reviews each step in the electronic distribution process.

- Section 8.5.1, "Copying Images to the Administration Server RSYNC Directory," on page 117
- Section 8.5.2, "Distributing Images to the Branch Server," on page 118
- Section 8.5.3, "Distributing Images to Point of Service Terminals," on page 119
- Section 8.5.4, "Image Install Notification," on page 119

NOTE: If you are unable to electronically distribute Point of Service images over your network, you must manually distribute the images uses CDBoot images. For more information on creating a CDBoot image, see Section 10.1, "Building a CDBoot Image," on page 171.

9.5.1 Copying Images to the Administration Server RSYNC Directory

The first step to distribute new client images is to copy the images from the `/opt/SLES/POS/image` directory to the RSYNC directory, `/opt/SLES/POS/rsync`. Client images must be located in the `/opt/SLES/POS/rsync/image` directory on the Administration Server before the RSYNC service can transmit the images to the Branch Server. Boot images must be located in `/opt/SLES/POS/rsync/boot`.

NOTE: Copying the client images to the RSYNC directory is done manually to control which client image types and versions are distributed to the Branch Servers.

Copying Client Images to the Administration Server's RSYNC Directory

The following example demonstrates how to put a previously-extended Browser client image in the Administration Server's RSYNC directory so it can be received, on request, by the Branch Server:

1. Copy the extended Browser client image:

   ```
   cp /opt/SLES/POS/image/myBrowser-2.3.10-2006-08-06
      /opt/SLES/POS/rsync/image/browser-2.3.10
   ```

2. Copy the corresponding Browser image MD5 checksum file:

   ```
   cp /opt/SLES/POS/image/myBrowser-2.3.10-2006-08-06.md5
      /opt/SLES/POS/rsync/image/myBrowser-2.3.10.md5
   ```

Copying Boot Images to the Administration Server's RSYNC Directory

The following example demonstrates how to copy the first and second stage boot images to the Administrations Server's RSYNC directory so they can be received, on request, by the Branch Server:

NOTE: Point of Service terminals boot two images, a first stage image (`initrd.gz`) and a second stage image (`linux`). For more information, see Section 3.6, "Booting the Point of Service Terminal," on page 35.

---

1. Copy the initrd-disknetboot image as `initrd.gz`:

   ```
   cp /opt/SLES/POS/image/initrd-disknetboot-version-date.gz
      /opt/SLES/POS/rsync/boot/initrd.gz
   ```

2. Copy the kernel image as `linux`:

```
cp /opt/SLES/POS/image/initrd-disknetboot-version-
date.kernel.kernel_version /opt/SLES/POS/rsync/boot/linux
```

9.5.2 Distributing Images to the Branch Server

If you create a new image or change an image version, you can run the `possyncimages.pl` script at the Branch Server to transfer new or updated images to the Branch Server after the images are in the Administration Server's RSYNC directory.

IMPORTANT: The RSYNC service must be properly configured and running on the Administration Server for the `possynimages.pl` script to run. For more information, see Section 6.3.3, "Adding an scServerContainer and scBranchServer Object," on page 69.

Additionally, each client image must have an associated scPosImage object in LDAP and the object's scPosImageVersion attribute must be set to active before `possyncimages.pl` will transfer the images to the Branch Server. For more information, see Section 6.5.2, "Activating Images," on page 81.

The basic process is as follows:

1. The `possyncimages.pl` script initially checks via the PID file to determine if an instance is already running.
2. The image files are then copied from the Administration Server to the Branch Server. Boot images are copied from the `/opt/SLES/POS/rsync/boot` directory on the Administration Server to the `/tftpboot/boot` directory on the Branch Server. Client images and their associated MD5 checksum files are copied from `/opt/SLES/POS/rsync/image` to `/tftpboot/image`.

For more information on the possyncimages, see Section A.3.10, "possyncimages.pl," on page 208.

IMPORTANT: Remember that distributing client images from the Administration Server to the Branch Servers is only one part of the process required to deploy new versions of a client image. You must also update the scPosImageVersion attribute within the Image Reference object (scPosImage) in the LDAP tree. Otherwise Point of Service terminals already registered in LDAP cannot boot the new client image version. For more details, refer to Section 6.5, "Managing Image Objects," on page 79 and Section A.3.5, "posldap2crconfig.pl," on page 206.For an illustration of Novell Linux Point of Service system dependences, see Section 1.2, "Dependencies Between LDAP, Branch Server, and Point of Service Terminal," on page 15.

After executing the possyncimages, verify the result by checking the contents of the following directories:

- `/tftpboot/image`
- `/tftpboot/boot`

9.5.3 Distributing Images to Point of Service Terminals

New or updated images are distributed to Point of Service terminals at boot time. For information on this process, see Section 3.6, "Booting the Point of Service Terminal," on page 35.

9.5.4 Image Install Notification

When the Branch Server distributes a new image to a Point of Service terminal, the system provides notification that the image was successfully installed on the Point of Service terminal. The notification is stored in the scWorkstation object in the LDAP directory on the Administration Server.

When the image is successfully installed on the Point of Service terminal, the linuxrc script running on the Point of Service terminal creates the *bootversion.MAC_Address* file in the `/tftpboot/upload` directory on the Branch Server. posleases2ldap then transfers the information to the scNotifiedimage attribute in the scWorkstation object in LDAP and deletes the *bootversion.MAC_address* file.

9.6 Incremental Update

With the image description diff tool (xdscr) included in the Novell Linux Point of Service 9 Patch Release and later releases, you have the ability to perform an incremental update of a client image. This feature lets you update the software on a Point of Service terminal without downloading an entire new image from the Branch Server. It is designed to make it easier to update your client images when new RPM modules or patches are released.

NOTE: In Novell Linux Point of Service 9 SSP3, a new `--delta` option was added to the `xscr` command to perform the same functionality as the `xdscr` command.

The process of doing an incremental update can be divided into three main steps:

- Section 9.6.1, “Creating the Delta Image File,” on page 165
- Section 9.6.2, “Adding the Delta Image Object in LDAP,” on page 166
- Section 9.6.3, “Copying the Delta Image Files to the Branch Server,” on page 167

9.6.1 Creating the Delta Image File

The command used to create the delta image files compares two images and builds a tarball containing a list of RPMs that need to be updated, a tarball checksum file, and a script to install the updated RPMs.

Before you run the command, you must create an updated image with your build distribution list pointing to the new software. To do this, follow the instructions in Section 9.4, “Building Images with xscr,” on page 144.

After you have created the updated image, run either of the following commands to create the delta image file:

```
xdscr --image old_image --with new_image --destdir directory
```

or

```
xscr --delta --image old_image --with new_image --destdir directory
```

Specify the image names in the format *image_name-version*; for example, `minimal-2.3.10`. You can abbreviate the options as follows:

- Substitute `-i` for `--image`

- Substitute `-w` for `--with`
- Substitute `-d` for `--destdir`

For example, the following command compares the minimal-2.3.10 Image Description Tree with the browser-2.3.10 Image Description Tree and saves the diff file to the `/home` directory:

```
xdscr -i minimal-2.3.10 -w browser-2.3.10 -d /home
```

The following is an excerpt from the resulting diff file:

```
#!/bin/bash
##Automatically generated by xdscr image diff tool

# /opt/SLES/POS/dist/NLD9/SP3/CD1/suse/i586/rsync-2.6.2
    -8.18.i586.rpm
# /opt/SLES/POS/dist/NLD9/FCS/CD1/suse/i586/libtool-1.5.2
    -56.2.i586.rpm
# /opt/SLES/POS/pac/IBMJava2-JRE-1.4.2-0.68.i586.rpm
# /opt/SLES/POS/pac/IBMJava2-JAVACOMM-1.4.2-0.21.i586.rpm

rpm -Uh rsync-2.6.2-8.18.i586.rpm \
        libtool-1.5.2-56.2.i586.rpm \
        IBMJava2-JRE-1.4.2-0.68.i586.rpm \
        IBMJava2-JAVACOMM-1.4.2-0.21.i586.rpm
```

In addition to the diff file, a tarball delta file and an MD5 file for the delta image are saved in the destination directory. The delta and MD5 filenames begin with `delta-` and contain the names of the old and new image files, followed by the date in the form yyyy-mm-dd and the appropriate extension (`.tar` and `.md5`). The diff filename begins with `diff-` and ends with the date.

NOTE: Both old and new image files must contain the RPM database in order for the delta image to be created successfully. By default, Minimal images have the RPM database stripped out. Before creating a delta image of an existing Minimal image, re-create the image using the `--keep-rpm` option and specify the resulting image as the old image in the `xdscr` command.

You can also use build distribution lists as inputs instead of the actual image files.

9.6.2 Adding the Delta Image Object in LDAP

The next step after creating the delta image is to add a corresponding object in the LDAP directory. This includes adding a new scPosDeltaImage object and setting the scPosDeltaImageDn attribute on the appropriate scCashRegister object.

Adding a New scPosDeltaImage Object

Use the `posAdmin` command to add a new image object in LDAP for the delta image. The syntax of the command is as follows (type the command all on one line):

```
posAdmin.pl --user admin_user --password password
--base base_dn --add --scPosDeltaImage --cn common_name
--scImageName name --scImageFile tarball_filename --scBsize size
```

For the --cn and --scImageName options, choose a short descriptive name that identifies the delta image. For --scImageFile, use the tarball filename without the date. For --scBsize, use the value displayed at the end of the .md5 file.

For example, to add an object for a delta image tarball file named delta-minimal-2.3.10-browser-2.3.10-2006-08-06.tar, enter the following command:

```
posAdmin.pl --user cn=admin,o=novell,c=us --password secret
--base cn=default,cn=global,o=novell,c=us --add --scPosDeltaImage
--cn minimal_browser_delta --scImageName minimal_browser_delta
--scImageFile delta-minimal-2.3.10-browser-2.3.10.tar
--scBsize 8192
```

Setting the scPosDeltaImageDn Attribute

Use the posAdmin command to set the scPosDeltaImageDn attribute on the scCashRegister object that the Point of Service terminal is associated with. The syntax of the command is as follows (type the command all on one line):

```
posAdmin.pl --user admin_user --password password
--base base_dn --modify --scCashRegister --multival
--scPosDeltaImageDn 'image_dn' --DN crtype
```

For *image_dn*, specify the DN of the scPosDeltaImage object, enclosed in single quotes. For *crtype*, specify the DN of the crtype object you are modifying.

For example, to set this attribute for the delta image object created above, enter the following command:

```
posAdmin.pl --user cn=admin,o=novell,c=us --password novell
--base cn=default,cn=global,o=novell,c=us --modify
--scCashRegister --multival --scPosDeltaImageDn
'=>cn=minimal_browser_delta,cn=default,cn=global,o=novell,c=us'
--DN cn=crtype3,cn=global,o=novell,c=us
```

9.6.3 Copying the Delta Image Files to the Branch Server

To get the new delta image files to the necessary Branch Server, you must first rename the tarball and checksum files and copy them to the /opt/SLES/POS/rsync/image directory on the Admin Server. From there, you can distribute them to the Branch Server in the usual way.

1. Rename the .tar and .md5 files for the delta image by removing the date portion of the filenames.

   For example, if the tarball file is named delta-minimal-2.3.10-browser-2.3.10-2006-08-06.tar, rename the file to delta-minimal-2.3.10-browser-2.3.10.tar.

2. Copy the renamed delta image files to the /opt/SLES/POS/rsync/image directory on the Admin Server.

   For example, to copy the delta image files mentioned in Step 1, enter:

   ```
   cp delta-* /opt/SLES/POS/rsync/image
   ```

3 On the Branch Server that the Point of Service terminals get their boot information from, run the following command to download the new image files to the Branch Server:

```
possyncimages.pl
```

4 On the Branch Server, run the following command to have the Branch Server update the configuration files for the Point of Service terminals on its subnet:

```
posldap2crconfig.pl --dumpall
```

The next time the Point of Service terminals reboot, linuxrc detects the delta image files and automatically installs the updated RPMs after loading the base image.

9.7 Updating the Product File in a Boot Image

The DiskNetboot image contains a product file that lists the network and storage drivers to be used for particular Point of Service terminal hardware types. The PXE boot routines use the driver internal to the network card to download the `initrd.gz` and `linux` files from the Branch Server. However, when the kernel is executed, it needs to find a working Linux network driver in order to download the actual image to be installed.

To find an optimal network driver, the Point of Service terminal first searches the product file for an entry that matches its Product ID. If none is found, the terminal cycles through various network drivers trying to find one that loads, which is not always successful. If your particular hardware is not in the product file and the terminal can't find the correct driver by trial and error, you can add an entry for your hardware in the product file.

In Novell Linux Point of Service 9 SSP3, you can update the product file without having to rebuild the DiskNetboot image, using either a standalone utility or a new option for the `xscr` command.

The syntax for the `posUpdateProductFile.sh` standalone utility is:

```
posUpdateProductFile.sh path_and_image_name-version
path_and_new_product_file
```

The syntax for the `xscr` command is:

```
xscr --update-product-file --image path_and_image_name-version
--with path_and_new_product_file
```

Use these commands as follows:

1 Create or obtain an updated product file.

The product file that is contained within the DiskNetboot image is named `IBMproduct`. It is a text file that lists each Product ID (PID) and its corresponding network driver and, optionally, storage drivers.

The following excerpt is from the `IBMproduct` file that ships with Novell Linux Point of Service 9:

```
IBM46942X5 net=pcnet32
IBM4810X3X  net=e100 storage=scsi_mod,sd_mod,libdata,sata_sil
```

You can add your own PID and driver entries to this file or you can create a new product file.

Save the new product file in a directory on the Administration Server.

2 On the Administration Server, run either `posUpdateProductFile.sh` or `xscr` with the `--update-product-file` option, providing the path and name of the DiskNetboot image and the path and name of the new product file as parameters.

The utilities do the following:

- Uncompress the DiskNetboot image.
- Replace the existing product file (`IBMProduct`) in the image with the new one.
- Recompress the image.
- Copy the new product file into the DiskNetboot Image Description Tree (if one exists) so that it is available for future image building.

NOTE: The `posUpdateProductFile.sh` and `xscr --update-product-file` commands work only with bootable images (DiskNetboot, CDboot, or any images extended from them). User-customized images must contain "boot" as part of the image name.

Building Specialized Images

10

This section reviews the procedures required to build the following Novell® Linux Point of Service specialized images:

- Section 10.1, “Building a CDBoot Image,” on page 171
- Section 10.2, “Building POSBranch Images,” on page 176
- Section 10.3, “Building an Automatic Branch Server Installation Image,” on page 179

10.1 Building a CDBoot Image

In environments where no network infrastructure is available to boot Point of Service systems over the LAN, you can use boot CDs. Boot CDs are also required to deploy POSBranch Servers.

The ImageBuilder tool includes an option to generate CDBoot images. ImageBuilder generates an ISO 9660-compliant CD image that is bootable according to the El Torito specification. The resulting CD contains a minimal Linux system image (CDBoot), a Linux system client image (Minimal, Java, Browser, or Desktop), and a `config.image` configuration file. The configuration file controls whether the client image is written into a RAM disk or if it must be placed on the hard disk of the booting node.

To build a CDBoot image, you must complete the following steps:

1. Prepare the client image you want to build with the CDBoot image and generate a test build.
2. Create the CD setup directory.
3. Create the config.image file.
4. Generate the CDBoot image and its associated client image.
5. Generate the CD ISO image.
6. Boot the CDBoot Image.

These steps are discussed in the following sections.

10.1.1 Preparing the Client Image

Before creating a CD ISO image, prepare the following:

- Select which client image you want to build with the CDBoot image (Minimal, Java, Browser, Desktop, POSBranch, or a custom image).
- Before you create the CDBoot image, generate a test build of the client image to verify there are no problems with the image.

For information on creating a preparing and generating client images, see Section 8.4, “Building Images with scr,” on page 109 or Section 9.4, “Building Images with xscr,” on page 144.

10.1.2 Creating the CD Setup Directory

When you create a CDBoot image, all files must be part of the CD and therefore, must be centrally located in a setup directory. Therefore, the first step in preparing the CDBoot image is putting all the configuration files required for the CDBoot image in a centralized setup directory. This includes the `config.image` file and any other files required to configure the Point of Service terminal such as XF86Config files.

Use the following command to create a CD setup directory:

`mkdir /tmp/`*`cdsetup_directory`*

After you create the CD setup directory, create the `config.image` file and copy over any other configuration files required for the CDBoot image.

10.1.3 Creating the config.image File

After creating the CD setup directory, you can create the `config.image` file. This file contains the parameters required to configure a specific Point of Service terminal during a CDBoot; that is, it indicates which client image the CDboot boot image should load and how to do it. It is an ASCII, line-based file that can be created in a text editor. It must be named “config.image” and it must be located in the CD setup directory.

The format of the `config.image` file is as follows:

```
IMAGE=device;name;version;compressed
CONF=source;dest,...,source;dest
PART=size;id;Mount,...,size;id;Mount
JOURNAL=ext3
DISK=device
FEATURE=<contents of the --feature option>
EXTEND=<contents of the --extend option>
PARAMS=<additional options of type bool>
```

Table 10-1 provides a detailed description of each parameter in the `config.image` file.

Table 10-1 *config.image file parameters*

Parameter	Variable	Description
IMAGE		Specifies the client image (*image*) and version (*version*) that will be loaded on the Point of Service terminal. When you generate the CDBoot image, ImageBuilder uses this information to generate the client image with the CDBoot image.
	device	The storage device to which the image is linked; for example, /dev/ram1 or /dev/hda2. **NOTE:** The Point of Service terminal partition hda2 defines the root file system (/) and hda1 is used for the swap partition. The RAM device should not be confused with the hard disk device which uses a partition table. On the RAM disk device, /dev/ram0 is used for the initial RAM disk and cannot be used as storage device for the client image. We recommend that you use /dev/ram1 for the RAM disk.

