

Novell Consulting Best Practices Guide

Automated Installation, Configuration, and Update for Open Enterprise Server 2015 SP1 Beta

January 2016

Novell®



Legal Notices

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see <https://www.novell.com/company/legal/>.

Copyright © 2016 Novell, Inc. All Rights Reserved.

Contents

About This Guide	7
1 Overview	9
Part I Using AutoYaST to Install Open Enterprise Server 11 or Later	11
2 AutoYaST Introduction	13
3 The AutoYaST Work Flow	15
3.1 Boot and Installation Process	15
3.1.1 Boot Process	15
3.1.2 Installation Process	15
3.2 AutoYaST Installation Process	15
4 Requirements for Unattended Installations via AutoYaST	17
4.1 Control File	17
4.2 Installation Repositories	17
4.2.1 Local Installation Repositories	18
4.2.2 Remote Installation Repositories	18
4.3 Network Repository Server	18
4.4 Installation Boot Medium	19
4.5 AutoYaST Control File	21
4.5.1 Control File Structure	23
4.5.2 Control File Repository	24
4.5.3 Retrieving a Control File	25
4.5.4 Summary	26
5 Installing and Configuring AutoYaST Components	27
5.1 Repository Server	27
5.1.1 Hardware Requirements	27
5.1.2 Software Requirements	28
5.2 Management Workstation	28
5.3 Operating System Setup	28
5.3.1 Disk Partition Layout	28
5.3.2 Pattern and Package Selection	29
5.3.3 Miscellaneous	29
5.4 Repositories	29
5.4.1 Installation Repositories	29
5.4.2 Control File and Classes Repository	31
5.4.3 Apache Web Server Configuration	32
5.4.4 Creating the Customized Boot Image	35
6 AutoYaST Extended: The Config File Approach	39
6.1 Design	39
6.1.1 Design Principles	40

6.1.2	Design Prerequisites	40
6.2	Implementation	41
6.2.1	Overview	41
6.2.2	Directory Structure	42
6.2.3	The Default File	44
6.2.4	pre-fetch.sh	45
6.2.5	Libraries	47
6.2.6	Configuration Files	47
6.2.7	XML Snippets	51
6.2.8	Post-Installation Scripts	53
7	Miscellaneous	55
7.1	SSH-Based Installation	55
7.2	Info File	56
7.3	Driver Updates	56
7.4	Boot Parameter y2confirm	56
7.5	Securely Downloading AutoYaST Control Files via HTTPS	57
7.6	Troubleshooting and Monitoring	57
7.6.1	Installation Server	57
7.6.2	Server Being Installed	57
7.7	Documentation	58
Part II	Using ZENworks 11 to Manage Open Enterprise Server 11 or Later	59
8	ZENworks Configuration Management Introduction	61
9	Server Installation	63
9.1	Prerequisites and Planning	63
9.1.1	Operating System	63
9.1.2	Quantity Structures	63
9.1.3	Network	65
9.1.4	Databases	66
9.1.5	Virtualization	67
9.1.6	Installation Worksheet for a Primary Server	67
9.1.7	Installation Worksheet for the Sybase Database	68
9.2	Server Installation Procedure	69
9.2.1	ZCM Primary Server Installation in GUI Mode	69
10	Server Configuration	79
10.1	Increasing the Content Replication Schedule	79
10.2	Configuring Content Primary Server Replication	80
10.3	Configuring the Inventory Service	81
10.4	Configuring the Inventory Schedule	83
10.5	Configuring the YUM Service Settings	84
10.6	Locations and Network Environments	84
10.6.1	Locations	85
10.6.2	Network Environment	87
11	Managing ZENworks Configuration Management	89
11.1	Overview	89
11.2	Folders	90

11.2.1	General Rules for Folder Names	90
11.2.2	Creating a New Folder	91
11.2.3	Deleting a Folder	92
11.2.4	Renaming a Folder	92
11.3	Devices	93
11.3.1	Folders in the Device Menu	93
11.3.2	Server Group Objects in the Device Menu	96
11.3.3	Device Registration	101
11.4	Bundles	109
11.4.1	Folders	109
11.4.2	Bundles and Bundle Groups	113
11.4.3	Creating a File Bundle	118
11.4.4	Bundle Groups and Bundles Developed by Novell Consulting	125
11.5	Subscriptions	142
11.5.1	Credential Vault	142
11.5.2	Proxy Setup	144
11.5.3	Creating a Subscription Folder Structure	144
11.5.4	Creating a Subscription	146
11.5.5	Pool Channels, Core Channels, and Online Channels	153
11.6	zman	154
12	Managing Linux Bundles	155
12.1	Frozen Patch Level	155
12.1.1	Naming Standards for Frozen Patch Levels	155
12.1.2	Creating a Frozen Patch Level	156
12.2	Bundle Deployment to SLES/OES Devices	160
12.2.1	Bundle Assignment	160
12.2.2	Manual Bundle Deployment	168
12.2.3	Scheduled Bundle Deployment	175
12.3	YUM Repositories Derived from a Frozen Patch Level	177
13	Agent Deployment	181
13.1	Manual Deployment	181
13.2	Deployment Task	183
13.3	Automated Deployment	190
14	Agent Commands	191
14.1	Core Commands	191
14.2	Linux Package Commands	191
14.3	Bundle Commands	192
15	Fault Diagnostics and Debugging	193
15.1	ZCM Server	193
15.1.1	ZENworks Control Center	193
15.1.2	ZENworks Web Service	193
15.1.3	CASA	193
15.1.4	Deployment	193
15.1.5	Discovery	194
15.1.6	zman	194
15.2	ZCM Agent	194
15.2.1	Changing the Log Level	194
15.2.2	Log Files	194

About This Guide



IMPORTANT: Although the latest version of OES mentioned in this guide is OES 11 SP1, the content and principles discussed apply equally well to OES 2015.

Based on the experiences gained and the challenges encountered over the past years, Novell Consulting Germany has created this *Novell Consulting Best Practices Guide* to share our knowledge and experience.

This guide is not a replacement for any training material. We highly recommend that you read the related [product documentation](http://www.novell.com/documentation/oes2015/index.html). (<http://www.novell.com/documentation/oes2015/index.html>) We assume that you already have a broad basic knowledge, so we cover only the specific details.

Moving from NetWare to Open Enterprise Server (OES) poses some major changes, which are mainly related to the change from the NetWare kernel to the SLES Linux kernel and the resulting differences in the implementation of the various OES services.

Throughout this guide, Open Enterprise Server refers to OES on the Linux kernel. Open Enterprise Server based on the NetWare kernel is always referred to as NetWare.

- ♦ [Chapter 1, “Overview,” on page 9](#)
- ♦ [Part I, “Using AutoYaST to Install Open Enterprise Server 11 or Later,” on page 11](#)
 - ♦ [Chapter 3, “The AutoYaST Work Flow,” on page 15](#)
 - ♦ [Chapter 4, “Requirements for Unattended Installations via AutoYaST,” on page 17](#)
 - ♦ [Chapter 5, “Installing and Configuring AutoYaST Components,” on page 27](#)
 - ♦ [Chapter 6, “AutoYaST Extended: The Config File Approach,” on page 39](#)
 - ♦ [Chapter 7, “Miscellaneous,” on page 55](#)
- ♦ [Part II, “Using ZENworks 11 to Manage Open Enterprise Server 11 or Later,” on page 59](#)
 - ♦ [Chapter 8, “ZENworks Configuration Management Introduction,” on page 61](#)
 - ♦ [Chapter 9, “Server Installation,” on page 63](#)
 - ♦ [Chapter 10, “Server Configuration,” on page 79](#)
 - ♦ [Chapter 11, “Managing ZENworks Configuration Management,” on page 89](#)
 - ♦ [Chapter 12, “Managing Linux Bundles,” on page 155](#)
 - ♦ [Chapter 13, “Agent Deployment,” on page 181](#)
 - ♦ [Chapter 14, “Agent Commands,” on page 191](#)
 - ♦ [Chapter 15, “Fault Diagnostics and Debugging,” on page 193](#)

Audience

This guide is primarily intended for skilled Novell NetWare administrators with a good basic knowledge of Linux who plan to migrate their Novell NetWare environment to Novell Open Enterprise Server.

Experienced Linux administrators who are interested in best practices for the deployment, configuration, and updating of SUSE Linux Enterprise Server (SLES) and OES systems should also read this guide.

Feedback

We want to hear your comments and suggestions about this guide and the other documentation included with this product. Please use the User Comments feature at the bottom of each page of the online documentation.

Documentation Updates

For the most recent version of this *Novell Consulting Best Practices Guide: Automated Installation, Configuration, and Update for OES 2015*, visit the [Open Enterprise Server 2015 Documentation web site \(https://www.novell.com/documentation/oes2015/mgmt_bp_guide_lx/data/bookinfo.html#bookinfo\)](https://www.novell.com/documentation/oes2015/mgmt_bp_guide_lx/data/bookinfo.html#bookinfo).

Additional Documentation

For the complete set of OES documentation, see the [Open Enterprise Server 2015 Documentation web site \(http://www.novell.com/documentation/oes2015\)](http://www.novell.com/documentation/oes2015).

1 Overview



In recent years, using automated, unattended installation methods for server installations and configurations have proven their value.

By enhancing SUSE's automatic installation method through AutoYaST and combining it with Novell ZENworks Configuration Management, we developed a way to easily implement standardized installation, configuration, and updates of SUSE Linux Enterprise Server (SLES) as well as Novell Open Enterprise Server (OES).