Parameter	Variable	Description
	image	The client image to load on the Point of Service terminal.
	version	The version of the client image to load on the Point of Service terminal.
	compressed	Specifies a compressed image boot. If the compressed variable is not included, the standard boot process is used. The the boot fails if you specify Compressed and the image isn't compressed. It also fails if you don't specify “'compressed”' and the image is compressed. **IMPORTANT:** The name of the compressed image must contain the suffix .gz and must be compressed with the gzip tool or by using the --gzip option at create time.
CONF		Specifies the configuration files to download to the Point of Service terminal. The data is provided in a comma-separated list of source:target configuration files.
	source	The path to the source configuration file within the directory.
	dest	An absolute path below the client image where the configuration file is saved.
PART		Specifies partitioning data. The data is provided in a comma-separated list. The first element of the list must define the swap partition. The second element of the list must define the root partition. Each element must include the size (*size*), the type number (*id*), and the mount point (*Mount*).
	size	The size of the partition. If a partition should take all the space left on a disk you can set a lower x letter as the size specification.
	id	The partition type number.
	mount	The partition mount point. **IMPORTANT:** The swap partition must not contain a mount point. Use a lowercase letter x instead.
JOURNAL		Specifies a journal to be added to the file system. The value for this parameter must be set to ext3 because the only journaled file system Novell Linux Point of Service supports is ext3. **NOTE:** If you have an existing ext2 image, you can change the file system by setting a flag in the scCashRegister or the scWorkstation objects rather than recreate the image. If ext3 is specified in either LDAP object, the Point of Service terminal extends the file system to ext3 when the image is deployed. The JOURNAL parameter is evaluated only if the DISK parameter is set.

Parameter	Variable	Description
DISK		Defines the device through which the hard disk can be addressed, for example /dev/hda. This parameter is used only with PART.
FEATURE		The value of FEATURE is the value of the --feature option used for building the client image. For information, refer to “scr Commands” on page 97 or “xscr Commands” on page 121. This optional parameter is only pertinent while the client image is created.
EXTEND		The value of EXTEND is the value of the --extend option used to extend an image with an additional RPM package. For information, refer to “scr Commands” on page 97. **IMPORTANT:** The EXTEND parameter may only be used for CDBoot images generated with scr. The xscr ImageBuilder tool uses the ImageSpecification.xml document to add additional packages to the CDBoot image. This optional parameter is only pertinent while the client image is created.
PARAMS		The value of PARAMS consists of bool options that are used for special actions. The PARAMS parameter is only pertinent while the client image is created. This parameter can be used with the --gzip option to compress the image. The CDboot linuxrc recognizes a compressed image referring to the suffix .gz. A compressed CD image is uncompressed on the fly while the image is installed. For POSBranch images, we recommend that you add the following line to the config.image file: `PARAMS=--keep-rpm` This allows you to use the YaST2 interface to configure POSBranch Servers. However, it does add approximately 30 MB to the size of the image. If the size of the image is an issue, you can leave the RPMs out; however, you will not have YaST2 functionality.

10.1.4 Generating the CDBoot Image

To build the CDBoot image, you must use ImageBuilder’s CD boot feature. This feature requires the CD setup directory as a parameter so ImageBuilder can locate the `config.image` file. It then uses the parameters defined in this file to create the CDBoot image. When you generate the CDBoot image, ImageBuilder also builds the client image designated in the `config.image` file’s IMAGE parameter. For information on the `config.image` file format and parameters, see Section 10.1.3, “Creating the config.image File,” on page 172.

IMPORTANT: Because ImageBuilder automatically generates the client image when it builds the CDBoot image, the Image Description Trees for both the CDBoot and client images must be complete before generating the CDBoot image.

The CDBoot build command syntax is as follows:

```
xscr --prepare --build --image image_name-version
--feature boot_cd:config=CD_setup_directory --destdir directory
```

NOTE: This procedure is written for xscr. However, the processes are the same for scr and xscr; only the executable changes.

For example, the following command creates the CDboot and client images as specified in the `config.image` file in the `/tmp/cdsetup/` directory and saves the images to `/tmp/mycd/`.

```
xscr --prepare --build --image cdboot-2.3.10
--feature boot_cd:config=/tmp/cdsetup --destdir /tmp/mycd
```

NOTE: If you plan to create an `.iso` file of an image that is larger than 650 MB, use the compression option so the resulting image will fit on a standard CD.

10.1.5 Creating the CD ISO Image

After the CDBoot and client images are generated, you are ready to create the final ISO image. ImageBuilder's `--create-iso` option builds the following components in the ISO image:

- The CD directory structure
- All necessary boot manager files
- The CDBoot and client images

The command syntax for creating an ISO image is as follows:

```
xscr --create-iso image_name.iso --destdir directory
```

NOTE: This procedure is written for xscr. However, the processes are the same for scr and xscr; only the executable changes.

For example, the following command creates the mycd ISO image in the `/tmp/mycd/` directory.

```
xscr --create-iso mycd.iso --destdir /tmp/mycd
```

IMPORTANT: The `--destdir` option defines the target and the source directory. The CDBoot and client images you previously created must be located in the destination directory. At the end of the process, you will have three images in the directory: the CDBoot image, the client image, and the final ISO image.

When the process is complete, use a CD recording program to burn only the ISO image (`/tmp/mycd/mycd.iso`) to a CD.

10.1.6 Booting the CDBoot Image

After you have burned the CDBoot ISO image to a CD, you can use the CD to boot and configure Point of Service terminals.

NOTE: If there are multiple CD drives in the Point of Service terminal, there is no way to designate which CD drive to use; the system chooses the first one it finds. If the Point of Service terminal does

not find the drive with the boot CD, it returns BIOS errors. To correct the problem, insert the CD in the bootable CD drive.

Depending on the client image (Minimal, Java, Browser, or Desktop) that resides on the boot CD, you should note the following restrictions:

- The Point of Service terminal must be upgraded with enough RAM to hold the client image.
- On diskless Point of Service terminals, there must be enough available RAM to load the first and second stage boot images. Otherwise the terminal returns a kernel panic error.

  NOTE: Keep in mind that onboard VGA reduces the Point of Service terminal's available RAM.

The behavior of Point of Service terminals booting from CD is similar to Point of Service terminals that receive the first and second stage boot image over the LAN from a Branch Server:

1. The client image (for example, the Browser image) is installed to a RAM or hard disk drive on the Point of Service terminal.

   The partition information resides in the `config.image` file located on the CD.

   NOTE: For electronic distributions, the partition information is created based on LDAP entries on the Administration Server,

   ---

2. The installed client image is booted from the RAM or hard disk drive on the Point of Service terminal.

10.2 Building POSBranch Images

For stores where the Branch Server is running only the Point of Service infrastructure (that is, the Branch Server is running no additional applications), the Branch Server can be deployed as a control terminal running on Point of Service hardware.

To build a POSBranch image with xscr, you must complete the following steps:

1. Prepare the Administration Server to create the image.
2. Clone the Image Description Tree.
3. Define a SLES-based Image Specification Document (ImageSpecification.xml) that includes branch.xml as a child document.
4. Build the POSBranch image.
5. Create the ISO image.

These steps are discussed in the following sections.

10.2.1 Preparing the Administration Server

For more information on this process, see Section 9.3, "Getting Ready to Build Images with xscr," on page 143.

10.2.2 Cloning the Image Description Tree

xscr builds images using a specific file system directory structure known as the Image Description Tree. The Image Description Tree provides the XML documents, scripts, configuration files, and other components required to build client images for Point of Service systems.

To create a POSBranch image, you can use any of the client Image Description Trees provided with Novell Linux Point of Service (Minimal, Java, Browser, or Desktop). The Image Description Trees are located at `/opt/SLES/POS/system/image_name-version/`.

To maintain a standardized source tree and simplify the upgrade process, we recommend you maintain the default Image Description Trees provided with Novell Linux Point of Service as master copies. To build a POSBranch image, you can clone one of the default Image Description Trees, then modify the cloned tree.

When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (`--dist nld|sles`). POSBranch images require the SLES distribution. When you define the image distribution as SLES, xscr adds a child document (`image_name-sles.xml`) to the parent Image Specification Document that includes the SLES RPMs in the image.

NOTE: NLD is the default distribution type. Therefore, if you do not specify the image distribution when cloning the Image Description Tree, xscr defaults to NLD.

The basic syntax to clone an Image Description Tree to create a POSBranch image is as follows:

```
xscr --create image_name-version --image image_name-version --dist
sles
```

For example, the following command clones the Desktop-2.0.21 Image Description Tree to create a new SLES-based Image Description Tree named myImage-1.1.1:

```
xscr --create myImage-1.1.1 --image desktop-2.0.21 --dist sles
```

The new Image Description Tree is located at `/opt/SLES/POS/system/myImage-1.1.1`. You can then modify the cloned Image Description Tree as required to create the POSBranch image. For a description of the individual Image Description Tree components, see Section 9.2.1, “Image Description Tree,” on page 126 and Appendix B, “Novell Linux Point of Service Files and Directory Structure,” on page 211.

10.2.3 Adding branch.xml to the Parent Image Specification Document

Image Specification Documents contain XML elements that define the structure, configuration, and other components required to build images for Point of Service terminals. In general, a master Image Specification Document (or parent document) defines general image parameters and individual image subcomponents such as add-on features, custom applications, and so forth are defined in sub-documents referred to as child documents.

The default Image Description Trees provided with Novell Linux Point of Service have a parent Image Specification Document at the root of the tree. After you clone the tree you want to use to build the POSBranch image, you must add branch.xml as a child document in the IncludeSpecificationList element within the parent Image Specification Document.

The syntax to include the branch.xml document in the IncludeSpecificationList element is as follows:

```
<IncludeSpecificationList>
   <IncludeSpecification URI="/opt/SLES/POS/system/templates/support
    /branch.xml"/>
</IncludeSpecificationList>
```

The `branch.xml` document provides the following Branch Server components:

- All the RPMs required for a functional Branch Server.
- The RPM database so YaST Online Update can be used to update the image.
- The Linux Kernel Crash Dump (LKCD) to provide a system for detecting, saving and examining system crashes.
- Branch Server configuration information obtained from the LDAP directory.

10.2.4 Building the POSBranch Image

The POSBranch image must be deployed on a bootable CD. This requires that you generate the POSBranch image with a CDBoot image.

Novell Linux Point of Service provides a default Image Description Tree (`/opt/SLES/POS/system/cdboot-`*`version`*) and Image Specification Document (`/opt/SLES/POS/system/templates/support/cdboot.xml`) for CDBoot images. You can use these components to generate the CDBoot image.

You must also create a `config.image` file. ImageBuilder uses the parameters defined in this file to build the CDBoot and POSBranch images. For information on creating the `config.image` file, see Section 10.1.3, “Creating the config.image File,” on page 172.

To build the CDBoot image, you must use ImageBuilder’s CD boot feature. This feature requires the CD setup directory as a parameter so ImageBuilder can locate the `config.image` file. It then uses the parameters defined in the file to create the CDBoot image. When you generate the CDBoot image, ImageBuilder also builds the client image designated in the `config.image` file’s IMAGE parameter.

IMPORTANT: You must designate the POSBranch image in the `config.image` file’s IMAGE parameter.

The syntax to generate the CDBoot and POSBranch images is as follows:

```
xscr --prepare --build --image image_name-version
--feature boot_cd:config=CD_setup_directory --destdir directory
```

For example, the following command creates the CDboot and client images as specified in the `config.image` file in the `/tmp/cdsetup` directory and saves the images to `/tmp/mycd`.

```
xscr --prepare --build --image cdboot-2.0.21
--feature boot_cd:config=/tmp/cdsetup --destdir /tmp/mycd
```

10.2.5 Creating the CD ISO Image

After the CDBoot and client images are generated, you are ready to create the final ISO image. For instructions on this procedure, see Section 10.1.5, “Creating the CD ISO Image,” on page 175.

10.3 Building an Automatic Branch Server Installation Image

In the two-tiered administration server and branch server architecture, the branch servers are assumed to be in a remote environment, sometimes far from knowledgeable Linux administrators. To simplify this task, a toolkit is provided that enables administrators to create autoinstall media to automatically install and set up branch servers with very little on-site effort.

This functionality is provided through AutoYaST. AutoYaST provides an automatic installation option that allows new branches to be set up at minimal expense. Novell Linux Point of Service provides an AutoYaST control file for the basic setup and the description files are generated from the LDAP directory. The resulting ISO file must then be burned to a CD for deployment at the Branch Server site.

NOTE: Automatic Branch Server images are created only in ISO format; therefore, CD is the only supported media.

To build an Automatic Branch Server Installation image, you must complete the following steps:

1. Prepare the Administration Server to create the image.
2. Create the Branch Server definition in the LDAP directory.
3. Modify the XML template file.
4. Generate the Automatic Branch Server Installation image.
5. Create the boot media.

These steps are discussed in the following sections.

10.3.1 Preparing the Administration Server

Before you can create the Automatic Branch Server installation image, you must complete the following:

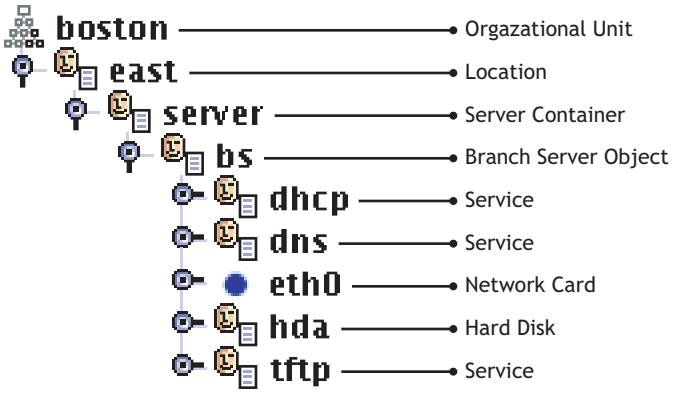
- Install ImageBuilder. For more information, see Section 8.3.1, “Installing ImageBuilder and Image Templates,” on page 108.
- Copy the SLES image source files from the Novell Linux Point of Service CDs to a central distribution directory. For more information, see Section 7.3.1, “Copying the Novell Linux Point of Service CDs,” on page 92.
- Define the AdminServer.conf.sles file. This file is automatically generated when you create AdminServer.conf. For more information, see Section 7.3.4, “Generating AdminServer.conf or Distribution.xml,” on page 94.
- Provide access to a CD writer to create the boot media.

  NOTE: The CD writer does not need to be installed on the administration server if the CD images can be transferred through the network to a machine equipped with a CD recorder.

10.3.2 Creating the Branch Server Definition in the LDAP Directory

To enable the autoinstall system to configure the branch server, detailed information about the hard disk and the network interfaces must be defined in the LDAP directory. Figure 10-1 represents the LDAP objects required to define the structure for a Branch Server.

Figure 10-1 *LDAP objects required for Branch Server definition*



reviews the attributes for each LDAP object required to provide the Branch Server definition. For more information about the LDAP directory, refer to Chapter 5, “The Novell Linux Point of Service LDAP Directory,” on page 55.

Table 10-2 *LDAP objects and attributes for defining a Branch Server*

LDAP Object	Required Attributes
scLocation	The scLocation object defines general information about the Branch Server network. Required attributes include the following: ◆ cn ◆ ipNetworkNumber ◆ ipNetmaskNumber ◆ scDhcpRange ◆ scDhcpFixedRange ◆ scDefaultGw ◆ scDynamicIp For information on adding this object class to the LDAP directory, see Section 6.3.2, “Adding an scLocation Object,” on page 68.
scServerContainer	The scServerContainer object is a container for the Branch Server definition. The only required attribute for this container object is the cn. For information on adding this object class to the LDAP directory, see Section 6.3.3, “Adding an scServerContainer and scBranchServer Object,” on page 69.

LDAP Object	Required Attributes
scBranchServer	The scBranchServer object is a container for the hardware objects that provide the Branch Server definition. The only required attribute for this container object is the cn. For information on adding this object class to the LDAP directory, see Section 6.3.3, “Adding an scServerContainer and scBranchServer Object,” on page 69.
scNetworkcard	The scNetworkCard object provides the configuration for a Branch Server network interface card. Required attributes include the following: ◆ The network device (scDevice) ◆ The IP address of the Branch Server (ipHostNumber) ◆ The loadable module (driver) that is necessary to activate the network card (scModul) ◆ The netmask of the Branch Server's network (ipNetmaskNumber) For example, the following posAdmin command creates a sample scNetworkcard object for a Branch Server: `posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret --base cn=bs,cn=server,cn=Lab,ou=solutions, o=mycorp,c=us --add --scNetworkcard --scDevice eth2 --ipHostNumber 192.168.1.150 --ipNetmaskNumber 255.255.248.0 --scModul e100`

LDAP Object	Required Attributes
scHarddisk	The scHarddisk object provides the configuration for the Branch Server's boot hard disk. Required attributes include the following: ◆ cn ◆ scDevice ◆ scHdSize ◆ scPartitionsTable The partitioning scheme for the Branch Server hard disk is the same as Point of Service terminals. Partitions are specified as 'size type mount point', where size is specified in megabytes, the type is either L for Linux file systems or S for swap space, and the mount point specifies where in the file system hierarchy the partition is mounted. The wildcard "x" must appear as a mount point for swap space partitions and can be used to automatically compute the size of the file system as follows: ◆ S partitions are created at twice the RAM size ◆ L partitions with mount point /boot get approximately 20 megabytes ◆ (Optional) The last partition entry in the list can specify an x wild card for the size parameter to use up the remaining space on the hard disk. Partition entries are separated with a semicolon (;}. For a simple branch server, the partition table x S x;x L / is suggested, which creates swap space and one large root file system. For example, the following posAdmin command creates a sample scHarddisk object for a Branch Server: `posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret --base cn=bs,cn=server,cn=Lab,ou=solutions, o=mycorp,c=us --add --scHarddisk --cn sda --scDevice dev/sda --scHdSize 40960 --scPartitionsTable 'x S x;x L /'`
scService	The scService object defines Branch Server services. This can include DNS, DHCP, FTFP, NTP, and RSYNC. Required attributes include the following: ◆ cn ◆ ipHostNumber ◆ scDnsName ◆ scServiceName ◆ scServiceStartScript ◆ scServiceStatus **NOTE:** High availability services (scHAServices) are not supported for Automatic Branch Server Installation. For information on adding this object class to the LDAP directory, see Section 6.3.3, "Adding an scServerContainer and scBranchServer Object," on page 69.