[Part I, "Using AutoYaST to Install Open Enterprise Server 11 or Later," on page 11](#) describes the Novell Consulting Installation Framework, which is based on AutoYaST. It explains how AutoYaST works and what is required to set up an installation server. Most of this section focuses on the customization of AutoYaST that has been developed and used by Novell Consulting.

[Part II, "Using ZENworks 11 to Manage Open Enterprise Server 11 or Later," on page 59](#) presents a methodology that uses ZENworks Configuration Management to provide centralized patch management and configuration management for SLES and OES servers.

In addition to the management of well-defined (frozen) patch levels for different staging areas such as development, test and production, this solution focuses on the management of configuration settings across many servers.

Using AutoYaST to Install Open Enterprise Server 11 or Later



Novell Consulting has developed a methodology to quickly and reproducibly install SUSE Linux Enterprise Servers (SLES) and Novell Open Enterprise (OES) through the AutoYaST framework.

- ♦ [Chapter 2, “AutoYaST Introduction,” on page 13](#)
- ♦ [Chapter 3, “The AutoYaST Work Flow,” on page 15](#)
- ♦ [Chapter 4, “Requirements for Unattended Installations via AutoYaST,” on page 17](#)
- ♦ [Chapter 5, “Installing and Configuring AutoYaST Components,” on page 27](#)
- ♦ [Chapter 6, “AutoYaST Extended: The Config File Approach,” on page 39](#)
- ♦ [Chapter 7, “Miscellaneous,” on page 55](#)

2 AutoYaST Introduction

In recent years, automated, unattended installation methods for server installations and configurations have proven to have value. They provide the following advantages:

- ♦ Installation standards can be implemented very easily.
- ♦ Servers can be configured and installed identically.
- ♦ Administrative efforts can be minimized.
- ♦ Large roll-outs become manageable in short time frames.
- ♦ Installations are reliable, traceable, and reportable.

To benefit from these advantages Novell Consulting recommends that you use AutoYaST for the SUSE automatic installation method in any environment where SUSE Linux Enterprise Server (SLES) and Open Enterprise Server (OES) are planned or are already deployed.

AutoYaST is also the basis of the Novell Consulting Installation Framework, a highly customized installation solution that has been developed and refined in a number of projects where Novell Consulting has been involved.

This framework requires the configuration of certain services, such as an installation repository, a service that provides remote access to the installation repository, a customized boot medium, and the control file for AutoYaST, which defines the main properties of the target devices.

This *Novell Consulting Best Practices Guide: Automated Installation, Configuration, and Update for OES 11* details the setup and basic mechanisms of this installation framework. The guide is not intended to replace the official installation documentation. Readers are expected to have a basic understanding of the SLES installation process and of AutoYaST.

The Novell Consulting Installation Framework is available as a Novell Cool Solution and is ready to be used after setting up your repository server and providing some customer-specific configuration information. See [The Novell Consulting Installation Framework—AutoYaST \(https://www.novell.com/communities/node/14216/novell-consulting-installation-framework-autoyast\)](https://www.novell.com/communities/node/14216/novell-consulting-installation-framework-autoyast).

3 The AutoYaST Work Flow

- ♦ [Section 3.1, “Boot and Installation Process,” on page 15](#)
- ♦ [Section 3.2, “AutoYaST Installation Process,” on page 15](#)

3.1 Boot and Installation Process

- ♦ [Section 3.1.1, “Boot Process,” on page 15](#)
- ♦ [Section 3.1.2, “Installation Process,” on page 15](#)

3.1.1 Boot Process

When a computer starts, certain BIOS routines are executed first in order to recognize and initialize hardware components such as the hard disk controllers, hard disks, network adapters, and so forth.

Afterwards, control of the boot process is taken over by the boot loader, which is located on a boot device such as a CD-ROM, hard disk, or network (PXE). The default boot loader for current SLES installations is GRUB. It starts a kernel and optionally an initial RAM disk (initrd).

Most operating systems accept parameters at the boot prompt to pass instructions to the kernel or `initrd`, which can influence such things as hardware initialization or function executions within the `initrd`.

3.1.2 Installation Process

In comparison to the boot process of a system already installed, other routines must be executed when you install a new system. Some of them are running in the background, and other routines might require manual administrative intervention. Most routines and modules that are necessary to install a new system on modern Linux distributions today are located within a special `initrd`, which is provided by the boot medium. This large `initrd` differs dramatically from the `initrd` used during a normal operating system boot.

Certain parameters specified at the boot prompt can influence how the system is installed. Some parameters ensure a proper network setup if necessary. Other parameters determine which installation repositories are used. An automated installation via AutoYaST must be initiated by a special boot parameter.

3.2 AutoYaST Installation Process

An unattended installation via AutoYaST is executed by specifying the following boot parameter:

```
autoyast=<URL to AutoYaST control file>
```

This parameter is recognized by the `initrd` and the execution process changes its direction. Instead of prompting administrators for various system settings, the control file located at the specified URL is parsed by YaST modules and the installation proceeds unattended by using all of the instructions defined in the AutoYaST control file.

4 Requirements for Unattended Installations via AutoYaST

- ♦ [Section 4.1, “Control File,” on page 17](#)
- ♦ [Section 4.2, “Installation Repositories,” on page 17](#)
- ♦ [Section 4.3, “Network Repository Server,” on page 18](#)
- ♦ [Section 4.4, “Installation Boot Medium,” on page 19](#)
- ♦ [Section 4.5, “AutoYaST Control File,” on page 21](#)

4.1 Control File

The single requirement for an AutoYaST installation is a valid control file. However, Novell Consulting recommends that you set up additional components to build a reliable, standardized, and easy-to-maintain installation framework.

4.2 Installation Repositories

Manual installations as well as unattended installations require one or multiple installation repositories that contain packages for the operating system to be installed (SLES).

Optionally, one or multiple add-on products (such as OES 11) and optional patches can be deployed as part of the installation.

The repository that contains the operating system packages and special metadata is mandatory. The other repositories mentioned are optional. The following repository types can be accessed by the installation engine:

- ♦ Local repositories located on CD-ROMs, DVDs, HDs, or USB devices.
- ♦ Remote repositories accessible by HTTP, FTP, NFS, SMB, or TFTP.

These repositories can be advertised via SLP.

Supported repository types are `yast2` and `rpm-md`. `yast2` repositories correspond to the SUSE installation media format and are the only repository type that can be used for operating system installations.

`rpm-md` (Repository Metadata) is the original repository type of YUM (Yellowdog Updater, Modified) and is supported only for the installation of add-on products or updates.

- ♦ [Section 4.2.1, “Local Installation Repositories,” on page 18](#)
- ♦ [Section 4.2.2, “Remote Installation Repositories,” on page 18](#)

4.2.1 Local Installation Repositories

Local installation repositories have advantages over remote repositories in rare cases where network bandwidth is a limiting factor.

One example is the installation of a branch office server. Novell Consulting has had very good experiences with the use of internal USB sticks to store the repositories to install this type of server.

4.2.2 Remote Installation Repositories

Wherever possible, remote installation repositories are recommended by Novell Consulting for an AutoYaST framework because they provide the following advantages:

- ♦ Independence from physical boot media
- ♦ Only one boot medium is required. Everything else is retrieved via the network
- ♦ Higher throughput can be achieved in most environments by network installations
- ♦ Multiple products can be installed together without the need to change the installation media (such as SLES 10 SP4 and OES 2 SP3, or SLES 11 SP2 and OES 11 SP1)
- ♦ A central repository for all servers prevents the necessity to distribute physical installation media to all systems
- ♦ Installation sources can be used for later deployments of additional software packages without the need to swap physical media

A remote installation repository must be specified at the boot prompt as follows:

```
install=<protocol>://<IP Address|DNS-Name>/<path to media content>
```

For example:

```
install=http://10.10.10.221/sles10sp4_x86_64
```

4.3 Network Repository Server

As pointed out in [“Installation Repositories” on page 17](#), various protocols can be selected to access network repositories. This requires you to set up an appropriate service, such as a Web server, FTP server, or SMB server.

Based on experiences gained over the past years Novell Consulting recommends that you use the HTTP protocol and the Apache 2 Web server to configure access to remote repositories.

The Apache 2 Web server provides the following advantages in comparison to other servers and protocols:

- ♦ Easy configuration
- ♦ Apache 2 is part of all server distributions delivered by Novell and SUSE
- ♦ Extensive logging mechanisms
- ♦ Symbolic links are supported
- ♦ Most administrators are familiar with configuration aspects of Apache 2

4.4 Installation Boot Medium

A boot medium is required to invoke the installation process by loading a kernel and the initrd that contains the installation logic.

As with repositories, a large diversity of available media exists. Servers can be booted from floppy disk, CD, DVD, USB devices, over the network (PXE), or from a local hard drive.

Our experience indicates that in most environments images or physical media are utilized for installations via remote management connections like ILO boards. In some rare cases, PXE is used.

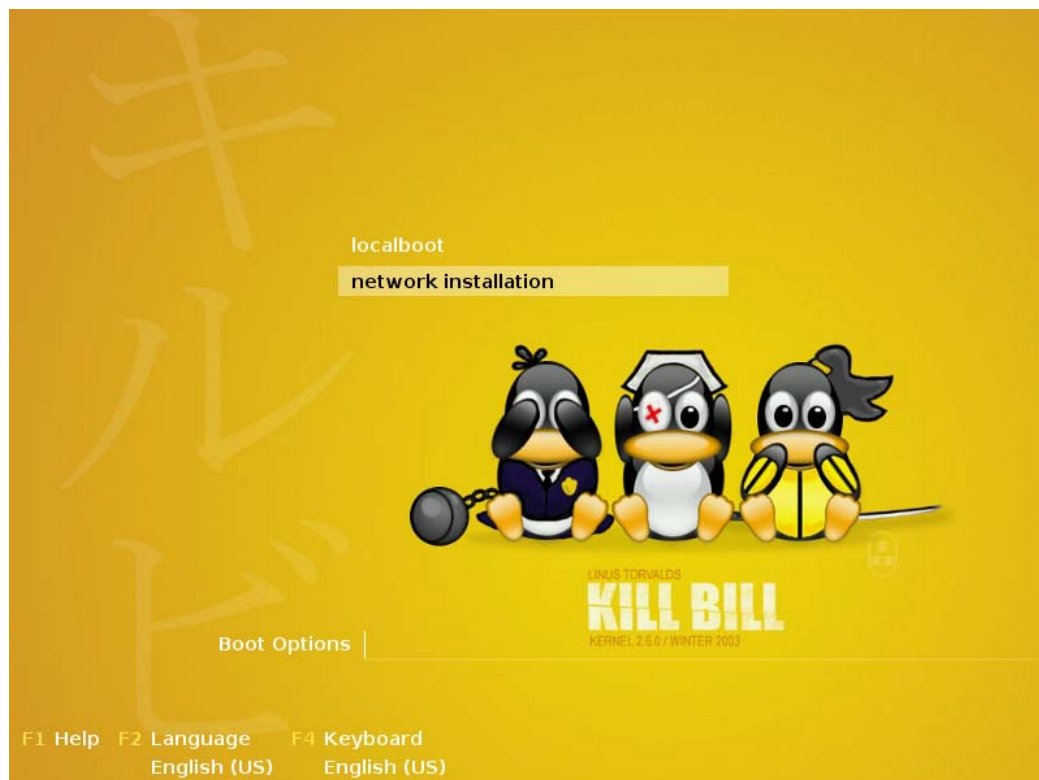
The standard image-based installation media is a normal SLES CD/DVD or an ISO image of the installation media provided by Novell via the download channels.

However, Novell Consulting favors a customized boot image with nested boot menus for the following reasons:

- ♦ Boot options/parameters can be precoded to the boot prompt.
- ♦ Multiple SLES versions can be installed via one single boot image, which was not possible with recent installation media of SLES.
- ♦ The image requires only a small number of kernels, initrds, and some boot loader data. This allows you to reduce the image size to just a few MB, depending on the number of SLES versions and SLES service packs that need to be supported.
- ♦ Customized menus can be created to reflect customer needs.
- ♦ Background images can be included to reflect corporate identity.

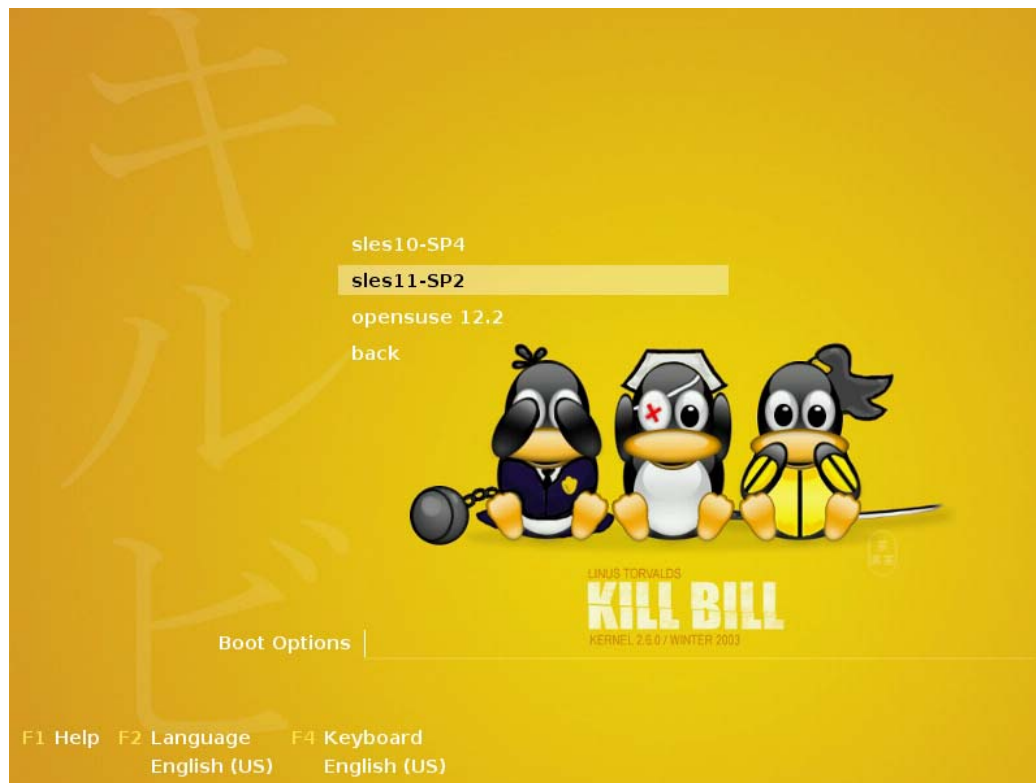
This type of customized boot image with nested menus and predefined boot parameters is illustrated in the following figures:

Figure 4-1 Customized Boot Image With Nested Menus (1)



To avoid loops in the installation process, you should provide an option for a local boot in the main menu and make it the default.

Figure 4-2 Customized Boot Image With Nested Menus (2)



Multiple operating systems can be chosen from the menu displayed in the previous figure. *Back* allows you to return to the main menu.

Figure 4-3 Customized Boot Image With Nested Menus (3)



This figure illustrates that boot options can be predefined for each menu selection. The menu again contains a *Back* option to let administrators correct wrong menu selections.

A detailed description of how to create and customize such an installation medium is found in section [Section 5.4.4, “Creating the Customized Boot Image,” on page 35](#).

4.5 AutoYaST Control File

An AutoYaST control file is in XML format. It can contain just a few or up to hundreds of instructions that determine the details of the system being installed.

In theory, a control file can be created manually with an editor. Because this approach is very error-prone, it is not recommended.

The easiest way to obtain a control file is to use the `/root/autoinst.xml` file as a template, if this file has been saved at the end of a manual installation by selecting the appropriate option.

This method should be used with great care, because the resulting XML file contains many settings that are specific for the machine where it was generated.

Control files can also be created via YaST (*YaST/YaST2 > Miscellaneous > Autoinstallation*), as illustrated in the following figures. The `autoyast2` and `autoyast2-installation` packages must be installed for this functionality to be available in YaST.

Figure 4-4 Invoking YaST > Miscellaneous > Autoinstallation

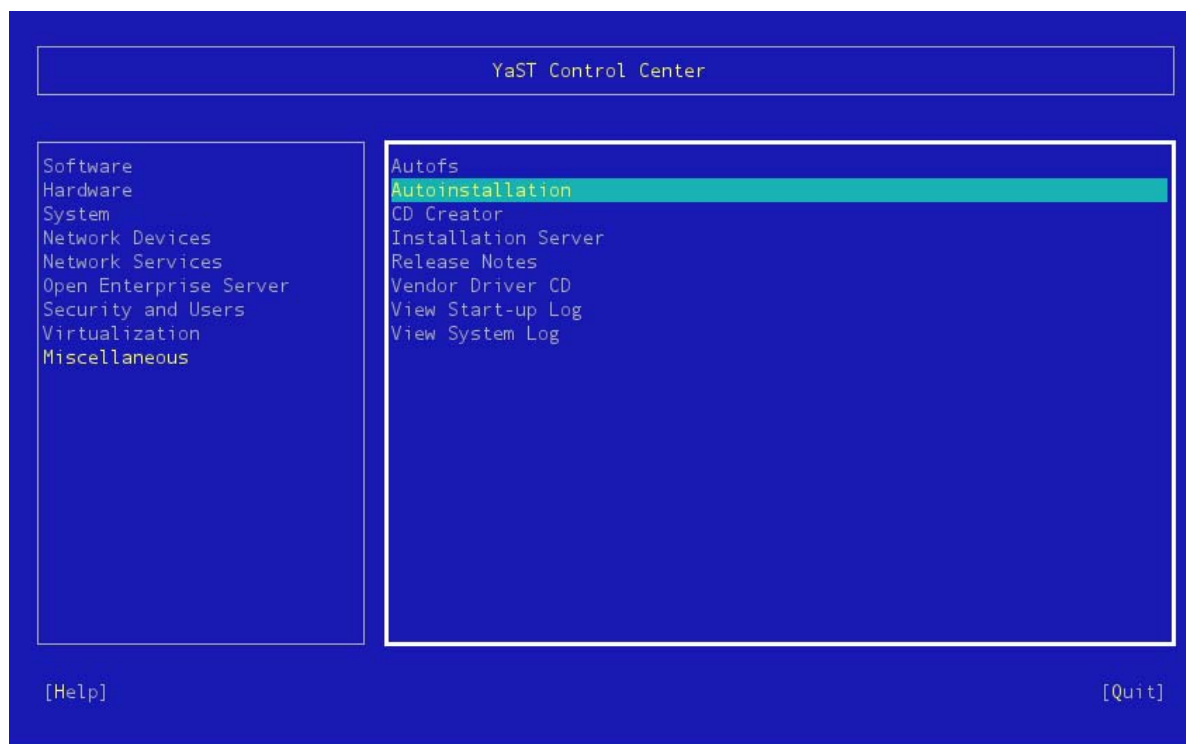
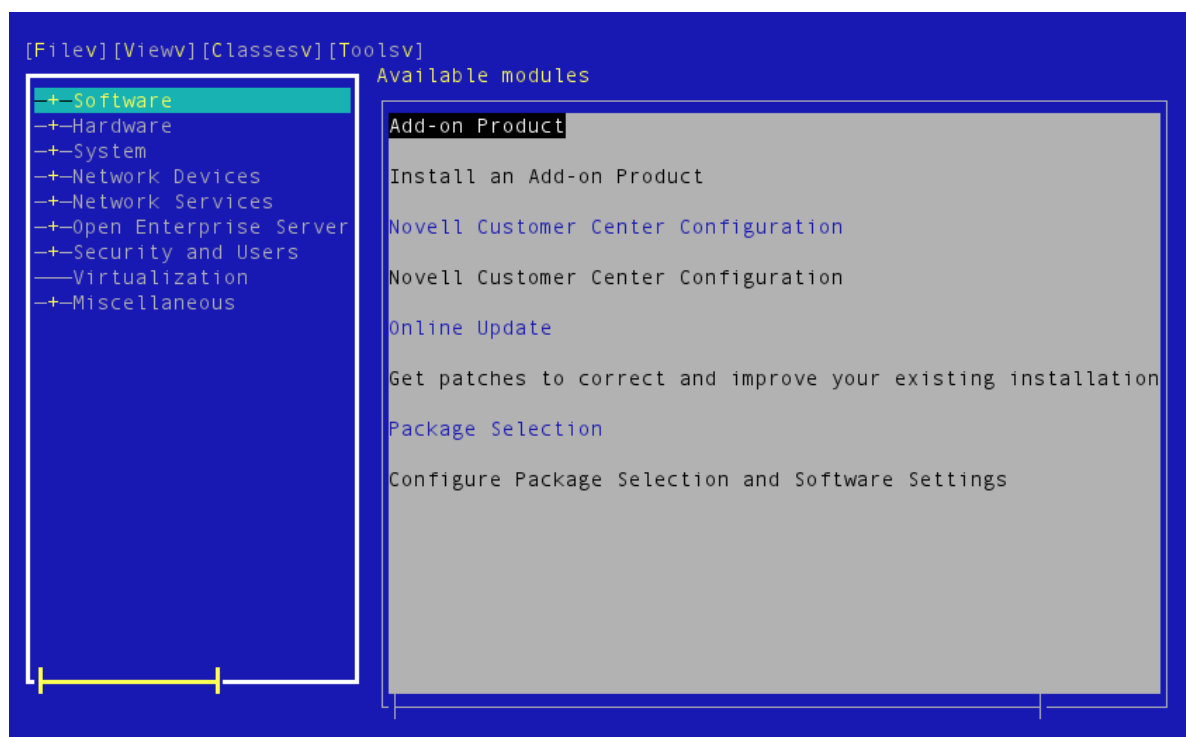