For more information about using posAdmin to create LDAP objects, see Chapter 6, "Using posAdmin to Manage the LDAP Directory," on page 65.

10.3.3 Modifying the Branch Server Configuration Template (template.xml)

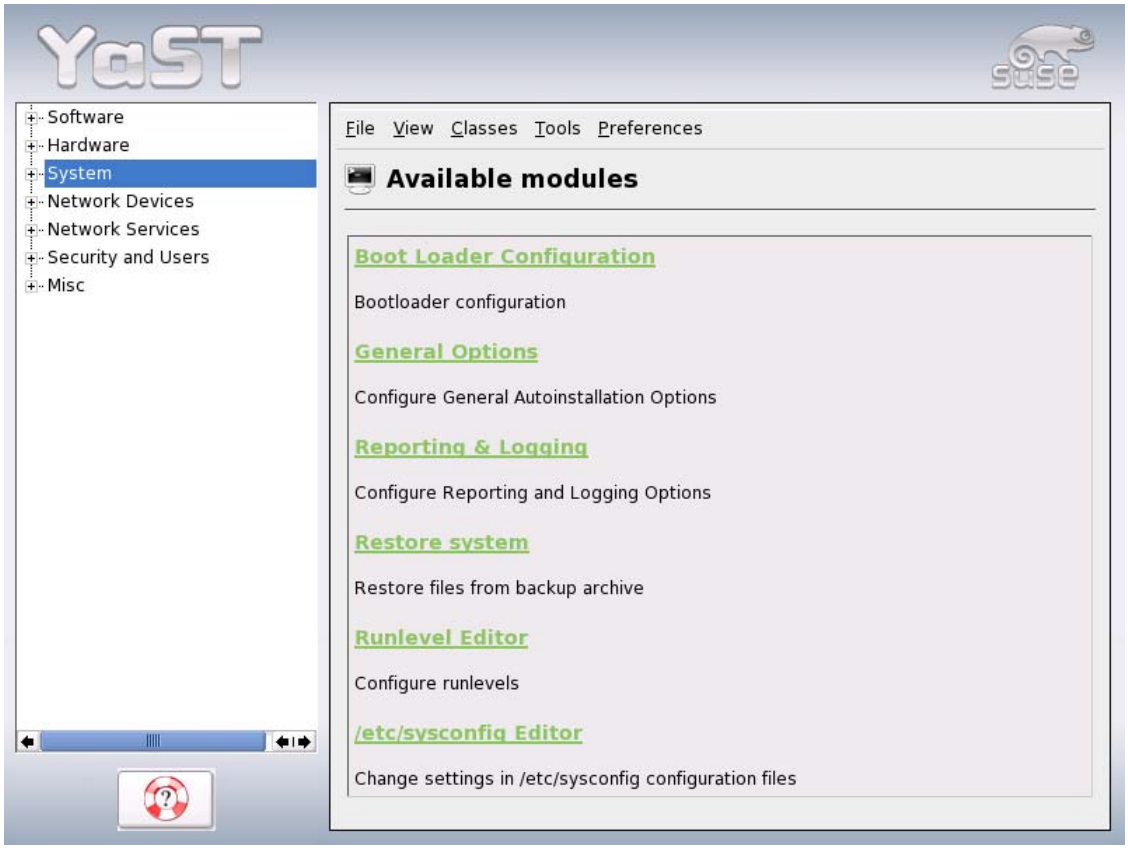
The default Branch Server configuration is defined in the XML template file, `/opt/SLES/POS/xml/template.xml`. The DTD (document type definition) which defines the template document's structure is found in `/usr/share/YaST2/include/autoinstall/profile.dtd`.

NOTE: As with the Image Specification and Distribution Source Documents, this template can be defined in an XML editor or in a standard text editor. However, it is strongly recommended that you modify the document only with the AutoYaST GUI system. The procedure to modify the Automatic Branch Server Installation template in this section is documented using the AutoYaST GUI system.

To modify the `template.xml` document using the AutoYaST system:

1. Start YaST with the `yast2 autoyast` command to display the YaST configuration management system.

Figure 10-2 *AutoYaST configuration management system*



2. Use *Preferences* to set the profile repository to the template file directory (`/opt/SLES/POS/xml/`).
3. Click *File* > *Open* and select the template file (`opt/SLES/POS/xml/template.xml`).

   Modify the sections of the template.

You can change time zone, add software packages, and change system parameters.

A menu system similar to the YaST configuration interface allows you to modify specific sections of the template.

4 After the XML template has been modified with AutoYaST, remove the DOCTYPE entry because it cannot be parsed correctly by the XSLT processor that is used to transform the file.

To perform this operation, run the following command to clean the template file:

```
xmllint --dropdtd template-yast2.xml > template.xml
```

10.3.4 Generating the Automatic Branch Server Installation Image

The `posldap2autoinstcd.pl` script is used to create the Automatic Branch Server Installation images. It generates an ISO file (`autoinst.iso`) and an XML document (`autoinst.xml`). This utility is located in the `/usr/sbin` directory.

The basic command line required to generate the Automatic Branch Server Installation images is:

```
posldap2autoinstcd.pl [options]
```

Table 10-3 summarizes the posldap2autoinstcd command options.

Table 10-3 *posldap2autoinstcd command options*

Option	Description
`--DN` *branchserverdn*	Defines the Branch Server distinguished name (DN). posldap2autoinstcd.pl uses the DN of the branch server to create the following: ◆ An ISO image, autoinst.iso, that fits the description in the template file. It contains all necessary software. ◆ An XML template file, autoinst.xml, that instructs AutoYaST to install the system, set up network interfaces, and configure the branch server system so the server is ready to use. This parameter is required.
`[--user` *ldapuser*`]`	Defines the user account the Branch Server uses to connect to the LDAP directory on the Administration Server. This parameter is optional. If it is not defined, the Branch Server uses the admin account and password created by the posInitLdap.sh or posInitEdir.sh script during the initial configuration of the Administration Server.
`[--pasword` *ldap_password*`]`	Defines the password the Branch Server uses to connect to the LDAP directory on the Administration Server. This parameter is optional. If it is not defined, the Branch Server uses the admin account and password created by the posInitLdap.sh or posInitEdir.sh script during the initial configuration of the Administration Server.

Option	Description
[--SLES *distribution_directory*]	Defines the path to the distribution directory where the SLES RPMs required to build the Automatic Branch Server Installation image are located. This parameter is optional. If it is not defined, posldap2autoinstcd.pl uses the default distribution directory, /opt/SLES/POS/dist/.
[--SP *directory*]	Specifies the directory where an SLRS or SLES service pack is available.This option is used to integrate a service pack into the boot or installation system. This parameter is optional. If it is not defined, posldap2autoinstcd.pl uses only the SLES RPM packages specified in the AdminServer.conf file or the Distribution.xml document. For more information on these files, see Section 8.2.2, "AdminServer.conf," on page 107 or Section 9.2.3, "Distribution Source Document (Distribution.xml)," on page 140.
[--output *directory*]	Defines the directory where the Automatic Branch Server Installation images-autoinst.iso and autoinst.xml-are created. This parameter is optional. If it is not defined, the images are created in the directory where posldap2autoinstcd.pl is executed.
[--tmp *directory*]	Defines the temp directory posldap2autoinstcd.pl uses directory for temporary files. The default is /tmp/.
[--xml *document*]	Defines the name of the Automatic Branch Server Installation XML document produced in the output. This parameter is optional. If it is not defined, the XML document is named autoinst.xml.
[--template *template*]	Defines the directory where the template file used to create the Automatic Branch Server Installation image is located. This parameter is optional. If it is not defined, posldap2autoinstcd.pl uses the default template, /opt/SLES/POS/xml/template.xml.

The following sample command creates an Automatic Branch Server Installation image:

```
posldap2autoinstcd.pl --user cn=admin,0=mycorp,c=us --password secret
--DN cn=bs,cn=server,cn=branch,ou=boston,o=mycorp,c=us
```

This image has the following parameters:

- The branch server uses the user account, cn=admin,o=mycorp, c=us, to log in to the LDAP directory. The password for this account is "secret."
- The Branch Server is associated with scBranchServer object, bs.server.branch.boston.mycorp.us.

10.3.5 Creating the Boot Media

After you generate the Automatic Branch Server Installation images, you must create the CD that will be used to boot and configure the Branch Server.

The build the boot CD, you must complete the following steps:

NOTE: In this example, cdrecord is used to create a CD on a CD recorder. cdrecord is a Linux command line program that is used to record data or audio on a DVD/CD recorder.

---

1. Use the following command to find your CD recorder device:

   ```
   cdrecord -scanbus
   ```

   Linux returns the following information:

   ```
   cdrecord dev=2,0,0
    2,0,0   200) 'PIONEER ' 'DVD-RW  DVR-106D' '1.07' Removable CD-ROM
   ```

2. Record the Automatic Branch Server Installation ISO image (`autoinst.iso`) to the CD:

   ```
   cdrecord dev=2,0,0 autoinst.iso
   ```

3. Create a file named “info” to control the AutoYaST process. It must contain the following lines:

   ```
   install=cd:///
   autoyast=floppy:///autoinst.xml
   autoyast2=floppy:///autoinst.xml
   ```

4. Create a file named “posInitBranchserver.auto.cfg” to control the automatic setup of the branch server software. It must contain the following lines:

   ```
   COMPANYNAME="your_company_name"
   COUNTRY="country"
   ADMINSERVER="IP_address"
   POSADMINDN="dn"
   PASSWORD="username_password"
   ```

   If the password should not be set up automatically for security reasons, it can be omitted. In this case, `posInitBranchserver.sh` requests the password when the Branch Server starts. For more information, see “Setting Up a Branch Server” in the *Novell Linux Point of Service 9 Installation Guide*.

5. Copy the following files to an MS-DOS formatted floppy disk:

   - `info`
   - `posInitBranchserver.auto.cfg`
   - `autoinst.xml`

6. At the Branch Server site, boot the Branch Server by inserting the floppy disk and CD, and then booting the server. Make sure the Branch Server is set to boot from CD.

   The automatic installation system starts. It requests confirmation at the start of the installation, but otherwise runs without interaction.

7. After the system is installed, log in as the root user to start the automatic configuration of the branch server software.

   The `posInitBranchserver.sh` script automatically starts, sets the parameters as specified in the `posInitBranchserver.auto.cfg` file, and requests any missing parameters.

8. Run `possyncimages.pl` to download the client images from the Administration Server.

Remotely Managing Point of Service Terminals with admind and adminc

11

In a Novell® Linux Point of Service system, the admind and adminc utilities allow you to perform tasks like shutdown, configuration reload or application restart on multiple Point of Service terminals from a single location. This section reviews the admind and adminc utilities.

11.1 admind

admind is a small daemon that allows simple commands to be executed on Point of Service terminals from a remote location. Using it with adminc, an administrator can perform tasks like shutdown, configuration reload, or application restart on multiple Point of Service terminals from a single location. admind is typically started by the inetd super-server, but can be run as a regular service.

IMPORTANT: admind does not provide strong authentication. Its level of security is adequate only for systems that boot from the network, thus relying on the integrity of the network infrastructure (DHCP and DNS in particular). Authentication is provided through verification of the hostname and user against a list in the configuration file.

admind writes its diagnostics to the LOG-DAEMON facility at syslog(3).

11.1.1 Command Line Options

admind has the following command syntax:

```
admind [-vIP] [configfile] [options]
```

Table 11-1 summarizes the available admind command line options.

Table 11-1 *admind command line options*

Option	Description
`-I` (uppercase i)	Does not require admind to look up identities to authenticate the calling user. This option is not recommended because it poses a security risk to your system.

Option	Description
-P	Does not require admind to verify the hostname. This option is not recommended because it poses a security risk to your system.
-v	Provides verbose output to syslog.

11.1.2 admind.conf

Standard configuration information for admind is located in /etc/opt/SLES/POS/admind.conf. The file format typically appears as follows:

```
S=hostname1
S=hostname2
U=username1
U=username1
X:0=init 0
X:6=init 6
X:r=/etc/init.d/rc/POSApplication restart
(...)
```

Option	Description
-S	Defines a valid server. The names of the connecting servers are compared against this list. Short names can be used and are expanded for the local domain.
-U	Defines a valid username on the connecting machine.
-X	Defines the fixed commands. Each command has a single letter or digit key (X:[0-9a-zA-Z]).

Executed commands are expected to terminate and deliver a return value. Long-running commands or commands that do not terminate must be wrapped in a script that executes the command in the background.

11.2 adminc

adminc distributes commands to Point of Service terminals running admind. It sends a command string to list of IP addresses. adminc attempts to connect to clients in parallel up to a specified maximum number.

adminc can also be used to start (wake) a series of terminals designated by MAC address.

11.2.1 Command Line Options

adminc has the following command syntax:

```
adminc [--port] portno
       [--parallel] maxparallel
       [--commands] keys IP [IP*]
adminc [--wake] MAC [MAC*]
```

summarizes the available options for adminc.

Table 11-2 *adminc command line options*

Option	Description
`--port`	The port number that admind listens on. The default is 8888.
`--parallel`	The maximum number of parallel sessions to start. The default is 8.
`--commands`	The command keys to be sent to clients. The command keys are specified in the client's admind.conf file.
--wake MAC *MAC_addresses*	The wake command starts the designated clients. Clients are designated by their MAC addresses.

11.2.2 adminc Examples

```
adminc --command 0 192.168.99.11 192.168.99.12 192.168.99.13
Node: 192.168.99.11   Exit Code: 0
Node: 192.168.99.12   Exit Code: 65280
Node: 192.168.99.13   Exit Code: 0
```

11.3 posGetIP

posGetIP is a helper script that is used in conjunction with adminc. It finds all addresses for Point of Service terminals that are managed by the local Branch Server. This tool must be run on the Branch Server. Output is the list of addresses, one line each.

Both IP and MAC addresses can be listed. Default is to list the IP addresses. It finds its server base by looking at the IP addresses that are configured on the local machines. /etc/opt/SLES/POS/branchserver.conf is used to find the LDAP connection information.

11.3.1 Command Line Options

posGetIP has the following command syntax:

```
posGetIP [--ip|noip] [--mac]
```

Table 11-3 summarizes the available posGetIP command options.

Table 11-3 *posGetIP command options*

Option	Description
`--ip`	Prints the IP addresses of all Point of Service terminals that are managed by the local Branch Server. This option is enabled by default.
`--noip`	Provides a screen dump of the Point of Service terminals that are managed by the local Branch Server. This option does not print the IP addresses of the Point of Service terminals managed by the current Branch Server.
`--mac`	Prints the MAC address of all Point of Service terminals that are managed by the local Branch Sever.

11.3.2 posGetIP Examples

```
adminc --command 6 'posGetIP'
adminc --wake 'posGetip --mac --noip'
```

11.4 Installing admind on a Point of Service Terminal

The following sections outline how to add admind to scr and xscr client images.

11.4.1 Adding admind to scr Images

1. Clone the Image Description Tree you want to use to build the image.

   For information on this procedure, see Section 8.4.1, "Cloning the Image Description Tree," on page 109.

2. To start the xinetd service on the Point of Service terminal, add the following line to the `config.system` file in the Image Description Tree (`/opt/SLES/POS/system/image_name-version/config.system`):

   ```
   sbin/insserv /etc/init.d/xinetd
   ```

3. Create the admind.conf file in the `/opt/SLES/POS/system/image_name-version/files-user/` directory.

   ```
   mkdir -p files-user/etc/opt/SLES/POS
   vi files-user/etc/opt/SLES/POS/admind.conf
   ```

4. Set the configuration parameters in the `admind.conf` file.

   4a Set the branch.local parameter to the fully qualified hostname of the Administration or Branch Server that you would like to run adminc on. This allows the terminals to trust the designated box. If you are running adminc from multiple stations, they must be included in this list. For example:

   ```
   S=branch.local
   S=branch2.local
   S=localhost
   ```

   4b Add all users with rights to execute commands on Point of Service terminals. For example:

   ```
   U=root
   U=lreiss
   ```

   4c Add any additional commands you want to execute on the POS terminals. For example:

   ```
   X:0=/sbin/init 0
   X:3=/sbin/init 3
   X:5=/sbin/init 5
   X:6=/sbin/init 6
   X:p=/sbin/poweroff
   X:r=/sbin/reboot
   ```

5. Build the image with the `--extend` option to include the `setup.admind` file.

NOTE: The `setup.admind` file is located in the `/opt/SLES/POS/system/templates/addons/` directory. It references the RPMs required to add the admind utility to a standard client image.