Figure 4-5 Autoinstallation > Configuration Items



With the YaST Autoinstallation plug-in, complete control files can be created by specifying all necessary configuration aspects or by cloning the device where YaST is being executed (YaST > *Miscellaneous* > *Autoinstallation* > *Tools* > *Create Reference Profile*). Furthermore, single class files also can be created if you are interested only in the configuration aspects of a particular module.

- ♦ [Section 4.5.1, “Control File Structure,” on page 23](#)
- ♦ [Section 4.5.2, “Control File Repository,” on page 24](#)
- ♦ [Section 4.5.3, “Retrieving a Control File,” on page 25](#)
- ♦ [Section 4.5.4, “Summary,” on page 26](#)

4.5.1 Control File Structure

An AutoYaST control file must be written in valid XML syntax. It is organized in various sections that are responsible for different aspects of the installation. The following sections could be part of an AutoYaST file that drives a SLES 11 installation:

- ♦ add-on
- ♦ boot loader
- ♦ CA management
- ♦ general
- ♦ host
- ♦ networking
- ♦ partitioning
- ♦ users
- ♦ ...

The example below shows an XML snippet to configure the YaST Certificate Authority section for SLES 11:

```
<?xml version="1.0"?>
<!DOCTYPE profile>
<profile xmlns="http://www.suse.com/1.0/yast2ns" xmlns:config="http://
www.suse.com/1.0/configs">
  <ca_mgm>
    <CAName>YaST_Default_CA</CAName>
    <ca_commonName>my-company.us</ca_commonName>
    <country>US</country>
    <importCertificate config:type="boolean">false</importCertificate>
    <locality>Provo</locality>
    <organisation>IT</organisation>
    <organisationUnit>Servermanagement</organisationUnit>
    <password>secret</password>
    <server_email>jclarc@novell.com</server_email>
    <state>UTAH</state>
    <takeLocalServerName config:type="boolean">true</takeLocalServerName>
  </ca_mgm>
</profile>
```

AutoYaST does not need to know every configuration detail. If configuration items are omitted from the control file, AutoYaST configures the system by using meaningful default values for the configuration items of the affected section. This works for most sections but can lead to unexpected settings in some circumstances. Therefore, Novell Consulting recommends that you create the appropriate configuration tags for every section where the target settings deviate from defaults.

Only a few instructions within a control file are specific for a given server. Most of them can be re-used. To support this, AutoYaST offers the possibility to divide sections into external class files. This procedure is also referred to as class-based AutoYaST installation.

A class file must also be in valid XML syntax and can contain only a single or multiple sections of the complete AutoYaST control file.

Classes have a few advantages in contrast to monolithic AutoYaST control files:

- Code is easier to maintain because class files contain only parts of the whole
- Classes can be reused for different installation scenarios; for example the NTP class can be used for the installation of different versions of SLES and OES

When you use this technique, it is possible to start with a minimal control file containing only class directives (see the control file default in [“Retrieving a Control File” on page 25](#)). The real configuration information is then provided by the class files.

A class within a control file must be specified by a class directive. This directive has attributes that define the location of the external class file. When a class configuration tag is found during control file processing, the corresponding external class file is loaded from the specified location and merged into the main control file temporarily stored in `/tmp/profile/autoinst.xml` on the system being installed. This control file is parsed in the same manner as a monolithic control file when using the non-class based approach.

For using classes, some preparations must be made at the AutoYaST repository:

- All class files must be located in a directory named `classes` immediately underneath the URL that was given as AutoYaST URL at the boot prompt. For example, if the AutoYaST URL has been specified as `http://10.10.10.221/xml/autoinst.xml`, the class files must be located in the `http://10.10.10.221/xml/classes` directory.
- A class directive always contains a `class_name` tag and a configuration tag:

```
<class>
  <class_name>general</class_name>
  <configuration>general.xml</configuration>
</class>
```

- A `class_name` tag represents a directory underneath the `classes` directory. For example, if `general` is used as the `class_name` tag, the following directory must exist:

```
http://10.10.10.221/xml/classes/general
```

- A configuration tag specifies the name of the XML file in the directory defined by the `class_name` tag. In the above example, the `general.xml` class file would be accessed through the following URL:

```
http://10.10.10.221/xml/classes/general/general.xml
```

A complete set of class files used in various customer projects by Novell Consulting is part of the Novell Cool Solutions. See [The Novell Consulting Installation Framework—AutoYaST \(https://www.novell.com/communities/node/14216/novell-consulting-installation-framework-autoyast\)](https://www.novell.com/communities/node/14216/novell-consulting-installation-framework-autoyast).

4.5.2 Control File Repository

An AutoYaST file can be located on the installation medium itself, on local devices, or at various network locations similar to installation repositories as described in [“Installation Repositories” on page 17](#).

4.5.3 Retrieving a Control File

The URL for a control file must be specified by one of the following syntaxes:

`autoyast=<protocol>://<IP Address|DNS-Name>/<path to control file or directory/>,`

For example:

`autoyast=http://10.10.10.221/xml/mycontrolfile`

or

`autoyast=http://10.10.10.221/xml/`

The first example points directly to a control file at the repository server. This locates, downloads, and processes the AutoYaST `mycontrolfile` control file.

The second example points to a directory, resulting in the following retrieval process:

1. A file whose name is equivalent to the hexadecimal value of the IP address of the system to be installed is searched in the specified directory. For example:

`192.168.2.91 -> http://10.10.10.221/xml/C0A8025B`

2. If this file does not exist, the last character of the hexadecimal value is removed and the system searches for the resulting file name. For example:

`http://10.10.10.221/xml/C0A8025`

3. If the file does not exist, the process is repeated until the file name has been truncated to one character. For example:

`http://10.10.10.221/xml/C0A802
http://10.10.10.221/xml/C0A80
http://10.10.10.221/xml/C0A8
http://10.10.10.221/xml/C0A
http://10.10.10.221/xml/C0
http://10.10.10.221/xml/C`

4. If a file with a name matching the name derived by this process can be located, the unattended installation starts. If no file can be found, the process continues.
5. The system tries to retrieve a file with a name matching the MAC address of the network adapter from which the connection to the Web server has been initiated.