The basic syntax is as follows (type the command all on one line):

```
scr --build --prepare --extend /opt/SLES/POS/system/templates/
addons/setup.admind --image image_name-version --destdir
destination_directory
```

For more information on this procedure, see Section 8.4.2, "Adding Software Packages or Add-on Options to an Image," on page 110.

6 Distribute the image to your Point of Service terminals.

For information on this procedure, see Section 8.5, "Distributing Images," on page 117.

11.4.2 Adding admind to xscr Images

1 Clone the Image Description Tree you want to use to build the image.

For information on this procedure, see Section 9.4.1, "Cloning the Image Description Tree," on page 144.

2 Add `admind.xml` to the IncludeSpecificationsList in the Image Specification Document (`/opt/SLES/POS/system/image_name-version/ImageSpecification.xml`):

NOTE: The `admind.xml` Image Specification Document is located in the `/opt/SLES/POS/system/templates/addons/` directory. It references the RPMs required to add the admind utility to a client image.

The basic syntax is as follows:

```
<IncludeSpecificationList>
   <IncludeSpecification URI="/opt/SLES/POS/system/templates
    /addons/admind.xml"/>
</IncludeSpecificationList>
```

For information on this procedure, see "Adding Features to Client Images" on page 146.

3 To start the xinetd service on the Point of Service terminal, add the following line to the `config.system` file in the Image Description Tree (`/opt/SLES/POS/system/image_name-version/config.system`):

```
sbin/insserv /etc/init.d/xinetd
```

4 Create the `admind.conf` file in the `/opt/SLES/POS/system/image_name-version/files-user/` directory.

```
mkdir -p files-user/etc/opt/SLES/POS
vi files-user/etc/opt/SLES/POS/admind.conf
```

5 Set the configuration parameters in the `admind.conf` file.

5a Set the branch.local parameter to the fully qualified hostname of the Administration or Branch Server that you would like to run adminc on. This allows the terminals to trust the designated box. If you are running adminc from multiple stations, they must be included in this list. For example:

```
S=branch.local
S=branch2.local
S=localhost
```

5b Add all users with rights to execute commands on Point of Service terminals. For example:

```
U=root
U=lreiss
```

5c Add any additional commands you want to execute on the POS terminals. For example:

```
X:0=/sbin/init 0
X:3=/sbin/init 3
X:5=/sbin/init 5
X:6=/sbin/init 6
X:p=/sbin/poweroff
X:r=/sbin/reboot
```

6 Build the image.

The basic syntax is as follows:

`xscr --prepare --image` *`image_name-version`* `--build --destdir` *`destination_directory`*

For information on this procedure, see Section 9.4.4, “Building the Image,” on page 162.

7 Distribute the image to your Point of Service terminals.

For information on this procedure, see Section 8.5, “Distributing Images,” on page 117.

11.5 Installing the admind Client on Administration and Branch Servers

To install admind on an Administration or Branch Server:

1 Install the admind-client RPM on the Administration or Branch Server. For example:

`rpm --install /opt/SLES/POS/dist/NLPOS9/FCS/CD4/suse/i586/admind-client--`*`version`*`.rpm`

NOTE: It may also be necessary to install the tcpd, inetd, and pidentd RPMs.

---

2 Start identd as follows:

```
chkconfig identd on
/etc/init.d/identd start
```

3 Add the identd start command to the server startup script.

Backing Up System Information and Providing Access Control

12

This section provides information about the following tasks in a Novell® Linux Point of Service system:

12.1 Backup and Restore

All system information (system structure, the configuration and deployment method for each Branch Server and Point of Service terminal, image information, and so forth) is stored in an LDAP directory on the Administration Server. This information must be backed up regularly to protect against data loss in case of storage failure and administration errors.

It is recommended that, at a minimum, you do an online logical backup to a local file before any complex reconfiguration of the system.

The following sections discuss methods you can use to backup and restore your Novell Linux Point of Service LDAP directory.

12.1.1 Offline Physical Backup

An offline backup must be executed on the Administration Server and does not put any load on the LDAP server. The drawback is that the LDAP server is not available during the time of the backup.

To perform a physical file backup of the LDAP directory:

1. Stop the LDAP server using the `/usr/sbin/rcldap stop` command.
2. Copy all the files in the /var/lib/ldap/ directory to an archive directory.
3. After the copy completes, start the LDAP server using the `/usr/sbin/rcldap start` command.

12.1.2 Offline Logical Backup

To perform a logical backup of the LDAP directory (database dump):

1. Stop the LDAP server using the `/usr/sbin/rcldap stop` command.
2. Run the `slapcat >ldap.\$(date +'\%Y\%m\%d-\%T')` command.

This generates an LDIF file named `ldap.datetime` where *datetime* is the current date and time. The output file can be archived, backed up on offline media, and restored with the slapadd command. The LDIF file is a structured ASCII file that can be viewed, for example, with the `less` command.

3 After the backup completes, start the LDAP server by using the `/usr/sbin/rcldap start` command.

12.1.3 Online Backup

An online backup uses the LDAP server to extract all data. This has the advantage that the server is available at all times and the backup can be taken from a remote machine that has an LDAP client.

Run the following command:

```
ldapsearch -h LDAPServer -x -b baseDN > ldap.\$(date +'\%Y\%m\%d-\%T')
```

where

LDAPServer is the LDAP server name or IP address.

baseDN is the base DN (distinguished name) of the LDAP structure (for example, o=mycorp,c=us).

This creates an LDIF file like the slapcat command used for offline backup.

This file must be added to the LDAP server with the ldapadd command. Do not use slapadd with this file.

If access controls are implemented on the LDAP server, an authenticated LDAP bind must be used. In this case, the previous command should be extended with the following arguments:

```
ldapsearch -x -D adminDN -w adminPassword
```

where

adminDN is the DN of the administrator user (for example, cn=admin,o=mycorp,c=us).

adminPassword is this user's password (for example, secret).

12.1.4 Restore

To restore an offline backup:

1 Stop the LDAP server using the `/usr/sbin/rcldap stop` command.

2 If you did a physical file backup, restore the files in `/var/lib/ldap`.

or

If you did a logical backup, run the slapadd command to restore the logical database dump:

```
slapadd -l backupfile
```

where *backupfile* is the file created by slapcat.

3 Start the LDAP server using the `/usr/sbin/rcldap start` command.

To restore an online backup, the LDAP server must be running. The LDAP server is able to run with an empty database. If the database has been corrupted, the database files in `/var/lib/ldap/` must be removed before restoring the online backup.

1 To restore a backup file taken with ldapsearch, run the following command:

```
ldapadd -D adminDN -x -w adminPassword -h LDAPServer -x -f
backupfile
```

12.2 Access Control

Access to the LDAP directory should be restricted to comply with your organization's security guidelines and policies.

IMPORTANT: Consult your company security policy to learn about security requirements for the LDAP server, especially local administrator rights and the security ratings for the administration infrastructure.

To restrict access to the directory, access control lists (ACLs) can be implemented in the LDAP server configuration file on the Administration Server. The configuration file is `/etc/openldap/slapd.conf`. For more information, see Section A.3.4, "posInitLdap.sh," on page 205 or man pages slapd.conf(5)and slapd.access(5) for details.

12.2.1 Access Control Example

To restrict access to a specific location, use the following ACLs:

NOTE: The examples use the standard schema of cn=*location*,ou=*orgUnit*,o=*mycorp*,c=*de*.

```
access to dn.base="" by * read
access to * attrs=userPassword
     by anonymous auth
      by self write
access to dn.regex="^.*(cn=.*,ou=.*,o=mycorp,c=us)$"
    by dn.regex="^.*,$1$" write
    by anonymous auth
    by users read
access to *
    by anonymous auth
    by users read
     by self write
```

For each location, create a location user. For example,

```
posAdmin.pl --user cn=admin,o=mycorp,c=us --password secret
--base cn=east,ou=boston,o=mycorp,c=us --add --scPOSUser
--cn EastBostonUser --userPassword "secretPassword"
```

Now the `--user` option can be set to the following in all posAdmin commands concerning the cn=east,ou=boston, o=mycorp, c=us location:

```
--user cn=EastBostonUser,cn=east,ou=boston,o=mycorp,c=us
```

The default LDAP user can now be replaced by this user, especially for the posInitBranchserver command.

```
...
Please enter the DN of the LDAP user for administration tasks [default:
cn=admin,o=mycorp,c=us]
cn=EastBostonUser,cn=east,ou=boston,o=mycorp,c=us
```

Troubleshooting

13

This section describes the analysis and correction of some specific error situations in a Novell® Linux Point of Service system.

13.1 Server Infrastructure

The server setup and operating procedures for Novell Linux Point of Service servers are easy in most circumstances. However, the distributed nature of the Novell Linux Point of Service system might provide some challenges. The following section describes frequently encountered difficulties with name resolution.

13.1.1 Name Resolution

Care must be taken to ensure that the system can resolve its own name to its IP address on the branch network, especially when configuring the Branch Servers with `posInitBranchserver.sh`.

If the system has only one network interface, or if the eth0 interface is the branch network interface, the correct resolution is done through the `/etc/hosts` file, where YaST adds the correct entries. Otherwise, add the corresponding line to `/etc/hosts` manually or make sure that DNS is able to resolve the hostname.

Symptoms

If the DHCP server configuration file `/etc/dhcpd.conf` is not created properly, `poscheckip.pl` returns the following error code:

```
# poscheckip.pl
# echo $?
1
```

If the `dhcpd.conf` file is created properly, `poscheckip.pl` returns the correct hostname, address, netmask and domain as follows:

```
# poscheckip.pl
bs      192.168.150.1   255.255.255.0   Lab.HQ.mycorp.us
# echo $?
0
```

Hints

- Make sure that `/etc/named.conf` lists the right parent. Configure the DNS servers as forwarders.
- Add the hostname to `/etc/hosts`.
- When using DHCP to configure the external (WAN) network interface of the Branch Server, set the DHCP client on the Branch Server to modify `named.conf` instead of `resolv.conf` in `/etc/sysconfig/network/config`. The variables are

MODIFY_RESOLV_CONF_DYNAMICALLY and *MODIFY_NAMED_CONF_DYNAMICALLY*. The template file is prepared for this.

13.2 Operation

The following sections describe frequently encountered difficulties with system operation.

13.2.1 Image Distribution

The `possyncimages.pl` tool distributes the boot and client images from the Administration Server to the Branch Server. It uses the RSYNC service to let the Branch Servers download only the files that need to be updated.

Enough space should be configured to keep at least two generations of image files. This redundancy ensures that there is a valid image available at all times.

NOTE: The SUSE® partitioning recommendation is described in "Partitioning Screen" in the *Novell Linux Point of Service 9 Installation Guide*.

RSYNC updates existing files, creates new files, and even deletes files that do not exist in the original download directory on the Administration Server.

Symptoms

The error message "`rsync: error writing 4 unbuffered bytes -exiting: Broken Pipe`" indicates that the Branch Server does not have enough disk space left to download all the images. Adequate space is required for both the staging area in `/opt/SLES/POS/rsync` and the service area in `/tftpboot`.

Hints

- Make sure that `posldap2crconfig.pl --dumpall` is executed after new images have been distributed, especially after old images have been deleted.
- Make sure that there is enough space for new images even before old images have been deleted, or delete old images before uploading new ones.

13.2.2 Point of Service Terminal Configuration

The process of registering new Point of Service terminals and updating the configuration information usually works without administrator intervention; however, it is a complex process. To facilitate this process, you must ensure the Administration Server has a valid image configuration at all times. In LDAP, the image versions must be entered and made active (see Section 6.5, "Managing Image Objects," on page 79 for details), and the image files must be made available with the right filename (*image_name-version*) and with the right permissions (world-readable).

Further information about the dependencies between LDAP entries, the Branch Server tftpboot directory, and the client image names is shown in Section 1.2, “Dependencies Between LDAP, Branch Server, and Point of Service Terminal,” on page 15.

Symptoms

The error message “`No Imageversion is available`” from `posleases2ldap.pl` or `posldap2crconfig.pl` means that no valid image file for the active version exists. Make sure that the image has been transferred to the Branch Server and that the version in LDAP has an active flag attached.

Hints

- Keep at least two generations of image files available and active in LDAP at all times. The Point of Service downloads the latest client image version available on the Branch Server.
- Ensure that the container with the scHardware object and cn=standards within LDAP always points to an existing scPosImage object. By default, this entry is set to use the Java client image during Administration Server configuration.

13.2.3 Loading CDBoot Images

If there are multiple CD drives in the Point of Service terminal, there is no way to designate which CD drive to use; the system chooses the first one it finds.

Symptoms

If the Point of Service terminal does not find the drive with the boot CD, it returns BIOS errors.

Solution

To correct the problem, insert the CD in the bootable CD drive.

Point of Service Scripts

A

In a Novell® Linux Point of Service system, a number of scripts are provided to initialize and maintain Administration and Branch Servers. This section describes these scripts and their usage.

A.1 Overview

All the programs required to manage the system and to generate configuration files are implemented in Perl and as shell scripts. All the filenames contain the prefix “pos,” so a quick overview of the available programs can be displayed using tab completion.

It is recommended that you use the `/opt/SLES/POS/bin` directory as the storage location for Novell Linux Point of Service scripts. All the scripts can be controlled transparently using the posAdmin metascript, as long as they are not run by cron. The posAdmin script is designed to operate in the same way on the Administration Server as on the Branch Servers.

The basic mechanism for all actions (image transfer to a Branch Server, data readout from the directory) is a pull mechanism from the Branch Servers that is run directly on the Branch Servers. One important element is central logging of all actions with success or failure flags on the Administration Server. For all actions, the rule must be transaction security or atomic execution to avoid, for example, inconsistent configuration files.

A.2 Core Script Process

When Point of Service terminals are being set up in a branch or subsidiary, the posleases2ldap script must be started as a daemon on the Branch Server for the respective branch. All other scripts are controlled by this script.

The interplay of scripts on the Branch Server occurs as follows:

1. posleases2ldap is started directly on the Branch Server. If the scDynamicIp attribute is not set to TRUE in the respective scLocation, the script immediately terminates.
2. posleases2ldap is running as a daemon process and monitors the `/var/lib/dhcpd/dhcpd-leases` file for changes. The script detects in which scLocation (branch) it is running, using the IP address of the server.
3. If posleases2ldap finds MAC addresses in the leases that are not yet entered in the directory, it generates new entries for the scWorkstation object class in the DN for the respective scLocation. The first items filled out are the required attributes macAddress, ipHostNumber, and the cn for the entry. The terminal's IP address and name are automatically generated, and the MAC address is taken from the leases file. These entries are like an outline.
4. A search is made through the upload directory on the TFTP server for files of the pattern `hwtype.MAC_Address` that are being uploaded by Point of Service terminals registered from the DiskNetboot system. The Point of Service hardware type is specified in these files. For more information, see Section 3.6, "Booting the Point of Service Terminal," on page 35. If any files of this type are found, the following process runs:
   a. Using the MAC address, the respective scWorkstation entry is looked up in the LDAP directory. With the content of the `hwtype.MAC_Address` file, the corresponding scRefPc (the reference hardware type in the global container) is searched. In the scRefPc object (named after the hardware type), the image type for this hardware type is specified as a reference to a scPosImage object in the attribute scPosImageDn, which points to the reference image in the global container. The information about the reference hardware and image are then added to the scWorkstation object as distinguished names (DN) and the attributes are named scRefPcDn and scPosImageDn.
   b. All information is collected to generate the `/tftpboot/CR/config.MAC_Address` configuration file. It is possible to specify hardware type or image type dependent configuration files, such as XF86config, which would be hardware type dependent. These files are generated in the `/tftpboot/CR/MAC_Address` directory. For this purpose, an object of the class scConfigFileTemplate can be added to the respective scRefPc or scPosImage object in the global container.

      At this point, the scConfigFileData attribute of the scConfigFileTemplate object contains the required file. Hardware or image dependent configuration files are always looked up by the hardware order image.

      All newly generated files are initially named with the prefix `TMP_`.
   c. The configuration files are renamed from `TMP_*` to their final names.The `/tftpboot/upload/hwtype.MAC_Address` file is deleted. The registration of a newly detected Point of Service terminal is complete.
5. posleases2ldap starts posldap2dns. The zone files for the DNS server are regenerated from the directory data as a temporary file and renamed. The DNS service is restarted if there are any changes.
6. posleases2ldap starts posldap2dhcp. The `dhcpd.conf` file is regenerated from the directory data as a temporary file and renamed. The DHCP service is restarted if there are any changes.
7. posleases2ldap runs in a loop starting at point 2 until it is terminated or the scDynamicIp attribute in the scLocation object for the branch is set to FALSE.
8. posleases2ldap starts the ImageNotify daemon. It monitors `/tftpboot/upload` for boot version, MAC address files, and transfers image notify data to LDAP.

A.3 Script Quick Reference

The remainder of this section provides a brief explanation of each Novell Linux Point of Service script, its function, and usage.

A.3.1 poscheckip.pl

`poscheckip.pl` is a helper script that looks up a server’s IP address in LDAP and outputs the netmask and domain name related to that entry.

Function

poscheckip is used from within `posInitBranchserver.sh` to determine the netmask and domain name related to the hostname of the Branch Server. The information is then used to configure the resolver (`/etc/resolv.conf`).

Usage

```
poscheckip.pl
```

Files

`/etc/opt/SLES/POS/branchserver.conf`

A.3.2 posInitBranchserver.sh

The purpose of `posInitBranchserver.sh` is to generate the central configuration file for all other Novell Linux Point of Service scripts used on a Branch Server, to generate header files needed for automated configuration of DNS and DHCP, to generate configuration files for the DNS and DHCP services, to add a multicast route for TFTP, to activate the DNS, DHCP, and TFTP services at boot time, and to start the services. Information from LDAP is used where applicable.