Two attempts are made: First, it looks for a file name with characters in all uppercase is looked for. If it is not found, a file name with characters in all lowercase is searched for. For example:

`http://10.10.10.221/xml/0080C8F6484C
http://10.10.10.221/xml/0080c8f6484c`

6. Finally, if this still does not deliver a control file, the process looks for a file named `default` in the directory that has been specified with the `autoyast=` parameter. For example:

`http://10.10.10.221/xml/default`

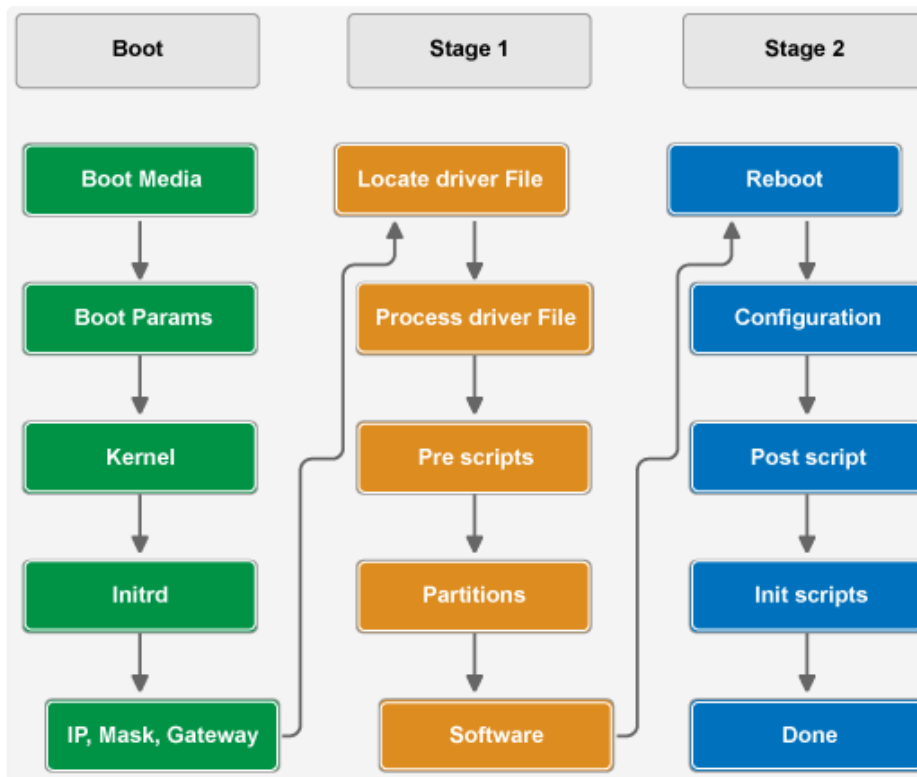
7. If the `default` file is found, it is evaluated and its instructions are used for the installation process. No further control file is searched for.

If the installation engine has not located a control file at this point, the installation process terminates and informs the administrator that the URL for the AutoYaST file is invalid.

Because the directory method provides more flexibility than specifying a file name, Novell Consulting has chosen this variant for all customer projects.

The following illustration summarizes the components and steps of a server installation using AutoYaST.

Figure 4-6 AutoYaST Process Overview



4.5.4 Summary

This section outlined the basic principals and components for the Novell Consulting Installation Framework. It is characterized by the following properties:

- ♦ Remote installation repositories are provided via an Apache 2 Web server
- ♦ Class-based implementation of the AutoYaST control file is used on the same Apache 2 server
- ♦ Customized boot media with predefined boot parameters and nested menus are used to initiate the installation process
- ♦ The AutoYaST URL specifies a directory where the initial control file default points to particular class files

The rest of the sections in this book describe how these components need to be set up and configured in detail.

5 Installing and Configuring AutoYaST Components

This section details the setup and configuration of the Novell Consulting Installation Framework. In addition, some standards are introduced (such as a naming standard for configuration paths, Apache 2 aliases, and so on). The goal is to provide a standardized installation environment that is easy to maintain and to adjust.

- ♦ [Section 5.1, “Repository Server,” on page 27](#)
- ♦ [Section 5.2, “Management Workstation,” on page 28](#)
- ♦ [Section 5.3, “Operating System Setup,” on page 28](#)
- ♦ [Section 5.4, “Repositories,” on page 29](#)

5.1 Repository Server

The repository server needs to meet the following specifications:

- ♦ [Section 5.1.1, “Hardware Requirements,” on page 27](#)
- ♦ [Section 5.1.2, “Software Requirements,” on page 28](#)

5.1.1 Hardware Requirements

- ♦ [“Physical or Virtual” on page 27](#)
- ♦ [“RAM and CPU” on page 27](#)
- ♦ [“Network” on page 27](#)
- ♦ [“Disk Capacity” on page 28](#)

Physical or Virtual

The AutoYaST repositories do not require high-performance hardware. The selected system must support an Apache 2 Web server. In many environments, a virtual instance on VMware, XEN, KVM, or other type of virtualization has been used successfully as a AutoYaST server.

RAM and CPU

A single processor and 4 GB of RAM meet all performance requirements.

Network

A 100 Mbits⁻¹ network adapter typically meets all bandwidth requirements. With many concurrent installations, it might be helpful to deliver the software packages via a gigabit network interface.

Disk Capacity

The disk space required for the repositories depends on the number and the diversity of the SLES/OES versions, service packs, and add-on products that need to be supported. In recent projects, 50 GB has been sufficient.

5.1.2 Software Requirements

Novell Consulting recommends that you always install the newest SLES version with the latest service pack and current patches as the basis for the repository server (SLES 11 SP2 at the time this document was written).

5.2 Management Workstation

A SUSE Linux workstation or server can be used to create and maintain the customized boot image. This could be the repository server itself, any other SLES server, or even an OpenSUSE workstation.

The following packages need to be installed on this system:

- ♦ `mkisofs` (part of the `cdrkit-cdrtools-compat` package)
- ♦ `grub`

5.3 Operating System Setup

- ♦ [Section 5.3.1, “Disk Partition Layout,” on page 28](#)
- ♦ [Section 5.3.2, “Pattern and Package Selection,” on page 29](#)
- ♦ [Section 5.3.3, “Miscellaneous,” on page 29](#)

5.3.1 Disk Partition Layout

Novell Consulting recommends the use of the Linux Volume Manager LVM for the system partitions (except `/boot`) and the data partition that accommodates the AutoYaST server. The following minimum partition layout is suggested:

Table 5-1 *Partition Layout for the Installation Server*

Partition	Type	Volume Group	Volume	Mount Point	File System	Size [GB]
1	Linux	n/a	n/a	/boot	ext3	0.2
2	LVM	system	swap	swap	swap	4
2	LVM	system	root	/	ext3	10
3	LVM	data	data	/data	ext3	50

5.3.2 Pattern and Package Selection

The server should be installed with a minimal software selection only. Therefore, the Minimal pattern must be chosen during installation.

In addition, the `apache2` package with all its dependencies must be installed. The `cdkit-cdrtools-compat` package is required if the repository server is also used to create and maintain the boot media.

5.3.3 Miscellaneous

All other configurations of the operating system are kept at their default settings. You must ensure that the required ports (usually port 80) are not blocked by any firewall mechanism. The server should get a DNS entry following the local naming standard. In addition, a reverse lookup entry must be created in the customer's Domain Name Service.

5.4 Repositories

- [Section 5.4.1, “Installation Repositories,” on page 29](#)
- [Section 5.4.2, “Control File and Classes Repository,” on page 31](#)
- [Section 5.4.3, “Apache Web Server Configuration,” on page 32](#)
- [Section 5.4.4, “Creating the Customized Boot Image,” on page 35](#)

5.4.1 Installation Repositories

An installation repository contains the same file structure as delivered with the corresponding installation media. The simplest way would be to copy the content of all DVDs for SLES 11 SP2 and OES 11 SP1 into a directory structure and to provide remote access to these directories over HTTP. However, to protect against unintended alterations of installation sources, image files are copied to a directory and their content presented via read-only loop mounts.

Because current SLES products only support eight loop devices by default, this value needs to be increased in `/etc/modprobe.d/` as follows:

```
options loop max_loop=64
```

- [“Directory Layout for the Installation Repositories” on page 29](#)
- [“/etc/fstab Entries for the Installation Repositories” on page 30](#)

Directory Layout for the Installation Repositories

The directory structure for the installation repository is defined as follows:

- All installation data is stored below `/data`
- ISO images are placed in the `/data/isos` directory
- For loop mounts, the following directory structure needs to be created:
 - The start directory is `/data/install`
 - A mount point consisting of the architecture, product name, version, service pack, and media number needs to be created in the start directory. For example, a SLES 11 SP2 image of architecture `x86_64` would be mounted below the `/data/install/x86_64/sles11/sp2/cd1` structure.

- Directories for initial releases are named `ga`
- If an image contains both architectures (such as SLEPOS), the architecture directory is named `biarch`

The suggested directory layout is summarized in the following table:

Table 5-2 Directory Layout for the Installation Repositories

Directory	Description
<code>/data</code>	Start directory
<code>/data/isos</code>	Contains ISO images
<code>/data/install</code>	Contains mount points for ISO images
<code>/data/install/x86_64</code>	Architecture directory
<code>/data/install/i586</code>	Architecture directory
<code>/data/install/biarch</code>	Architecture directory
<code>/data/install/x86_64/sles11</code>	Product + version directory
<code>/data/install/x86_64/oes11</code>	Product + version directory
<code>/data/install/x86_64/sles11/sp2</code>	SLES 11 service pack directory
<code>/data/install/x86_64/oes11/sp1</code>	OES 11 service pack directory
<code>/data/install/x86_64/sles11/sp2/cd1</code>	SLES 11 SP2 media directory
<code>/data/install/x86_64/oes11/sp1/cd1</code>	OES 11 SP1 media directory

/etc/fstab Entries for the Installation Repositories

Loop mounts that are required to persist a server boot must be entered in `/etc/fstab`:

- The full name to the image must be specified in column 1.
- Column 2 holds the corresponding mount point.
- In column 3, the file system type must be specified. It is `iso9660` for image files.
- Column 4 defines the mount properties for the mount point, which should be `auto,ro,loop`
- The last column determines whether the appropriate file system should be checked on boot or not. `0 0` (no file system check on boot) is the recommended entry here.