Function

When running this script, you are prompted to enter the company name, country abbreviation, IP address, and the LDAP administrator password of the Administration Server. The `/etc/opt/SLES/POS/branchserver.conf` configuration file is generated by filling in the LDAP base,

LDAP administrator password, and the IP address of the Administration Server. The `/etc/opt/SLES/POS/template/branchserver.conf.template` file is used as template.

The `posInitBranchserver.sh` script uses `poscheckip.pl` to find its own IP address in LDAP. It only works correctly if the Branch Server data in LDAP was created properly in advance using the posAdmin tool after the installation of the Administration Server. For further information, refer to Chapter 6, "Using posAdmin to Manage the LDAP Directory," on page 65.

The `poscheckip.pl` script also yields the domain name for this branch, which is used to generate proper configuration header files for the DHCP and DNS services, which in turn are needed for `posldap2dns.pl` and `posldap2dhcp.pl`.

The zone file header for `posldap2dns.pl` is generated from `/etc/opt/SLES/POS/template/dns-zonefile.header.template` and written to `/var/named/ldap_generated/dns-zonefile.header`.

The resolver configuration `/etc/resolv.conf` is written, then `posldap2dns.pl` and `posldap2dhcp.pl` are run and the DNS and DHCP services are started.

Finally, a multicast route is set up and the TFTP service is started. The configuration of the multicast route is also stored in `/etc/sysconfig/network/routes` so it is activated at boot time.

Usage

Run `posInitBranchserver.sh` on a Branch Server.

Files

```
/etc/opt/SLES/POS/named/named.conf
/etc/opt/SLES/POS/template/dhcpd.conf.header.template
/etc/opt/SLES/POS/dhcpd/dhcpd.conf.header
/etc/opt/SLES/POS/template/dns-zonefile.header.template
/var/named/ldap_generated/dns-zonefile.header
/etc/opt/SLES/POS/template/resolv.conf.template
/etc/resolv.conf
/etc/sysconfig/network/routes
```

A.3.3 posInitEdir.sh

The purpose of `posInitEdir.sh` is to configure the LDAP directory in Novell eDirectory™. You are prompted to enter the tree name, company name, country abbreviation, and the LDAP administration password. Company name and country abbreviation are used to compose the LDAP base DN in the form o=company,c=us.

Function

`posInitEdir.sh` uses `/etc/opt/SLES/POS/template/edir.schema` to create the LDAP directory. The LDAP base DN, and password are replaced with the corresponding user entries. After generating the configuration file, eDirectory is started.

`posInitEdir.sh` uses `posReadPassword.pl` during the password entry to hide the password characters.

Usage

Run `posInitEdir.sh` on an Administration Server.

WARNING: Running this script overwrites any existing eDirectory tree on the server.

Files

`/etc/opt/SLES/POS/template/edir.schema`

A.3.4 posInitLdap.sh

The purpose of `posInitLdap.sh` is to configure the OpenLDAP directory server software and to create the initial data in the LDAP directory. You are prompted to enter the company name, country abbreviation, and the LDAP administration password. You can also enable or disable SSL communication. Company name and country abbreviation are used to compose the LDAP base DN in the form o=company,c=us.

Function

`posInitLdap.sh` uses `/etc/opt/SLES/POS/template/slapd.conf.template` to create the OpenLDAP configuration file, `/etc/openldap/slapd.conf`. The LDAP base DN and password are replaced from the `posInitLdap.sh` script with the corresponding user entries. After generating the configuration file, the OpenLDAP service is started.

`posInitLdap.sh` then uses a template file, `/etc/opt/SLES/POS/template/ldif.pos.template`, to create an LDAP data file, `/etc/opt/SLES/POS/template/ldif.pos`, which it then imports into the LDAP directory. Now the initial LDAP directory structure is available on the Administration Server.

`posInitLdap.sh` uses `posReadPassword.pl` during the password entry to hide the password characters.

Usage

Run `posInitLdap.sh` on an Administration Server.

WARNING: Running this script destroys any existing data in LDAP.

Files

`/etc/openldap/ldap.conf`
`/etc/openldap/slapd.conf`
`/etc/opt/SLES/POS/template/slapd.conf.template`
`/etc/init.d/ldap`
`/etc/opt/SLES/POS/template/ldap.template`
`/etc/opt/SLES/POS/template/ldif.pos.template`

A.3.5 posldap2crconfig.pl

`posldap2crconfig.pl` creates or updates configuration files for Point of Service terminals. Those configuration files are generated by gathering data from LDAP; they contain the information required to boot the Point of Service terminal such as partition information, image, partitioning, hard drive, and so forth.

Function

In normal operation, `posldap2crconfig.pl` does a part of what is done by `posleases2ldap.pl`: it looks for `hwtype.`*`MAC_address`* files uploaded by Point of Service terminals, looks up the terminal's LDAP entry, assigns the hardware type and the default image for this hardware type in terminal's LDAP entry, and finally generates the configuration files in the CR subdirectory under the tftpboot directory. The file uploaded by the Point of Service terminal is then removed from the `/tftpboot/upload` directory.

`posldap2crconfig.pl` can optionally be run with the `--dumpall` parameter. Using this mode, `posldap2crconfig.pl` regenerates the `config.`*`MAC_address`* and hardware configuration files for all Point of Service terminals found in LDAP.

NOTE: When posldap2crconfig generates syslog messages, these messages are displayed in all open shell windows of the Branch Server, if the default setting of the configuration file `/etc/syslog.conf` is used. To avoid this behavior, edit the following line in `/etc/syslog.conf` and change it as shown below:

```
# *.emerg                         *
```

Usage

```
posldap2crconfig.pl [--dumpall]
```

Files

```
/etc/opt/SLES/POS/branchserver.conf
```

A.3.6 posldap2dhcp.pl

`posldap2dhcp.pl` generates the DHCP daemon configuration file from LDAP.

Function

`posldap2dhcp.pl` is called by `posleases2ldap.pl` at regular intervals. First, all scLocation objects are looked up in LDAP. Each of these objects defines a subnet and for each of them a subnet declaration in the `dhcpd.conf` is generated.

The header zone file is taken from the file specified in the configuration file directive LDAP2DHCP_TEMPLATEFILE, which is `/etc/opt/SLES/POS/dhcpd/dhcpd.conf.header` by default. The content of the header file is adapted to the installation by `posInitBranchserver.sh` (see Section A.3.2, "posInitBranchserver.sh," on page 203).

The value of the scDhcpRange attribute in a scLocation object is translated into a range statement in the subnet declaration.

In addition, the options for tftpboot are written into each subnet declaration. For each scCashRegister, a fixed address declaration is generated.

Function

The new `dhcpd.conf` file is first generated in a temporary directory. If it differs from the working version, dhcpc is run with the temporary file in check mode. If it passes the check, it is copied over the working file and the command to restart the DHCP daemon is returned to be executed by `posleases2ldap.pl`.

Usage

`posldap2dhcp.pl` is called by `posleases2ldap.pl`.

Files

```
/etc/opt/SLES/POS/branchserver.conf
/etc/dhcpd.conf -> /etc/opt/SLES/POS/dhcpd/dhcpd.conf
/etc/opt/SLES/POS/dhcpd/dhcpd.conf.header
```

A.3.7 posldap2dns.pl

`posldap2dns.pl` generates DNS configuration and zone files from LDAP.

Function

`posldap2dns.pl` is called by `posleases2ldap.pl` at regular intervals. First, all scLocation objects are looked up in LDAP. Each of these objects defines a subnet and for each of them a zone file is created.

The header of each zone file is taken from the file specified in the configuration file directive POS_LDAP2DNS_ZONETEMPLATE, which is `/var/named/ldap_generated/dns-zonefile.header` by default. The content of the zone file header is adapted to the installation by `posInitBranchserver.sh` (see Section A.3.2, “posInitBranchserver.sh,” on page 203).

The value of the scDhcpRange attribute in a scLocation object is translated into a \$GENERATE directive. For each scService or scHAService, an A record is created or, if multiple objects of that kind point to the same IP address, a CNAME record. After that, an A record for each Point of Service terminal is generated.

Finally, the `/var/named/ldap_generated/named.zones` file containing the definitions of all generated zones is created. It is included from within `/etc/named.conf`. If zones were changed, `posldap2dns.pl` returns the appropriate commands to restart the DNS service. The commands are executed by `posleases2ldap.pl`.

Usage

`posldap2dns.pl` is called by `posleases2ldap.pl`.

Files

```
/etc/opt/SLES/POS/branchserver.conf
/var/named/ldap_generated/
```

/var/named/ldap_generated/dns-zonefile.header
/var/named/ldap_generated/named.zones
/etc/named.conf

A.3.8 posleases2ldap.pl

posleases2ldap.pl registers new Point of Service terminals in LDAP and transfers image install notification data to LDAP. It also triggers posldap2crconfig.pl.

Function

See Section A.2, "Core Script Process," on page 201 for a detailed description of posleases2ldap.pl.

Usage

In normal operation, posleases2ldap.pl is run as a daemon. It can be started by using the /etc/init.d/posleases2ldap init script, which is also used to start the daemon at boot time. To enable this, use chkconfig posleases2ldap on.

If posleases2ldap.pl is started manually, it immediately backgrounds itself. To avoid this, use the optional parameter -d. If started in this way, posleases2ldap closes when the shell is closed.

Files

/etc/opt/SLES/POS/branchserver.conf
/tftpboot/upload/hwtype.*MAC_address*

A.3.9 posReadPassword.pl

posReadPassword.pl is a helper script for password entry that does not show the entered password.

Function

posReadPassword.pl is called by posInitLdap.sh, posInitEdir.sh, and posInitBranchserver.sh for password entry purposes.

Usage

From within shell scripts, use a line such as
PASSWORD=`posReadPassword.pl`

Files

None.

A.3.10 possyncimages.pl

The possyncimages.pl script must be run on a Branch Server to download or update the images from the Administration Server. It uses RSYNC and requires that the RSYNC service is properly configured and running on the Administration Server. This script can be run manually, but

depending on your system requirements, you can create a cron job that runs the script every night to keep the images up to date.

Function

`possyncimages.pl` reads the `/etc/opt/SLES/POS/branchserver.conf` configuration file and uses the definitions POS_REMOTE_SYNC_COMMANDS and POS_LOCAL_SYNC_COMMANDS from that file. POS_REMOTE_SYNC_COMMANDS contains a list of RSYNC commands that obtain the data from the Administration Server. These commands are executed first.

On success, the commands in the POS_LOCAL_SYNC_COMMANDS directory are executed to update the final destination of the images.

Usage

Run `possyncimages.pl` on a Branch Server or set up a cron job. A crontab line for nightly run at 1 a.m. might look like this:

```
0 1 * * * /usr/sbin/POSsyncimages.pl
```

Files

`/etc/opt/SLES/POS/branchserver.conf`

Novell Linux Point of Service Files and Directory Structure

B

This section provides a quick reference of Novell® Linux Point of Service directory structure.

B.1 Administration Server Directory Structure

Directory	Description
/etc/openldap/	
slapd.conf	The LDAP server configuration file. To restrict access to the LDAP directory, access control lists (ACLs) can be implemented in the slapd.conf file.
/etc/opt/SLES/POS/	
admind.conf	A small daemon that allows simple commands to be executed on Point of Service terminals from a remote location. Using it with adminc, an administrator can perform tasks like shutdown, configuration reload or application restart on multiple terminals from a single location. For more information, see Section 11.1, “admind,” on page 187.
AdminServer.conf	An ASCII, line-based file that provides the paths to the installation source tree where you have copied the Novell Linux Point of Service CDs. Scr references AdminServer.conf when it builds images. To generate this file, run posCDTool or POSCopyTool. For more information, see Section 7.3.4, “Generating AdminServer.conf or Distribution.xml,” on page 94. For more information, see Section 8.2.2, “AdminServer.conf,” on page 107.
AdminServer.conf.nld	A master template file for AdminServer.conf. This version of the AdminServer.conf file lists the source paths for standard client (NLD-based) images. To generate this file, run POSCDTool or POSCopyTool. For more information, see Section 7.3.4, “Generating AdminServer.conf or Distribution.xml,” on page 94.

Directory	Description
/etc/opt/SLES/POS/	**continued**
AdminServer.conf.sles	A master template file for AdminServer.conf. This version of the AdminServer.conf file lists the source paths for POSBranch (SLES-based) images. posldap2autoinst.pl references this file to generate AutoBranch images. To generate this file, run POSCDTool or POSCopyTool. For more information, see Section 7.3.4, “Generating AdminServer.conf or Distribution.xml,” on page 94.
branchserver.conf	The standard configuration file for Administration and Branch Servers.
ImageSpecification.xsd	The Novell Linux Point of Service XML schema document
atftp/	The atftp directory contains sample configuration files for the TFTP service the Branch Server uses to download images and configuration files to Point of Service terminals.
dhcpd/	The dhcpd directory contains sample configuration files for the DHCP service provided by Branch Servers for Point of Service terminals.
ha/	The ha directory contains sample configuration files for high availability services provided by Branch Servers for Point of Service terminals. For more information on configuring high availability services, see “Setting Up High Availability Branch Servers” in the *Novell Linux Point of Service 9 Installation Guide*.
keys/	The keys directory contains the keys and certificates required to secure LDAP communication between the Administration and Branch Servers. During installation of the Administration Server, Novell Linux Point of Service automatically installs a CA and generates self-signed certificates to secure communication between the Administration and Branch Servers. However, the CA’s public key is distributed to the branch servers only if you enable LDAP SSL during installation. For more information on setting up LDAP SSL, see “Running posInitLdap.sh” in the *Novell Linux Point of Service 9 Installation Guide*.
ca/	The ca directory contains the CA certificate and keys.
ca.crt	The public key for the CA that signed the server certificate. This is copied over to the RSYNC directory only if you enable LDAP SSL during installation of the Administration Server. The CA's public key allows the Branch Servers to trust the Administration Server.
ca.db.certs	A database that tracks the server certificates the CA has signed.
ca.key	The CA’s private key.

Directory	Description
/etc/opt/SLES/POS/keys/	**continued**
certs/	The certs directory contains the Administration Server certificate and keys.
server.crt	The Administration Server certificate public key. This certificate is used to secure LDAP communication between the administration and Branch Server.
server.csr	The Administration Server's Certificate Signing Request (CSR). This form is submitted to the CA. The CA signs the CSR to create the server certificate.
server.key	The server certificate's private key.
named/	The named directory contains a sample configuration file (named.conf) for the DNS service provided by Branch Servers for Point of Service terminals.
rsync/	The rsync directory contains the configuration files for the RSYNC service.
rsyncd.conf	The Administration Server's RSYNC configuration file.
rsyncdbranch.conf	The Branch Server's RSYNC configuration file.
template/	The template directory contains the template files required for the administration and Branch Server services.
adminserver.conf.template	The template file for the adminserver.conf file.
branchserver.conf.template	The template file for the Branch Server configuration file.
dhcpd.conf.header.template	The template file for the DHCP service.
dns-zonefile.header.template	The template file for the DNS service.
edir.schema	The Novell eDirectory™ schema file used by posInitEdir.sh to create the LDAP directory.
openldap.template	A sysconfig template file that posInitLdap.sh uses for LDAP configuration.
ldif.pos	The LDAP file that posInitLdap.sh imports into the Administration Server's openLDAP directory. This file uses the structure of ldap.pos.template, but is populated with the names provided during installation.
ldif.pos.template	The template for the ldif.pos file.

Directory	Description
/etc/opt/SLES/POS/template/	**continued**
pxelinux.cgf.template	The template file for pxe.linux.cfg files. Pxelinux.cfg files are stored on the Branch Server. They indicate which kernel and RAM disk to load for the Point of Service terminal. These files enable Branch Servers to distribute SLRS 8 and Novell Linux Point of Service 9 images. Novell Linux Point of Service automatically creates the pxelinux.cfg files based on the distribution container configurations in the LDAP directory.
resolv.conf.template	The template file for DNS configuration.
slapd.conf.template	The template file posInitLdap.sh uses to create the openLDAP server configuration file, /etc/openldap/slapd.conf.
/opt/SLES/POS	
imagexml.pl	This file is part of the ImageBuilder program. imagexml.pl is launched by the xscr script in the /usr/bin/ directory.
image.pl	This file is part of the ImageBuilder program. image.pl is launched by the scr script in the /usr/bin/ directory.
bin/	The bin directory contains Novell Linux Point of Service scripts.
dist/	The dist directory contains the archived Novell Linux Point of Service CDs. ImageBuilder references these directories to locate the RPMs required to build client images.
NLD9/	The NLD9 directory contains the Novell Linux Desktop 9 RPMs
SLES9/	The SLES9 directory contains the SUSE® LINUX Enterprise Server 9 RPMs.
NLPOS9/	The NLPOS9 directory contains the Novell Linux Point of Service RPMs.
image/	The image directory is the default location for binary image files generated with ImageBuilder.
maintenance/	The maintenance directory is an “override” directory. When scr generates an image, it first looks in this directory to find the RPM packages required to create the image. You can add any RPM to this directory that you want scr to use in lieu of the default RPMs in the distribution directories.
nld	The nld directory contains override RPMs for the NLD distribution. RPMs located in this directory take precedence over RPMs located in the distribution directories.
suse	

Directory	Description
/opt/SLES/POS/ maintenance/nld/suse/	**continued**
i586	The i586 directory contains override RPMs specific to i586 machines. RPMs located in this directory take precedence over RPMs located in the distribution directories.
devs-9-16.11.i586.rpm	The devs package creates files in the /dev directory that are required to access system hardware.
glibc-2.3.3-98.47. i586.rpm	The glibc package is the GNU libc program for i586 machines.
i686	The i686 directory contains override RPMs specific to i686 machines. RPMs located in this directory take precedence over RPMs located in the distribution directories.
glibc-2.3.3-98.47. i686.rpm	The glibc package is the GNU libc program for i686 machines.
sles	The sles directory contains override RPMs for the SLES distribution. RPMs located in this directory take precedence over RPMs located in the distribution directories.
suse	
i586	The i586 directory contains override RPMs specific to i586 machines. RPMs located in this directory take precedence over RPMs located in the distribution directories.
devs-9-16.11.i586.rpm	The devs package creates files in the /dev directory that are required to access system hardware.
glibc-2.3.3-98.47. i586.rpm	The glibc package is the GNU libc program for i586 machines.
i686	The i686 directory contains override RPMs specific to i686 machines. RPMs located in this directory take precedence over RPMs located in the distribution directories.
glibc-2.3.3-98.47. i686.rpm	The glibc package is the GNU libc program for i686 machines.
pac/	The pac directory contains the Novell Linux Point of Service kernel.