An example how to specify loop mounts in `/etc/fstab` is given below:

Figure 5-1 Sample `fstab` Entries for an Installation Server

```

File Edit View Bookmarks Settings Help
192.168.1.255: root
/dev/disk/by-id/ata-ST3500320NS_9QMAXC75-part1 /boot ext3 noatime,noacl 1 2
/dev/system/swap swap defaults 0 0
/dev/system/root / ext3 defaults 1 1
/dev/system/var /var ext3 defaults 1 2
proc /proc proc defaults 0 0
sysfs /sys sysfs noauto 0 0
debugfs /sys/kernel/debug debugfs noauto 0 0
usbfs /proc/bus/usb usbfs noauto 0 0
devpts /dev/pts devpts mode=0620,gid=5 0 0

/data/isos/SLES-10-SP4-DVD-x86_64-GM-DVD1.iso /data/install/x86_64/sles10/sp4/cd1 iso9660 auto,ro,loop 0 0
/data/isos/oes2sp3-x86_64-CD1.iso /data/install/x86_64/oes11/sp1/cd1 iso9660 auto,ro,loop 0 0
/data/isos/SLES-11-SP2-DVD-x86_64-GMC-DVD1.iso /data/install/x86_64/sles11/sp2/cd1 iso9660 auto,ro,loop 0 0
/data/isos/install/OES11-SP1-addon-x86_64-CD1.iso /data/install/x86_64/oes11/sp1/cd1 iso9660 auto,ro,loop 0 0

```

5.4.2 Control File and Classes Repository

The repository for the control files contains the main control file (`default`) which contains class declarations and the individual class files located in the directory structure outlined below.

- ♦ [“Directory Layout for the Control File Repository” on page 31](#)

Directory Layout for the Control File Repository

In parallel to the installation repositories, some standards are defined for control files:

- ♦ The top-level directory for AutoYaST is `/data/autoyast`
- ♦ Control files are stored in `/data/autoyast/xml` and its subdirectories
- ♦ The main control file is `/data/autoyast/xml/default`
- ♦ All classes must be located underneath `/data/autoyast/xml/classes` because this directory name is hard-coded in the AutoYaST plug-ins
- ♦ Below the `classes` directory, the Novell Consulting Installation Framework currently requires the following subdirectories: `ask`, `general`, `scripts`

The following table summarizes the directory layout for control files:

Table 5-3 *Directory Layout for the Control File Repository*

Directory	Description
<code>/data/autoyast</code>	Base directory
<code>/data/autoyast/xml</code>	Contains the <code>classes</code> directory and the main control file
<code>/data/autoyast/xml/classes</code>	Contains the class subdirectories
<code>/data/autoyast/xml/classes/ask</code>	Contains all classes tagged with <code>class_name ask</code>
<code>/data/autoyast/xml/classes/general</code>	Contains all classes tagged with <code>class_name general</code>
<code>/data/autoyast/xml/classes/scripts</code>	Contains all classes tagged with <code>class_name scripts</code>
<code>/data/autoyast/xml/classes/general/net.xml</code> <code>/data/autoyast/xml/classes/general/...</code>	Single class files tagged with <code>class_name general</code>

The following example is for the main control file that sources particular class files:

```
<?xml version="1.0"?>
<!DOCTYPE profile>
<profile xmlns="http://www.suse.com/1.0/yast2ns" xmlns:config="http://
www.suse.com/1.0/configns">
  <classes config:type="list">
    <class>
      <class_name>general</class_name>
      <configuration>bootloader.xml</configuration>
    </class>
    <class>
      <class_name>general</class_name>
      <configuration>ca_mgm.xml</configuration>
    </class>
    <class>
      <class_name>general</class_name>
      <configuration>general.xml</configuration>
    </class>
    <class>
      <class_name>general</class_name>
      <configuration>net.xml</configuration>
    </class>
    <class>
      <class_name>scripts</class_name>
      <configuration>scripts.xml</configuration>
    </class>
    <class>
      <class_name>general</class_name>
      <configuration>system.xml</configuration>
    </class>
  </classes>
</profile>
```

5.4.3 Apache Web Server Configuration

- ♦ [“Global Configuration” on page 32](#)
- ♦ [“Directory Configuration” on page 33](#)
- ♦ [“Alias Configuration” on page 33](#)
- ♦ [“Validating the Apache Configuration for the Repository” on page 34](#)

Global Configuration

At this stage, the contents of installation repositories and the AutoYaST control file repository are only available in the file system of the AutoYaST server.

This section explains how to publish these repositories via the Apache 2 server for remote access.

- ♦ [“Interface Binding” on page 32](#)
- ♦ [“Apache 2 Service Activation” on page 33](#)

Interface Binding

Apache 2 listens for requests on all interfaces by default as defined by the following directive in `/etc/apache2/listen.conf`:

```
Listen 80
```

If a particular interface must be used, its IP address must be added to this directive. For example:

```
Listen 10.10.10.221:80
```

Apache 2 Service Activation

To start Apache 2 automatically after a server reboot or shutdown, `chkconfig` must be executed to create the standard runlevel entries:

```
chkconfig apache2 on
```

Directory Configuration

In the next step, the directories whose content is to be published for remote access must be configured. Apache 2 has a default directive that ensures that all files located in the `/etc/apache2/conf.d` directory that have the `conf` suffix are treated as additional configuration directives and will be read in when Apache 2 is started.

With this in mind, you need to create a file, such as `inst_server.conf`, in `/etc/apache2/conf.d` to provide all configuration directives for installation repositories and control file directories.

The configuration part responsible for the directory configuration is displayed below:

```
<Directorymatch "/data/autoyast|install|isos">
    Options +Indexes
    IndexOptions +NameWidth=*

    Order allow,deny
    Allow from all
</Directorymatch>
```

`Directorymatch` directives allow you to specify a regular expression for directory configurations. By doing this, access to multiple directories can be configured with one configuration statement achieving the following:

- ♦ The `/data/autoyast`, `/data/install`, and `/data/isos` directories are made accessible via HTTP.

In a strict sense, the `isos` directory is not required to be remotely accessible. However, administrators sometimes need to download the images themselves, so this directory is also allowed to be accessed via HTTP.

- ♦ Access is allowed from everywhere
- ♦ The content of these directories is displayed if no index document exists

The `Directorymatch` directive does not support symbolic links. For more information, see the [ASF Bugzilla Web site \(https://issues.apache.org/bugzilla/show_bug.cgi?id=53581\)](https://issues.apache.org/bugzilla/show_bug.cgi?id=53581).

If you need support for symbolic links in any of your repository directories, you need to configure it with an ordinary `Directory` directive. For example:

```
<Directory "/data/install">
    Options +Indexes +FollowSymLinks
    IndexOptions +NameWidth=*

    Order allow,deny
    Allow from all
</Directory>
```

Alias Configuration

With this configuration, the `/data/install/x86_64/sles11/sp2/cd1` URL needs to be specified to access the installation source for SLES 11 SP2. For example:

```
http://<IP-Address|DNS Name>/data/install/x86_64/sles11/sp2/cd1
```

URLs to installation repositories or control files must be specified in different places. If the URL is complicated, there is a technique to overcome this possible error source.

Apache 2 provides an alias mechanism to shorten a path specification and to make access to deeply nested file structures easier. The alias directive has the following syntax:

```
Alias URL-path file-path|directory-path
```

Novell Consulting has defined the following standard for creating aliases in `/etc/apache2/conf.d/inst_server.conf`:

- ♦ Aliases must contain one path component only (a single '/')
 - ♦ Aliases for installation repositories are made up of the following components:
 - ♦ product
 - ♦ version
 - ♦ service pack (GA, SP1, and so forth)
The shortcut `ga` is used in place of a service pack if the corresponding installation repository is the initial release of a product
 - ♦ Architecture (i586|x86_64|biarch) separated by an underline character from the preceding part ('_')
 - ♦ The alias for the Novell Consulting Installation Framework base directory is `/autoyast`
 - ♦ The alias to specify the main directory of the control file repository is `/xml`

Applying these rules results in the following alias configuration for the installation repository for SLES 11 Service Pack 2, 64-bit:

```
Alias /sles11sp2_x86_64 /data/install/x86_64/sles11/sp2/cd1
```

The following table summarizes alias definitions for the AutoYaST base directory, installation repositories, the control file repository, and the directory that contains ISO images.

Table 5-4 Main Alias Configuration Directives

Alias	Path
<code>/autoyast</code>	<code>/data/autoyast</code>
<code>/isos</code>	<code>/data/install/isos</code>
<code>/oes11sp1_x86_64</code>	<code>/data/install/x86_64/oes11/sp1/cd1</code>
<code>/sles11sp2_x86_64</code>	<code>/data/install/x86_64/sles11/sp2/cd1</code>
<code>/xml</code>	<code>/data/autoyast/xml</code>

Validating the Apache Configuration for the Repository

After you have completed your Apache configuration, ensure that you validate it by issuing the following command before restarting your Apache server:

```
apache2ctl -t
```

Then restart the Apache daemon gracefully:

```
rcapache2 graceful
```

5.4.4 Creating the Customized Boot Image

The main intention of a customized boot image is to easily provide all required boot parameters to invoke unattended installations from remote repositories for different installation variants that fit into a specific customer environment.