Directory	Description
/opt/SLES/POS/	**continued**
rsync/	The rsync directory contains the files and images that are distributed by the Administration Server over RSYNC.
boot/	The boot directory contains active boot images that are distributed by the Administration Server over RSYNC. Ultimately, these images are distributed by the Branch Server to Point of Service terminals over TFTP.
initrd.gz	The initrd.gz file ships with Novell Linux Point of Service as initrd-disknetboot-*version-date*.gz. The initrd.gz image is the first bootstrap image used to PXE boot the Point of Service terminals. **IMPORTANT:** The initrd-disknetboot-*version-date*.gz image must be copied to the opt/SLES/POS/rsync/boot/ directory as initrd.gz before running posSynchImages.pl on the Branch Server. For more information on this process, see "Setting Up the Administration Server" in the *Novell Linux Point of Service 9 Installation Guide*.
linux	The linux file ships with Novell Linux Point of Service as initrd-disknetboot-*version-date*.kernel.*kernel_version*. The Linux image provides the Linux kernel used to PXE boot the Point of Service terminals. **IMPORTANT:** The DiskNetboot-*version-date*.kernel.*version*-SLRS image must be copied to the opt/SLES/POS/rsync/boot/ directory as linux before running posSynchImages.pl on the Branch Server. For more information on this process, see "Setting Up the Administration Server" in the *Novell Linux Point of Service 9 Installation Guide*.
certs/	The certs directory stores the CA's public key.
ca.crt	The public key for the CA that signed the server certificate. This is copied over to the RSYNC directory only if you enable LDAP SSL during installation of the Administration Server. The CA's public key allows the Branch Servers to trust the Administration Server.
config/	The config directory contains hardware configuration files that are distributed by the Administration Server over RSYNC. Ultimately, these configuration files are distributed by the Branch Server to Point of Service terminals over TFTP. **IMPORTANT:** Any configuration files referenced in the scConfigFileSyncTemplate object must be located in /opt/SLES/POS/rsync/config/.

Directory	Description
/opt/SLES/POS	**continued**
image/	The image directory contains active client images that are distributed by the Administration Server over RSYNC. Ultimately, these images are distributed by the Branch Server to Point of Service terminals over TFTP.
system/	The system directory is the staging area for client images. The information used to build client images is stored in this directory and its sub-directories.
image_name-version/	The image directories contain the Image Specification Documents and Image Description Trees ImageBuilder requires to build client images. For more information on these files, see Section 8.2.1, “Image Description Tree,” on page 102 and Section 9.2, “xscr Image Building Components,” on page 126.
config	A configuration file that indicates the image size, type, and base name. The structure of the file corresponds to the format Key: Value.
config.cleanup	An optional configuration script for the image. This script is called at the end of the installation and after all the installation scripts have run. It is designed to clean up the image system. The target programs and files are those needed only while the installation scripts are running.
config.system	An optional configuration script for the image. This script is called at the end of the installation but before the installation scripts have run. It is designed to configure the image system, such as the activation or deactivation of certain services (insserv). The call is not made until after the switch to the image has been made with chroot.
IMAGE	An unformatted file that contains a brief description of the image and its function.
ImageSpecification.xml	The ImageSpecification.xml documents contain XML elements that define the structure, configuration files, and other components required to build client images for Point of Service systems.
setup	A configuration file that indicates which packages make up the image and which RPM options must be used to install them. Each package can also be accompanied by a specific version of the package.
setup.txt	An optional information file for the LDAP system. This file contains information regarding which configuration files are required by the image and whether they are hardware or system--dependent.
setup.user	An optional configuration file that can be present in addition to setup. The file has the same format as the setup file, but a path to the package can be indicated after the package version.

Directory	Description
/opt/SLES/POS/system/image_name-version/	**continued**
VERSION	A file that contains the version number of the Image Description Tree, such as 1.1.2. If you want to change the version number of your Image Description Tree, you must edit the VERSION and the name of the Image Description Tree directory. If you only modify the version included in the directory, the ImageBuilder does not list the correct version number.
files/	A subdirectory that contains special files, directories, and scripts, This function of this directory is to ensure that the RPM is used as the package manager before any packages are installed in the image. This directory cannot contain any libraries or binary files. Any binaries and libraries required before the first RPM call must be extracted from the corresponding packages in advance.
files-user/	A subdirectory that contains special files, directories, and scripts for adapting the image environment after the installation of all the image packages.
package/	A subdirectory in which searches for packages occur. The directory is automatically initialized depending on the entries in the ImageBuilder /etc/opt/SLES/POS/AdminServer.conf configuration file.
script/	A subdirectory that contains Bash scripts that are called after a package is installed, primarily to remove the parts of a package that are not needed for the Point of Service system.
templates/	The templates directory contains the components ImageBuilder requires to build client images.
Distribution.xml	The Distribution Source Document defines the media to be used when generating the image.
dataImage.xml	The data image template file is used by xscr for internal processes. **WARNING:** Do not modify or move this file.
addons/	The addons directory contains the Image Specification Documents for features that can be added to Point of Service terminals. For more information, see Section 4.4, “Client Image Add-On Features,” on page 49 and “Adding Features to Client Images” on page 146.
admind.xml	This Image Specification Document adds admind to client images. For more information, see Section 11.4.2, “Adding admind to xscr Images,” on page 191.
alsa.xml	This Image Specification Document adds the Advanced Linux Sound Library (ALSA) to client images. ALSA provides audio and MIDI functionality for Point of Service terminals.

Directory	Description
/opt/SLES/POS/system/templates/addons/	**continued**
debug.xml	Adds debugging tools to client images for troubleshooting purposes.
evtouch.xml	This Image Specification Document adds the driver for evtouch screens in ncurses mode. **NOTE:** This driver does not support evtouch screens in X11 mode.
gnome.xml	This Image Specification Document adds the GNOME desktop to NLD-based client images. This feature can be added only to the NLD Desktop image.
gnome-sles.xml	This Image Specification Document adds the GNOME desktop to SLES-based images used for POSBranch. This feature can be added only to the SLES Desktop image.
ibmjava.xml	This Image Specification Document adds the IBM Java Runtime Environment (JRE) to NLD-based client images. This feature can be added to the NLD-based Java, Browser, or Desktop images.
ibmjava-sles.xml	This Image Specification Document adds the IBM Java Runtime Environment (JRE) to SLES-based images used for POSBranch. This feature can be added to the SLES-based Java, Browser, or Desktop images.
kde.xml	This Image Specification Document adds the KDE desktop to NLD-based client images. This feature can be added only to the NLD Desktop image.
kde-sles.xml	This Image Specification Document adds the KDE desktop to SLES-based images used for POSBranch. This feature can be added only to the SLES Desktop image.
mozilla.xml	This Image Specification Document adds the Mozilla browser to client images. This feature can be added to the Browser or Desktop images.
samba.xml	This Image Specification Document provides Common Internet File System (CIFS) file access for Windows and Linux clients. **NOTE:** The Samba 3 server is included with Novell Linux Point of Service. This feature can be added to any client image.

Directory	Description
/opt/SLES/POS/system/templates/addons/	**continued**
setup.admind	This file is used to extend standard client images to include admind. For more information, see Section 11.4.1, “Adding admind to scr Images,” on page 190.
vim.xml	This Image Specification Document adds Vim (Vi IMproved) to client images. Vim is an almost compatible version of the UNIX editor vi. Almost every possible command can be performed using only ASCII characters. Many new features have been added such as multilevel undo, command line history, filename completion, block operations, and editing of binary data. Vi is available for the AMIGA, MS-DOS, Windows NT, and various versions of UNIX.
vnc.xml	This Image Specification Document adds the VNC 4 Remote Control client to the image so you can remotely control the Point of Service terminal over any TCP/IP connection. This feature can be added to Browser or Desktop images.
yast2.xml	This Image Specification Document adds the YaST2 console to client images. YaST2 is the system configuration console. It can configure hardware (sound cards, printers, keyboards, mice), network connections (network cards, ISDN cards, modems, DSL connections), network clients and services (NFS, NIS), as well as a general system options (language, partitioning, software, bootloader).
/opt/SLES/POS/system/templates/drivers/	The drivers directory contains the default configuration information for all kernel drivers. New images can include or exclude these drivers in the ImageSpecification.xml file. For more information, see “Adding Drivers” on page 148.
drivers.xml	An XML document specifying general system-level drivers.
net.xml	An XML document specifying network drivers.
scsi.xml	An XML document specifying SCSI drivers.
usb.xml	An XML document specifying USB drivers.
/opt/SLES/POS/system/templates/locale/	
de_DE	The de_DE directory contains the German locale documents.
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.

Directory	Description
/opt/SLES/POS/system/templates/locale/	**continued**
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
mozilla.xml	This Image Specification Document provides the language files required to support the Mozilla browser in Desktop or Browser client images.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
es_ES	The es_ES directory contains the Spanish locale documents.
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
fr_FR	The fr_FR directory contains the French locale documents.
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.

Directory	Description
/opt/SLES/POS/system/templates/locale/	**continued**
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
it_IT	The it_IT directory contains the Italian locale documents.
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
ja_JP	The ja_JP directory contains the Japanese locale documents.
browser.xml	This Image Specification Document provides the language files required to support the NLD-based Browser image.
browser-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Browser image.
desktop.xml	This Image Specification Document provides the language files required to support the NLD-based Desktop image.
desktop-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Desktop image.
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
java.xml	This Image Specification Document provides the language files required to support the NLD-based Java image.
java-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Java image.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.

Directory	Description
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
mozilla.xml	This Image Specification Document provides the language files required to support the Mozilla browser in Desktop or Browser client images.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
ko_KR	The ko_KR directory contains the Korean locale documents.
browser.xml	This Image Specification Document provides the language files required to support the NLD-based Browser image.
browser-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Browser image.
desktop.xml	This Image Specification Document provides the language files required to support the NLD-based Desktop image.
desktop-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Desktop image.
java.xml	This Image Specification Document provides the language files required to support the NLD-based Java image.
java-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Java image.
mozilla.xml	This Image Specification Document provides the language files required to support the Mozilla browser in Desktop or Browser client images.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
pt_PT	The pt_PT directory contains the Portuguese locale documents.
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
zh_CN	The zh_CN directory contains the locale documents for simplified Chinese.

Directory	Description
/opt/SLES/POS/system/templates/locale/	**continued**
browser.xml	This Image Specification Document provides the language files required to support the NLD-based Browser image.
browser-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Browser image.
desktop.xml	This Image Specification Document provides the language files required to support the NLD-based Desktop image.
desktop-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Desktop image.
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
java.xml	This Image Specification Document provides the language files required to support the NLD-based Java image.
java-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Java image.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
zh_TW	The zh_TW directory contains the locale documents for traditional Chinese.
browser.xml	This Image Specification Document provides the language files required to support the NLD-based Browser image.
browser-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Browser image.
desktop.xml	This Image Specification Document provides the language files required to support the NLD-based Desktop image.
desktop-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Desktop image.
gnome.xml	This Image Specification Document provides the language files required to support the GNOME desktop in NLD-based client images.

Directory	Description
/opt/SLES/POS/system/templates/locale/	**continued**
gnome-sles.xml	This Image Specification Document provides the language files required to support the GNOME desktop in SLES-based images used for POSBranch.
java.xml	This Image Specification Document provides the language files required to support the NLD-based Java image.
java-sles.xml	This Image Specification Document provides the language files required to support the SLES-based Java image.
kde.xml	This Image Specification Document provides the language files required to support the KDE desktop in NLD-based client images.
kde-sles.xml	This Image Specification Document provides the language files required to support the KDE desktop in SLES-based images used for POSBranch.
yast2.xml	This Image Specification Document provides the language files required to support the YaST2 console in client images.
/opt/SLES/POS/system/templates/support/	The support directory contains the Image Specification Documents used to create client and boot images. You can customize these template documents to create the final image. For more information, see Section 4.2, “Point of Service Boot Images,” on page 43 and Section 4.3, “Point of Service Client Images,” on page 45. For information on generating an image, see Section 9.4, “Building Images with xscr,” on page 144.
branch.xml	The POSBranch template (branch.xml) provides the following Branch Server components: ◆ All the RPMs required for a functional Branch Server. ◆ The Linux Kernel Crash Dump (LKCD) to provide a system for detecting, saving and examining system crashes. ◆ The RPM database so YAST2-Online can be used to update the image. ◆ Branch Server configuration information obtained from the LDAP directory.

Directory	Description
/opt/SLES/POS/system/templates/support/	**continued**
browser.xml	A child document for the Browser Image Specification Document that includes the NLD RPMs in the Browser client image. When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (--dist nld|sles). If you define the image distribution as NLD, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document. **NOTE:** NLD is the default distribution type. Therefore, if you do not specify the image distribution when cloning the Image Description Tree, xscr defaults to NLD. In general, most Point of Service images are created using the NLD distribution.
browser-base.xml	The base template Image Specification Document for the Browser image. This file specifies the drivers and RPMs required to create the Browser image. It is included as a child document in the ImageSpecification.xml document at the root of the Browser Image Description Tree. It includes all elements of the Minimal and Java images, but is also equipped with Mozilla as a Web browser. The image can be extended to include other Web browsers. The Browser image supports console-based C/C++ applications, Java programs in a Java2 runtime environment, and X11 applications. The required maximum size of the Browser image is 150 MB compressed. This image is intended for diskful systems; however, if you have enough RAM, you can deploy the image in memory. To deploy the default Desktop image on a diskless system, the terminal must have at least 1 GB of RAM.To deploy the image on a diskful system, the terminal must have 150 MB of available hard disk space and 256 MB of RAM.
browser-sles.xml	A child document for the Browser Image Specification Document that includes the SLES RPMs in the Browser client image. When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (--dist nld|sles). If you define the image distribution as SLES, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document. The only Point of Service images that require the SLES distribution are POSBranch images. For more information on POSBranch, see Section 4.5, “POSBranch Images,” on page 51.

Directory	Description
/opt/SLES/POS/system/templates/support/	**continued**
cdboot.xml	CDBoot includes all the files and directories required to boot diskless and preinstalled diskful systems from CD. To boot diskless systems, the image loads RAM disks from a fixed CD image file. Novell Linux Point of Service includes a binary version of the CDBoot image that is used to boot Point of Service terminals from a CD. This image must be combined with a client image and the config.image configuration file to create CD that can be used to boot Point of Service terminals. For information on creating CDBoot images, see Section 10.1, “Building a CDBoot Image,” on page 171.
desktop.xml	A child document for the Desktop Image Specification Document that includes the NLD RPMs in the Desktop client image. When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (--dist nld|sles). If you define the image distribution as NLD, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document. **NOTE:** NLD is the default distribution type. Therefore, if you do not specify the image distribution when cloning the Image Description Tree, xscr defaults to NLD. In general, most Point of Service images are created using the NLD distribution.
desktop-base.xml	The base template Image Specification Document for the Desktop image. This file specifies the drivers and RPMs required to create the Desktop image. It is included as a child document in the ImageSpecification.xml document at the root of the Desktop Image Description Tree. It includes one Web browser (Mozilla) with plug-ins and a full graphical user interface (KDE 3.2 or GNOME 2.6). The Desktop image supports console-based C/C++ applications, Java programs in a Java2 runtime environment, and X11 applications. This image is intended for diskful systems; however, if you have enough RAM, you can deploy the image in memory. To deploy the default Desktop image on diskless systems, the terminal must have at least 1 GB of RAM.

Directory	Description
/opt/SLES/POS/system/templates/support/	**continued**
desktop-sles.xml	A child document for the Desktop Image Specification Document that includes the SLES RPMs in the Desktop client image. When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (--dist nld|sles). If you define the image distribution as SLES, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document. The only Point of Service images that require the SLES distribution are POSBranch images. For more information on POSBranch, see Section 4.5, “POSBranch Images,” on page 51.
disknetboot.xml	DiskNetboot includes all the files and directories (including partitioning and boot loader installation) required to boot diskful and diskless Point of Service systems from the network. Novell Linux Point of Service includes binary versions of the first and second stage boot images used to PXE boot Point of Service terminals. **IMPORTANT:** The boot images must be copied to the /opt/SLES/POS/rsync/boot directory as initrd.gz and linux before Branch Servers can use the images to boot Point of Service terminals. For more information on this procedure, see “Copying Boot Images to the Administration Server’s RSYNC Directory” on page 118. For more information on the DiskNetboot image, see Section 4.2.1, “DiskNetboot,” on page 44.
java.xml	A child document for the Java Image Specification Document that includes the NLD RPMs in the Java client image. When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (--dist nld|sles). If you define the image distribution as NLD, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document. **NOTE:** NLD is the default distribution type. Therefore, if you do not specify the image distribution when cloning the Image Description Tree, xscr defaults to NLD. In general, most Point of Service images are created using the NLD distribution.

Directory	Description
/opt/SLES/POS/system/templates/support/	**continued**
java-base.xml	The base template Image Specification Document for the Java image. This file specifies the drivers and RPMs required to create the Java image. It is included as a child document in the ImageSpecification.xml document at the root of the Java Image Description Tree. It includes the X11 server and configuration.The Java image supports console-based C/C++ applications, Java programs in a Java2 runtime environment, and X11 applications. The required maximum size of the Java image is 100 MB compressed and 128 MB of RAM is required to boot the image.
java-sles.xml	A child document for the Java Image Specification Document that includes the SLES RPMs in the Java client image. When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (--dist nld|sles). If you define the image distribution as SLES, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document. The only Point of Service images that require the SLES distribution are POSBranch images. For more information on POSBranch, see Section 4.5, “POSBranch Images,” on page 51.
minimal.xml	A child document for the Minimal Image Specification Document that includes the NLD RPMs in the Minimal client image. When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (--dist nld|sles). If you define the image distribution as NLD, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document. **NOTE:** NLD is the default distribution type. Therefore, if you do not specify the image distribution when cloning the Image Description Tree, xscr defaults to NLD. In general, most Point of Service images are created using the NLD distribution. Novell Linux Point of Service includes a binary version of the Minimal NLD image that can be used for system testing. The binary file is /opt/SLES/POS/image/minimal-*version-date*.gz.

Directory	Description
/opt/SLES/POS/system/templates/support/	**continued**
minimal-base.xml	The base template Image Specification Document for the Java image. This file specifies the drivers and RPMs required to create the Java image. It is included as a child document in the ImageSpecification.xml document at the root of the Java Image Description Tree. It includes only the runtime environment for native code (that is C and C++) and the ncurses library for user interface support. The Minimal image supports only console-based applications. The required maximum size of the Minimal image is 35 MB compressed and 64 MB of RAM is required to boot the image.
minimal-sles.xml	A child document for the Java Image Specification Document that includes the SLES RPMs in the Java client image. When you clone an Image Description Tree using xscr, you can define the image distribution as NLD or SLES (--dist nld|sles). If you define the image distribution as SLES, xscr adds this child document to the IncludeSpecificationList element in the parent Image Specification Document. The only Point of Service images that require the SLES distribution are POSBranch images. For more information on POSBranch, see Section 4.5, “POSBranch Images,” on page 51.
/opt/SLES/POS/xml/	The xml directory contains files posldap2autoinstcd.pl requires to build Automatic Branch Server Installation images. For more information, see Section 10.3, “Building an Automatic Branch Server Installation Image,” on page 179.
add_harddisk.xsl	A stylesheet specifying the presentation of XML elements that define the hard disk configuration in the Automatic Branch Server Installation images, autoinst.ISO and autoinst.XML.
add_hd_partition.xsl	A stylesheet specifying the presentation of XML elements that define the hard disk partition configuration in the Automatic Branch Server Installation images, autoinst.ISO and autoinst.XML.
add_hostname.xsl	A stylesheet specifying the presentation of XML elements that create DNS hostname entries in the Automatic Branch Server Installation images, autoinst.ISO and autoinst.XML.
add_interface.xsl	A stylesheet specifying the presentation of XML elements that define the client interface (for example KDE or GNOME) in the Automatic Branch Server Installation images, autoinst.ISO and autoinst.XML.
add_routes.xsl	A stylesheet specifying the presentation of XML elements that define routes in the Automatic Branch Server Installation images, autoinst.ISO and autoinst.XML.

Directory	Description
/opt/SLES/POS/xml/	**continued**
template.xml	The Branch Server configuration template file. posldap2autoinstcd.pl uses this file to build the Automatic Branch Server Installation images, autoinst.ISO and autoinst.XML. For more information, see Section 10.3.3, “Modifying the Branch Server Configuration Template (template.xml),” on page 183.
/usr/bin/	
scr	The ImageBuilder tool used to build standard client images. For more information, see Chapter 8, “Building Images with the scr ImageBuilder Tool,” on page 97.
xscr	The ImageBuilder tool used to build client images. For more information, see Chapter 9, “Building Images with the xscr ImageBuilder Tool,” on page 121.

B.2 Branch Server Directory Structure

Directory	Contents
/etc/opt/SLES/POS/	
branchserver.conf	The LDAP base configuration file. Both the Administration and Branch Server use this file.
keys/	The keys directory contains the keys and certificates required to secure LDAP communication between the Administration and Branch Servers. During installation of the Administration Server, Novell Linux Point of Service automatically installs a CA and generates self-signed certificates to secure communication between the Administration and Branch Servers. However, the CA's public key is distributed to the branch servers only if you enable LDAP SSL during installation. For more information on setting up LDAP SSL, see “Running posInitLdap.sh” in the *Novell Linux Point of Service 9 Installation Guide*.
certs/	The certs directory contains the Administration Server certificate and keys.
ca.crt	The public key for the CA that signed the Administration Server's server certificate. This file is distributed to Branch Servers only if you enable LDAP SSL during installation of the Administration Server. The CA's public key allows the Branch Servers to trust the Administration Server.
/tftpboot/	The TFTP server directory on the Branch Server.

Directory	Contents
/tftpboot/	**continued**
CR/	The CR directory contains config.*MAC_ Address* image configuration files for every registered Point of Service terminal on the current Branch Server.
MAC_address/	The *MAC_address* directory contains system configuration files for individual Point of Service terminals, such as XF86config.
boot/	The boot directory contains the boot images and configuration files required to boot Point of Service terminals.
linux	The linux file is actually the DiskNetboot-*version-date*.kernel.*version*-SLRS image. The Linux image provides the Linux kernel used to PXE boot the Point of Service terminals. **IMPORTANT:** The DiskNetboot-*version-date*.kernel.*version*-SLRS image must be copied to the opt/SLES/POS/rsync/boot/ directory as linux before running posSynchImages.pl on the Branch Server. For more information on this process, see "Setting Up the Administration Server" in the *Novell Linux Point of Service 9 Installation Guide*.
initrd.gz	The initrd.gz file is actually the DiskNetboot.gz image. The initrd.gz image provides the second bootstrap image used to PXE boot the Point of Service terminals. **IMPORTANT:** The DiskNetboot.gz image must be copied to the opt/SLES/POS/rsync/boot/ directory as initrd.gz before running posSynchImages.pl on the Branch Server. For more information on this process, see "Setting Up the Administration Server" in the *Novell Linux Point of Service 9 Installation Guide*.
pxelinux.0	The pxelinux.0 image is the first boostrap image used to PXE boot the Point of Service terminals.
pxelinux.config/	The pxelinux.config directory contains the configuration files required to PXE boot the Point of Service terminals. Pxelinux.cfg files indicate which kernel and RAM disk to load for the Point of Service terminal. These files enable Branch Servers to distribute SLRS 8 and Novell Linux Point of Service 9 images. Novell Linux Point of Service automatically creates the pxelinux.cfg files based on the distribution container configurations in the LDAP directory. For more information, see Chapter 5, "The Novell Linux Point of Service LDAP Directory," on page 55.

Directory	Contents
/tftpboot/boot/pxelinux.config/	**continued**
default	The default configuration file for Point of Service terminal's PXE boot. This configuration file is used by all Point of Service terminals that boot the Novell Linux Point of Service 9 kernel.
IP_address_hex_value	Custom pxelinux configuration files. If you have a custom distribution container, Novell Linux Point of Service automatically generates custom pxelinux configuration files that distribute the kernel specified by that distribution directory. For example, if you migrate from SLRS 8 to Novell Linux Point of Service 9, the migration script creates the SLRS 8 distribution container. This container stores all the scPosImage objects for SLRS 8 images and points to the SLRS 8 kernel. The migration script also updates the existing scCashRegister objects to point to scPosImage objects in the SLRS 8 container. When you run posldap2crconfig or posleases2ldap, Novell Linux Point of Service automatically generates custom pxelinux configuration files for the Point of Service terminals that boot images in the SLRS 8 container. For more information, see “Migrating from SLRS 8 to Novell Linux Point of Service 9” in the *Novell Linux Point of Service 9 Installation Guide*. The filename for custom pxelinux configuration files is the IP address of the booting client. For example, if the client IP address is 10.1.1.1, the filename of the corresponding pxelinux configuration file is OA0101.
image/	The image directory contains the client images that are distributed to Point of Service terminals and their associated checksums. **NOTE:** On Novell Linux Point of Service, the Branch Server can simultaneously distribute SLRS 8 and Novell Linux Point of Service 9 terminal images.
upload/	The directory into which the hwtype.MAC_ Address files for newly registered Point of Service terminals are uploaded. These files are used to create the Point of Service terminal's workstation object in LDAP. This directory also stores temporary files for Image Install Notification. When a new image is installed on a Point of Service terminal, the *bootversion.MAC_address* file is created in tftpboot/upload. posleases2ldap transfers the information in the *bootversion.MAC_address* file to the scNotifiedimage attribute in the scWorkstation object in LDAP.

Sample Files

C

This section provides samples of the following XML template documents in a Novell® Linux Point of Service system:

C.1 Sample setup File

The setup file (`/opt/SLES/POS/system/`*`image_name-version`*`/setup`) is a configuration file that indicates which packages make up the image and which RPM options must be used to install them. Each package can also be accompanied by a specific version of the package.

The structure of the file is as follows:

`package_basename` `:` *`RPM_option`* `:` *`package_version`*

The following is a sample of the setup file:

```
aaa_base            :    x                    :          x
aaa_skel            :    x                    :          x
acl                 :    x                    :          x
ash                 :    x                    :          x
attr                :    x                    :          x
#atftp              :    x                    :          x
bash                :    x                    :          x
cracklib            :    x                    :          x
cyrus-sasl          :    x                    :          x
coreutils           :    x                    :          x
cpio                :    x                    :          x
db                  :    x                    :          x
devs                :    x                    :          x
dhcpcd              :    x                    :          x
diffutils           :    x                    :          x
e2fsprogs           :    x                    :          x
file                :    x                    :          x
filesystem          :    x                    :          x
fillup              :    x                    :          x
findutils           :    x                    :          x
logrotate           :    x                    :          x
gawk                :    x                    :          x
gdbm                :    x                    :          x
glibc               :    x                    :          x
glibc-locale        :    x                    :          x
grep                :    x                    :          x
gzip                :    x                    :          x
heimdal-lib         :    x                    :          x
iproute2            :    x                    :          x
```

```
insserv            :    x                      :           x
info               :    x                      :           x
kbd                :    x                      :           x
less               :    x                      :           x
libgcc             :    x                      :           x
libstdc++          :    x                      :           x
libxcrypt          :    x                      :           x
libcap             :    x                      :           x
libselinux         :    x                      :           x
libacl             :    x                      :           x
libattr            :    x                      :           x
libjpeg            :    x                      :           x
libtool            :    x                      :           x
liblcms            :    x                      :           x
libtiff            :    x                      :           x
mktemp             :    x                      :           x
module-init-tools  :    x                      :           x
mkinitrd           :    x                      :           x
mingetty           :    x                      :           x
ncurses            :    x                      :           x
net-tools          :    x                      :           x
netcfg             :    x                      :           x
openldap2-client   :    x                      :           x
openssh            :    x                      :           x
openssl            :    x                      :           x
openslp            :    x                      :           x
pam                :    x                      :           x
pam-modules        :    x                      :           x
perl               :    x                      :           x
permissions        :    x                      :           x
popt               :    x                      :           x
portmap            :    --nodeps,--noscripts   :           x
procps             :    x                      :           x
psmisc             :    x                      :           x
pwdutils           :    x                      :           x
resmgr             :    x                      :           x
readline           :    x                      :           x
sed                :    x                      :           x
sld-release        :    x                      :           x
sysconfig          :    x                      :           x
sysvinit           :    x                      :           x
submount           :    x                      :           x
tar                :    x                      :           x
tcpd               :    x                      :           x
timezone           :    x                      :           x
util-linux         :    x                      :           x
zlib               :    x                      :           x
XFree86            :    x                      :           x
XFree86-libs       :    x                      :           x
XFree86-fonts-scalable: x                      :           x
XFree86-server     :    x                      :           x
freetype2          :    x                      :           x
utempter           :    x                      :           x
compat             :    x                      :           x
```

```
unixODBC           :    x                    :           x
xntp               :    x                    :           x
syslogd            :    x                    :           x
iputils            :    x                    :           x
lukemftp           :    x                    :           x
hotplug            :    x                    :           x
lsof               :    x                    :           x
scsi               :    x                    :           x
usbutils           :    x                    :           x
setserial          :    x                    :           x
pciutils           :    x                    :           x
hwinfo             :    x                    :           x
openmotif          :    x                    :           x
udev               :    x                    :           x
openmotif-libs     :    x                    :           x
expat              :    x                    :           x
fontconfig         :    x                    :           x
cpp                :    x                    :           x
XFree86-Mesa       :    x                    :           x
libpng             :    x                    :           x
cabextract         :    x                    :           x
XFree86-server-glx :    x                    :           x
xf86tools          :    x                    :           x
libusb             :    x                    :           x
##### IBM  requires telnet##########
telnet             :    x                    :           x
wget               :    x                    :           x
rsync              :    x                    :           x
rsh                :    x                    :           x
#apmd              :    --nodeps,--force     :           x
```

C.2 Sample setup.user File

The setup.user file (`/opt/SLES/POS/system/`*`image_name-version`*`/setup.user`) is an optional configuration file that can be present in the Image Description Tree in addition to the setup file. It has the same format as setup, but it can indicate a path to the package after the package version.

The structure of the file is as follows:

```
Package Basename : RPM Option : Package Version : Path
```

The following is a sample of the `setup.user` file:

```
POS_evtouch_binary : x             : x
atftp              : x             : x
IBMJava2-JRE       :  --nodeps     : x
IBMJava2-JAVACOMM  :  --nodeps     : x
```

C.3 Sample ImageSpecification.xml Documents

Novell Linux client Image Specification Documents can be defined in an XML editor or in a standard text editor. XML editors provide the advantage of a graphical user interface. Typically, XML elements are presented as graphical objects and are visually organized in the schema hierarchy. Element attributes are defined as fields within the element objects. After the XML

template is defined, the template can be saved as a standard XML document. The graphics in Section 9.2.2, “Image Specification Documents,” on page 128 were taken in an XML editor. They show XML schema in a graphical format.

Novell Linux client Image Specification Documents can also be defined in a standard text editor. Text-based XML documents are more complicated because the schema hierarchy and element attributes are defined through the document syntax and organization. The following sample documents are presented in text format.

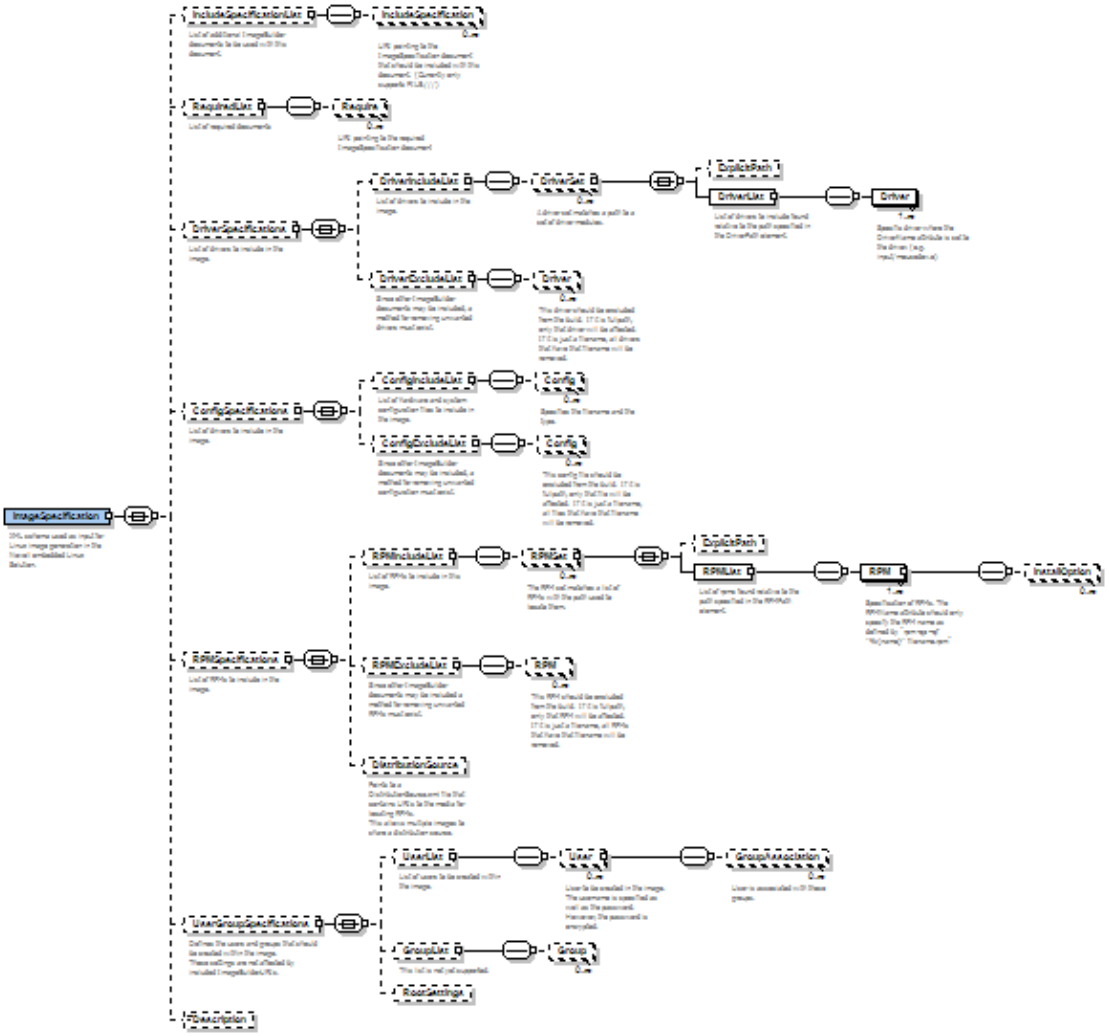
The following are examples of Novell Linux client Image Specification Documents:

- Section C.3.1, “ImageSpecification.xml Template,” on page 238
- Section C.3.2, “Defined ImageSpecification.xml Document,” on page 240

C.3.1 ImageSpecification.xml Template

Figure C-1 is a graphical representation of the Image Specification Document template. It demonstrates the organization of Novell Linux client Image Specification Documents.