The following table lists the most important parameters that can be specified at the boot prompt to manage AutoYaST:

Table 5-5 *Boot Parameters*

Parameter	Value and Meaning
kernel (mandatory)	Path to the kernel image of the target system on the boot medium, such as <code>kernel /kernel/sles11sp2/x86_64/linux</code>
domain (optional)	Domain search path for DNS
nameserver (optional)	Comma-separated list of DNS servers; required if you want to use DNS names instead of IP addresses
netsetup (mandatory)	Invokes initialization of network components in the pre-installation system to gain network access to repositories and special metadata files. A value of 1 ensures that dialog windows open to specify the IP address, net mask, gateway, and name server.
netmask gateway nameserver	Corresponding values can be predefined at the boot prompt. Dialog windows contain these values as the default. All three parameters are optional.
netwait	Delay after activating the network interface in seconds
autoyast (mandatory for AutoYaST framework)	URL to AutoYaST control file or directory. For example: <code>http://<IP-Address>/xml/</code>
install (mandatory)	URL to the installation repository. For example: <code>http://<IP-Address>/sles11sp2_x86_64</code>

- ♦ [“Requirements” on page 36](#)
- ♦ [“Directory Structure of the Boot Image” on page 36](#)
- ♦ [“LST Files” on page 37](#)
- ♦ [“Image Creation” on page 38](#)

Requirements

For each target platform, the corresponding kernel and initrd are required on the boot medium. Typically, they must be copied from the `/boot/<architecture>/loader` directory on the original boot media shipped by Novell/SUSE. The kernel file is named `linux`.

Other pieces needed are the data structures to configure the graphical boot menu and to predefine the boot parameters described above.

Finally the boot loader (GRUB) binaries that are necessary to build a bootable image must be present on the boot medium.

Novell Consulting recommends the configuration and creation of boot media at a workstation that already contains all needed software parts or at the AutoYaST server itself.

Directory Structure of the Boot Image

A boot medium can be built below a working directory that represents the root of the image.

Novell Consulting suggests the following main directory structure for the boot medium:

- ♦ `/boot` – Contains the initial GRUB configuration file, GRUB stage files, and the message file that is responsible for the graphical aspects of the boot media (typically also copied from a SUSE installation medium)
- ♦ `/grub` – Contains nested configuration files for GRUB menus
- ♦ `/kernel` – Contains kernel and initrd images for all platforms that need to be installed from this medium

The following table displays a complete directory structure of a boot medium as an example for a certain configuration:

Table 5-6 Directory Structure of the AutoYaST Boot Medium

Directory	Content
<code>/boot/grub</code>	<code>stage1</code> , <code>stage2</code> , <code>iso9660_stage1_5</code> , <code>message</code> (always copy from a current system). <code>menu.lst</code> contains the first level of the nested menus; see Figure 4-1 on page 19 .
<code>/grub/main</code>	<code>message</code> (customized settings and background, optional). <code>instserver.lst</code> contains the second level of the nested menus; see Figure 4-2 on page 20 .
<code>/grub/sles10-sp4</code>	<code>sles10-sp4.lst</code> GRUB configuration file that contains predefined parameters to invoke an AutoYaST installation for SLES 10 SP4-based products.
<code>/grub/sles11-sp2</code>	<code>sles11-sp2.lst</code> GRUB configuration file that contains predefined parameters to invoke an AutoYaST installation for SLES11 SP2-based products.
<code>/kernel/sles10-sp4/x86_64</code>	<code>initrd</code> and <code>kernel</code> of the original installation medium, 64-bit, SLES 10 SP4.

Directory	Content
/kernel/sles11-sp2/x86_64	initrd and kernel of the original installation medium, 64-bit, SLES 11 SP2.

LST Files

GRUB by default searches its configuration directives in the path `/boot/grub/menu.lst` on the boot medium. An example of the `menu.lst` file is shown below:

```
# autoinst level 0
color white/blue black/light-gray
default 0
timeout 15
root (cd)
gfxmenu (cd)/grub/main/message

### localboot
title localboot
    rootnoverify (hd0)
    chainloader +1

### network sles10-sp4 64bit
title network installation sles10-sp4 64bit
    configfile /grub/sles10-sp4/sles10-sp4.lst

### network sles11-sp2 64bit
title network installation sles11-sp2 64bit
    configfile /grub/sles11-sp2/sles11-sp2.lst
```

Lines beginning with a `#` sign are comments.

The above configuration forces GRUB to use the first boot entry when no user interaction takes place for 15 seconds (timeout 15, default 0).

The `gfxmenu` directive controls which file provides the graphical background image and user customized setting, such as the language or keyboard mapping stored in `/boot/grub` by default.

You should store a customized version of the file in the `/grub/main` directory to protect against accidentally overwriting it when updating the GRUB files to a newer version.

By using the `configfile` parameter, GRUB can be directed to a second configuration file, which can contain another `configfile` parameter, and so forth. This technique is used to implement nested menus.

If the second menu entry, `network installation sles11-sp2 64bit`, is chosen, GRUB evaluates configuration directives specified in the `/grub/sles11-sp2/sles11-sp2.lst` file on the ISO image.

The content of `sles11-sp2.lst` is shown below. This example shows required boot parameters predefined within a GRUB configuration file of a customized boot image.

```
# autoinst level 2
color white/blue black/light-gray
default 0
timeout 1
root (cd)
gfxmenu /grub/main/message

### sles11 SP2 x86_64
title sles11-sp2 x86_64
    kernel /kernel/sles11-sp2/x86_64/linux gateway=10.10.10.1 netsetup=1 ...
    ... autoyast=http://10.10.10.221/xml/...
    ... install=http://10.10.10.221/sles11sp2_x86_64/
    initrd=/kernel/sles11-SP2/x86_64/initrd

### back
title back
    configfile /boot/grub/menu.lst
```

IMPORTANT: If you use this example as a model, ensure that you type the kernel directive on a single line.

You can see that the `configfile` parameter is also used to implement the Back functionality.

Image Creation

Finally, when kernel and `initrd`, boot binaries, and GRUB configuration files are prepared, the boot image must be created. This is achieved by executing `mkisofs` with the following parameters:

```
mkisofs -f -rV "AUTOYAST-BOOT-IMAGE" -b boot/grub/iso9660_stage1_5 -no-emul-boot ?
-boot-load-size 4 -boot-info-table -D -o autoyast-boot.iso
<path_to_working_directory>
```

The size of the resulting `autoyast-boot.iso` image file that will be stored in the path provided as the last parameter of the command depends on the number of platforms whose installation should be supported by the boot image. The corresponding kernels and `initrds` are consuming the most space, but it is still far smaller than the size of the original boot media.

6 AutoYaST Extended: The Config File Approach

AutoYaST control files created by YaST modules always contain values from the system where the file has been created. As a consequence, every such control file can only be used to install the same server again.

Installation of a different server implies that a new XML file with the specific information for this new server needs to be provided. This could be achieved by copying the control file of an existing server and making the required adjustments. However, this would require a highly skilled administrator and would also result in a lot of additional effort before a new server could be installed. As a result, the advantage of an automated installation would be greatly reduced by such a cumbersome configuration approach.

To overcome this limitation Novell Consulting has developed the Novell Consulting Installation Framework, which almost completely automates the creation of the AutoYaST control file and allows for fast, standardized, unattended installations with minimal administrative intervention.

This approach uses standard AutoYaST processes explained in [Section 4.5, “AutoYaST Control File,” on page 21](#), as well as some additional AutoYaST features described in this section.

A set of text files is used to store information about the environment and the specific server and a shell scripting framework connects all these pieces together.

- ♦ [Section 6.1, “Design,” on page 39](#)
- ♦ [Section 6.2, “Implementation,” on page 41](#)

6.1 Design

After evaluating numerous automated OES server installations over the past years, Novell Consulting came to the following conclusions:

- ♦ AutoYaST control files contain properties that are identical for all customer environments, such as meta XML information, system user settings, and system runlevel.
- ♦ There are always properties that differ between customer environments, and that might differ between locations or sites of a particular customer, but are the same for all servers in that particular environment, such as time settings, name resolution (DNS, SLP, hosts), LDAP server, and replica server.
- ♦ Control files typically contain server-specific information such as the host name, IP addresses, network masks, MAC addresses, and PCI IDs.
- ♦ [Section 6.1.1, “Design Principles,” on page 40](#)
- ♦ [Section 6.1.2, “Design Prerequisites,” on page 40](#)

6.1.1 Design Principles

Novell Consulting decided that the installation framework must be based on the following design principles:

- ♦ The installation is based on XML class files.
- ♦ The configuration of SLES and OES services is split into individual XML files (snippets) that can be combined into service types as needed.
- ♦ All dynamic information is removed from the XML files and replaced by placeholder variables. These modified XML files containing placeholders are referred to as template files.
- ♦ Information relevant for multiple systems is stored in a set of configuration text files supporting a similar set of variables to allow overwriting general configuration information with deviating information specific for a smaller subset of systems.
- ♦ Server-specific information is stored in a semicolon-separated server configuration file.
- ♦ The system being installed retrieves the required template files to its `/tmp/profile` directory and merges them into a complete template control file.
- ♦ The system being installed also retrieves the required configuration files to its `/tmp/profile` directory and replaces all placeholders with the actual values, resulting in a server-specific `autoinst.xml` file.
- ♦ The framework is implemented in a main library that is developed and maintained by Novell Consulting.
- ♦ The framework must fit into every customer environment without code modification.
- ♦ Customer-specific requirements can be implemented in a custom library that is already integrated in the framework. It can use every function of the main library.
- ♦ A custom server configuration file for additional server-specific information processed by the custom library is also supported.

6.1.2 Design Prerequisites

This design approach has two fundamental requirements: the ability to execute scripts on the system being installed and the ability to modify the original `/tmp/profile/autoinst.xml` control file before the actual installation starts.