Figure C-1 *ImageSpecification.sml schema structure*



The following is a text representation of the Image Specification Document template. It demonstrates the syntax of Novell Linux client Image Specification Documents; however, it is not populated with actual data.

```
<ImageSpecification xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="I:\ImageSpecification.xsd"
SchemaVersion="1" SchemaRevision="1" ImageName="String"
ImageType="diskful" ImageVersion="String" AddonSize="String"
Type="ext2" Timezone="String" Locale="String" Keytable="String">
      <IncludeSpecificationList>
                  <IncludeSpecification URI="String"/>
               </IncludeSpecificationList>
           <RequiredList>
         <Require URI="String"/>
     </RequiredList>
      <DriverSpecifications>
                    <DriverIncludeList>
                              <DriverSet Name="String">
                              <ExplicitPath URI="String"/>
                           <DriverList>
                           <Driver Name="String"/>
                        </DriverList>
                  </DriverSet>
               </DriverIncludeList>
            <DriverExcludeList>
                 <Driver Name="String"/>
           </DriverExcludeList>
      </DriverSpecifications>
           <ConfigSpecifications>
                    <ConfigIncludeList>
                       <Config Type="HWD" Name="String"/>
         </ConfigIncludeList>
                    <ConfigExcludeList>
           <Config Name="String"/>
                    </ConfigExcludeList>
        </ConfigSpecifications>
           <RPMSpecifications>
         <RPMIncludeList>
                  <RPMSet>
                        <ExplicitPath URI="String"/>
                           <RPMList>
                              <RPM Name="String" Version="String">
                              <InstallOption Option="--noscripts"/>
                              </RPM>
                           </RPMList>
                  </RPMSet>
           </RPMIncludeList>
           <RPMExcludeList>
                  <RPM Version="String" Name="String"/>
           </RPMExcludeList>
           <DistributionSource URI="String" ImageClass="String"/>
      </RPMSpecifications>
        <UserGroupSpecifications>
                  <UserList>
```

```
      <User Name="String" EncryptedPassword="String" HasPassword="1"
UserId="String" HomeDirectory="String" MainGroup="String">
                        <GroupAssociation Name="String"/>
                    </User>
            </UserList>
               <GroupList>
                        <Group Name="String" GroupId="String"/>
               </GroupList>
            <RootSettings DisableRootAccess="false"
EncryptedRootPassword="x"/>
         </UserGroupSpecifications>
           <Description>
String
</Description>
</ImageSpecification>
```

C.3.2 Defined ImageSpecification.xml Document

The following Image Specification Documents are populated with image data. They illustrate how an Image Specification Document can be defined.

Java Image Specification Document

```
<ImageSpecification xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="/opt/SLES/POS/
ImageSpecification.xsd" SchemaVersion="1" SchemaRevision="1"
ImageName="java" ImageVersion="2.0.21" Type="ext2" ImageType="diskful"
Timezone="US/Mountain" Locale="en_US" Keytable="us.map.gz">
   <IncludeSpecificationList>
      <IncludeSpecification URI="/opt/SLES/POS/system/templates
        /support/java.xml"/>
      <IncludeSpecification URI="/opt/SLES/POS/system/templates
        /addons/ibmjava.xml"/>
      <IncludeSpecification URI="/opt/SLES/POS/system/templates
        /addons/evtouch.xml"/>
      <IncludeSpecification URI="/opt/SLES/POS/system/templates
        /drivers/net.xml"/>
      <IncludeSpecification URI="/opt/SLES/POS/system/templates
        /drivers/scsi.xml"/>
      <IncludeSpecification URI="/opt/SLES/POS/system/templates
        /drivers/drivers.xml"/>
      <IncludeSpecification URI="/opt/SLES/POS/system/templates
        /drivers/usb.xml"/>
   </IncludeSpecificationList>
   <DriverSpecifications>
      <DriverIncludeList>
         <DriverSet Name="netdrivers">
         <ExplicitPath URI="/kernel/drivers/net/"/>
            <DriverList>
               <Driver Name="pcnet32.ko"/>
               <Driver Name="mii.ko"/>
               <Driver Name="natsemi.ko"/>
               <Driver Name="tulip/tulip.ko"/>
```

```
            <Driver Name="eepro100.ko"/>
            <Driver Name="e100.ko"/>
            <Driver Name="e1000/e1000.ko"/>
         </DriverList>
      </DriverSet>
      <DriverSet Name="drivers">
      <ExplicitPath URI="/kernel/"/>
         <DriverList>
            <Driver Name="net/packet/*"/>
            <Driver Name="fs/ext3/*"/>
            <Driver Name="fs/jbd/*"/>
            <Driver Name="drivers/ide/*"/>
            <Driver Name="drivers/dump/*"/>
            <Driver Name="drivers/cdrom/*"/>
            <Driver Name="drivers/char/dcs/*"/>
            <Driver Name="drivers/pci/*"/>
            <Driver Name="drivers/input/keyboard/ikbps.ko"/>
         </DriverList>
      </DriverSet>
      <DriverSet Name="usbdrivers">
      <ExplicitPath URI="/kernel/drivers/usb/"/>
         <DriverList>
            <Driver Name="host/uhci-hcd.ko"/>
            <Driver Name="host/ohci-hcd.ko"/>
            <Driver Name="core/usbcore.ko"/>
            <Driver Name="host/ehci-hcd.ko"/>
            <Driver Name="storage/usb-storage..ko"/>
            <Driver Name="input/hid.ko"/>
         </DriverList>
      </DriverSet>
      <DriverSet Name="scsidriver">
      <ExplicitPath URI="/kernel/drivers/scsi/"/>
         <DriverList>
            <Driver Name="scsi_mod.ko"/>
            <Driver Name="sg.ko"/>
            <Driver Name="sd_mod.ko"/>
            <Driver Name="st.ko"/>
            <Driver Name="sr_mod.ko"/>
         </DriverList>
      </DriverSet>
   </DriverIncludeList>
</DriverSpecifications>
<RPMSpecifications>
   <RPMIncludeList>
      <RPMSet>
         <RPMList>
            <RPM Name="acl"/>
            <RPM Name="atftp"/>
            <RPM Name="bash"/>
            <RPM Name="grep"/>
            <RPM Name="gzip"/>
            <RPM Name="ncurses"/>
            <RPM Name="netcfg"/>
            <RPM Name="openslp"/>
```

```
                <RPM Name="openssh"/>
                <RPM Name="openssl"/>
                <RPM Name="perl"/>
                <RPM Name="permissions"/>
                <RPM Name="readline"/>
                <RPM Name="resmgr"/>
                <RPM Name="sed"/>
                <RPM Name="setserial"/>
                <RPM Name="sld-release"/>
                <RPM Name="submount"/>
                <RPM Name="sysconfig"/>
                <RPM Name="syslogd"/>
                <RPM Name="tar"/>
                <RPM Name="tcpd"/>
                <RPM Name="usbutils"/>
                <RPM Name="zlib"/>
                <RPM Name="telnet"/>
             </RPMList>
          </RPMSet>
       </RPMIncludeList>
       <DistributionSource URI="/opt/SLES/POS/system/templates
         /Distribution.xml" ImageClass="NLD"/>
    </RPMSpecifications>
    <UserGroupSpecifications>
       <RootSettings DisableRootAccess="false" EncryptedRootPassword=
         "x"/>
    </UserGroupSpecifications>
</ImageSpecification>
```

Firefox Image Specification Document

The Firefox Image Specification Document adds the Firefox browser to a client image. The RequiredList element indicates that this add-on feature requires the `browser.xml` file; therefore, it can be added only to the Browser or Desktop client images.

```
<ImageSpecification xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="/opt/SLES/POS/
ImageSpecification.xsd" SchemaVersion="1" SchemaRevision="1">
   <Description>
   Mozilla Firefox, formerly known as Phoenix and Firebird, is a
   redesign of the Mozilla browser component, similar to Galeon,
   K-Meleon and Camino, but written using the XUL user interface
   language and designed to be cross-platform. It's a standalone
   application instead of part of the Mozilla Application Suite.
   </Description>
   <RequiredList>
   <Require URI="/opt/SLES/POS/system/templates/support/browser.xml"/>
   </RequiredList>
   <RPMSpecifications>
      <RPMIncludeList>
         <RPMSet>
            <RPMList>
               <RPM Name="MozillaFirefox"/>
               <RPM Name="gconf2"/>
```

```
                <RPM Name="gnome-vfs2"/>
                <RPM Name="libbonobo"/>
                <RPM Name="libgnome"/>
                <RPM Name="orbit2"/>
                <RPM Name="alsa"/>
                <RPM Name="audiofile"/>
                <RPM Name="esound"/>
                <RPM Name="libxml2"/>
                <RPM Name="libidl"/>
                <RPM Name="gnome-mime-data"/>
                <RPM Name="desktop-data-SLD"/>
                <RPM Name="bzip2"/>
                <RPM Name="cdparanoia"/>
                <RPM Name="fam"/>
                <RPM Name="libsmbclient"/>
                <RPM Name="dialog"/>
                <RPM Name="gtk2"/>
            </RPMList>
        </RPMSet>
    </RPMIncludeList>
  </RPMSpecifications>
</ImageSpecification>
```

C.4 Sample Distribution.xml Documents

Novell Linux Point of Service Distribution Source Documents can be defined in an XML editor or in a standard text editor. XML editors provide the advantage of a graphical user interface. Typically, XML elements are presented as graphical objects and are visually organized in the schema hierarchy. Element attributes are defined as fields within the element objects. After the XML template is defined, the template can be saved as a standard XML document. The graphics in Section 9.2.3, “Distribution Source Document (Distribution.xml),” on page 140 were taken in an XML editor. They show XML schema in a graphical format.

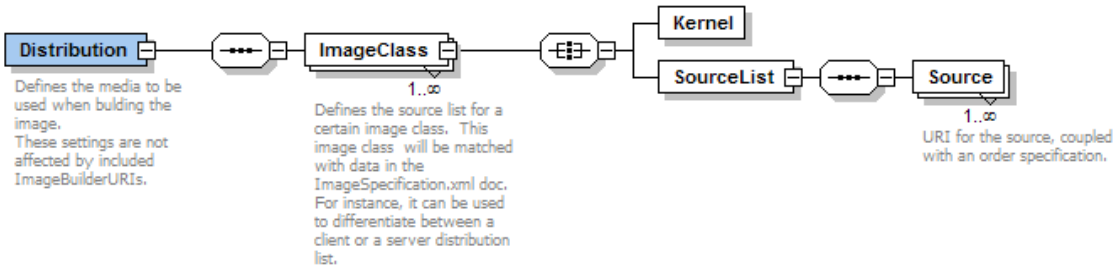
Distribution Source Documents can also be defined as XML documents in standard text format. These documents are more complicated because the schema hierarchy and element attributes are defined through the document syntax and organization. The following documents are presented in text format.

The following are examples of Distribution Source Documents:

C.4.1 Distribution.xml Template

Figure C-2 is a graphical representation of the Distribution Source Document template. It demonstrates the basic organization of Distribution Source Documents.

Figure C-2 *Distribution.xml schema structure*



The following is a textual representation of the Distribution Source Document template. It demonstrates the basic organization and syntax of Distribution Source Documents; however, it is not populated with actual data.

```
<xs:element name="Distribution">
   <xs:complexType>
      <xs:sequence>
        <xs:element name="ImageClass" maxOccurs="unbounded">
           <xs:complexType>
              <xs:all>
                 <xs:element name="Kernel">
                    <xs:complexType>
                    <xs:attribute name="Name" type="xs:string"
                       use="required"/>
                    <xs:attribute name="Version" type="xs:string"
                       use="required"/>
                    <xs:attribute name="Path" type="xs:string"
                       use="optional"/>
                    </xs:complexType>
                 </xs:element>
                 <xs:element name="SourceList">
                    <xs:complexType>
                    <xs:sequence>
                    <xs:element name="Source" maxOccurs="unbounded">
                    <xs:complexType>
                    <xs:attribute name="URI" type="xs:string"
                       use="required"/>
                    <xs:attribute name="Order" type="xs:int"
                       use="required"/>
                    </xs:complexType>
                    </xs:element>
                    </xs:sequence>
                    </xs:complexType>
                 </xs:element>
              </xs:all>
           <xs:attribute name="Name" type="xs:string" use="required"/>
           </xs:complexType>
        </xs:element>
```

```
      </xs:sequence>
      <xs:attribute name="SchemaVersion" type="xs:string"
         use="required" fixed="1"/>
      <xs:attribute name="SchemaRevision" type="xs:string"
         use="required" fixed="2"/>
   </xs:complexType>
</xs:element>
```

C.4.2 Defined Distribution.xml Document

The following Distribution Source Document is populated with image data. It illustrates how a Distribution Source Document can be defined.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- generated by poscdtool -->
<Distribution xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="/opt/SLES/POS/Distribution.xsd"
SchemaVersion="1" SchemaRevision="2">
   <ImageClass Name="nld">
      <Kernel Name="kernel-SLRS" Path="/opt/SLES/POS/pac"
           Version="2.6.5-7.155"/>
      <SourceList>
         <Source URI="/opt/SLES/POS/maintenance/nld" Order="0"/>
         <Source URI="/opt/SLES/POS/dist/NLD9/SP1/CD1" Order="1"/>
         <Source URI="/opt/SLES/POS/dist/NLD9/SP1/CD2" Order="2"/>
         <Source URI="/opt/SLES/POS/dist/NLD9/FCS/CD1" Order="3"/>
         <Source URI="/opt/SLES/POS/dist/NLD9/FCS/CD2" Order="4"/>
         <Source URI="/opt/SLES/POS/dist/NLD9/FCS/CD3" Order="5"/>
         <Source URI="/opt/SLES/POS/dist/NLPOS9/FCS/CD4" Order="6"/>
      </SourceList>
   </ImageClass>
   <ImageClass Name="sles">
      <Kernel Name="kernel-SLRS" Path="/opt/SLES/POS/pac"
           Version="2.6.5-7.155"/>
      <SourceList>
         <Source URI="/opt/SLES/POS/maintenance/sles" Order="0"/>
         <Source URI="/opt/SLES/POS/dist/NLPOS9/FCS/CD1" Order="1"/>
         <Source URI="/opt/SLES/POS/dist/NLPOS9/FCS/CD2" Order="2"/>
         <Source URI="/opt/SLES/POS/dist/NLPOS9/FCS/CD4" Order="3"/>
         <Source URI="/opt/SLES/POS/dist/SLES9/FCS/CD1" Order="4"/>
         <Source URI="/opt/SLES/POS/dist/SLES9/FCS/CD2" Order="5"/>
         <Source URI="/opt/SLES/POS/dist/SLES9/FCS/CD3" Order="6"/>
      </SourceList>
   </ImageClass>
</Distribution>
```

Documentation Updates

D

This section contains information about documentation content changes made to the *Novell Linux Point of Service 9 Administration Guide* since the initial release of Novell® Linux Point of Service 9. The information can help you to keep current on updates to the documentation.

All changes that are noted in this section were also made in the documentation. The documentation is provided on the Web in two formats: HTML and PDF. Both the HTML and PDF documentation are kept up-to-date with the documentation changes listed in this section.

The documentation update information is grouped according to the date the changes were published. Within a dated section, the changes are alphabetically listed by the names of the main table of contents sections for the guide.

If you need to know whether a copy of the PDF documentation you are using is the most recent, the PDF document contains the date it was published on the front title page.

This document was updated on the following dates:

D.1 August 15, 2006 (NLPOS 9 SSP3)

Throughout this document, version references have been updated, minor errors have been corrected, and the page design has been reformatted to comply with revised Novell documentation standards.

In addition, updates have been made to the following sections:

D.1.1 ImageBuilder Overview

This information was moved to Section 4.1, "Image Building Overview," on page 41 and the separate chapter was removed from the document.

D.1.2 Building Images with the xscr ImageBuilder Tool

The following changes have been made to this section:

Location	Change
Section 9.1, "xscr Commands," on page 121	Added information about new command line options for creating delta images and updating the product file.
Section 9.6, "Incremental Update," on page 165	This section is new.

Location	Change
Section 9.7, “Updating the Product File in a Boot Image,” on page 168	This section is new.

D.1.3 Backing Up System Information and Providing Access Control

This chapter has been renamed from “Best Practices” to better reflect the content.