- ♦ [“Pre-Script Stage” on page 40](#)
- ♦ [“autoinst.xml, modified.xml, and pre-autoinst.xml” on page 40](#)

Pre-Script Stage

The first requirement is met by AutoYaST's pre-script stage at the beginning of an installation within the initial RAM disk. In this stage, shell and Perl scripts specified in the control file are executed. In addition, network access for retrieving external files is possible.

autoinst.xml, modified.xml, and pre-autoinst.xml

After the pre-script stage, AutoYaST checks for a file named `/tmp/profile/modified.xml`. If this file exists, `autoinst.xml` is renamed to `pre-autoinst.xml` and `modified.xml` is renamed to `autoinst.xml`.

The installation then starts and processes the information in this new control file. If that file does not exist, the original control file is used instead.

6.2 Implementation

This section gives details of the implementation of the Novell Consulting Installation Framework based on the design principles outlined previously.

- [Section 6.2.1, “Overview,” on page 41](#)
- [Section 6.2.2, “Directory Structure,” on page 42](#)
- [Section 6.2.3, “The Default File,” on page 44](#)
- [Section 6.2.4, “pre-fetch.sh,” on page 45](#)
- [Section 6.2.5, “Libraries,” on page 47](#)
- [Section 6.2.6, “Configuration Files,” on page 47](#)
- [Section 6.2.7, “XML Snippets,” on page 51](#)
- [Section 6.2.8, “Post-Installation Scripts,” on page 53](#)

6.2.1 Overview

The `autoyast` parameter at the boot prompt of the installation boot medium points to the URL `http://<AutoYaST-Server>/xml/` and therefore finds and retrieves the initial default control file, which contains class statements to configure general aspects of the system.

The specified classes contain template variables for their dynamic content. These classes are retrieved from the AutoYaST server and merged by standard AutoYaST functionality into the initial control file, `/tmp/profile/autoinst.xml`. This merged file still contains the template variables.

AutoYaST then enters the pre-script stage and retrieves the `pre-fetch.sh` script via HTTP as defined in the `scripts` section of the default control file.

`pre-fetch.sh` is the command center of the Novell Consulting Installation Framework that manages all aspects of building the customized control file for the new server.

It first retrieves and sources the `CUSTOMER.txt` customer configuration file and the main library. From this point on, `pre-fetch.sh` uses functions from the main `ay_lib.sh` library.

It then locates and parses the entry for the server being installed in the `server.txt` server configuration file.

In the next step, the original control file, `/tmp/profile/autoinst.xml`, is copied to `/tmp/profile/modified.xml`. All subsequent modifications of the control file happen on this copy.

If the new server requires a tree name configuration file or a location configuration file, they are retrieved and sourced next.

Additional template files are retrieved based on the information obtained from the configuration files and are merged into the control file. Among other things, these files determine the disk partitioning of the new server and which patterns and packages need to be installed.

The service type specified in Field 14 of the server configuration file is evaluated next and the required XML files are also merged into the control file. These files contain information about the configuration of the various OES services such as eDirectory, NSS, and CIFS, but they are also used to specify some SLES configuration information, such as the memory reservation for the `kdump` kernel.

The custom library is retrieved and sourced next. As the last execution step, `pre-fetch.sh` replaces all placeholders that originated from the different template files and XML snippets that are now part of the `/tmp/profile/modified.xml` control file with the values that have been derived from the various configuration files.

If an error occurs during this process, `pre-fetch.sh` prints an appropriate error message indicating where the error has occurred. The administrator can inspect the error message and decide whether to abort or to continue the installation.

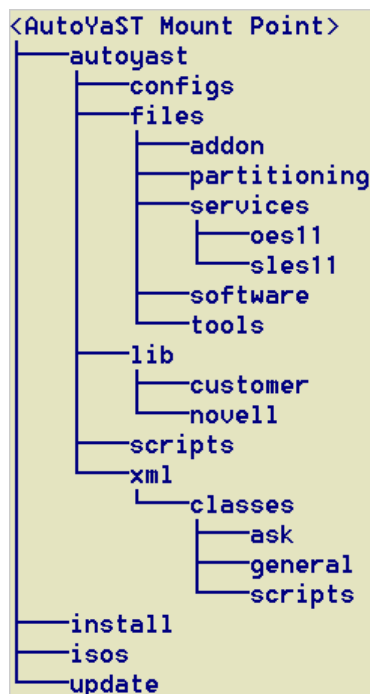
When `pre-fetch.sh` terminates without error, AutoYaST recognizes the `/tmp/profiles/modified.xml` control file and processes it as explained in “[autoinst.xml, modified.xml, and pre-autoinst.xml](#)” on page 40.

The remainder of this section explains this process in detail.

6.2.2 Directory Structure

[Figure 6-1](#) illustrates the Novell Consulting Installation Framework stored in the directory structure underneath `.../autoyast`.

Figure 6-1 Directory Structure



`.../autoyast/xml` is the control file repository directory introduced and explained in “[Control File Repository](#)” on page 24.

The `.../isos` and `.../install` subdirectories are used to store the installation sources as discussed in “[Directory Layout for the Installation Repositories](#)” on page 29.

The `.../update` subdirectory holds the signed YUM repositories containing updates for the operating system (SLES 11 SP2) and the add-on product (OES 11 SP1) being installed.

These YUM repositories are integrated into the installation process as additional add-on products.

This is a temporary workaround that is required because ZENworks Configuration Management cannot sign the YUM repositories it creates, and unsigned repositories cause warning messages whenever they are accessed in YaST.

The purpose of the other directories underneath `.../autoyast` is explained in the following table, using the default names for files and directories suggested by Novell Consulting.

Table 6-1 Directory Layout for the Novell Consulting Installation Framework

Directory	Description
<code>.../configs</code>	<p>All configuration files are stored in the following locations:</p> <ul style="list-style-type: none"> ◆ <code>AY_MAIN.txt</code> ◆ <code>CUSTOMER.txt</code> ◆ Your <code><TreeName>.txt</code> files ◆ Your location files ◆ <code>server.txt</code>
<code>.../files/addon</code>	Holds the XML class files defining the add-on products for the various server types used in Field 02 of <code>server.txt</code> .
<code>.../files/partitioning</code>	Holds the XML class files for the partitioning of the various Disk Types used in Field 07 of <code>server.txt</code> .
<code>.../files/services/oes11</code>	Holds the XML class files configuring the OES 11 services that are part of the service types defined in <code>CUSTOMER.txt</code> .
<code>.../files/services/sles11</code>	Holds the XML class files configuring the SLES 11 services that are part of the service types defined in <code>CUSTOMER.txt</code> .
<code>.../files/software</code>	Holds the XML class files for the different software selections used in Field 08 of <code>server.txt</code> .
<code>.../files/tools</code>	Stores the <code>check_errors.xml</code> ask file that is required for error handling.
<code>.../lib/customer</code>	Stores the custom library <code>customer_lib.sh</code> that can be used to implement customer-specific functions to further customize the installation process.
<code>.../lib/novell</code>	<p>Stores the main library <code>ay_lib.sh</code> provided by Novell Consulting.</p> <p>Most of the functionality of the Novell Consulting Installation Framework is implemented in this library. It must not be changed or modified in any way.</p> <p>Use the customer library instead.</p>
<code>.../scripts</code>	All scripts that are executed in the pre-, post- or init- phases of the installation process are stored in this directory.

6.2.3 The Default File

The full default file used by the Novell Consulting Installation Framework is displayed below:

```
<?xml version="1.0"?>
<!DOCTYPE profile>
<profile xmlns="http://www.suse.com/1.0/yast2ns" xmlns:config="http://
www.suse.com/1.0/configns">
  <classes config:type="list">
    <class>
      <class_name>general</class_name>
      <configuration>bootloader.xml</configuration>
    </class>
    <class>
      <class_name>general</class_name>
      <configuration>ca_mgm.xml</configuration>
    </class>
    <class>
      <class_name>general</class_name>
      <configuration>system.xml</configuration>
    </class>
    <class>
      <class_name>general</class_name>
      <configuration>general.xml</configuration>
    </class>
    <class>
      <class_name>scripts</class_name>
      <configuration>scripts.xml</configuration>
    </class>
    <class>
      <class_name>general</class_name>
      <configuration>net.xml</configuration>
    </class>
  </classes>
  <scripts>
<!--
    This section is mandatory and must not be removed.
    Amongst other things it is used to determine the
    IP address of the AutoYaST server in the post-inst
    phase
-->
    <pre-scripts config:type="list">
      <script>
        <interpreter>shell</interpreter>
        <filename>pre-fetch.sh</filename>
        <location>http://10.0.4.221/bs/scripts/pre-fetch.sh</location>
        <feedback config:type="boolean">false</feedback>
        <debug config:type="boolean">false</debug>
      </script>
    </pre-scripts>
  </scripts>
</profile>
```

This file contains two different kinds of information. First, class directives define class files that need to be merged to create the initial `/tmp/profile/autoinst.xml` control file and that configure the following aspects of the system:

- ♦ `bootloader.xml` contains boot loader specific settings.
- ♦ `ca_mgm.xml` configures the local certificate authority (CA).
- ♦ `system.xml` configures settings for system users (root), and the default runlevel and sysconfig entries.
- ♦ `general.xml` contains settings for language, time zone, keytable, and local firewall configurations.

- ♦ `scripts.xml` is a class file that specifies scripts that are executed in different installation stages (mainly in the post-installation stage).

Scripts are used to configure system properties that cannot be handled by YaST, because no XML tags exist to configure these system properties.

For example, Novell Consulting is using an init-script to install the ZCM agent and to register the new device at the ZCM server.

- ♦ `net.xml` configures network-specific parameters.

The second type of information in the file is in the scripts section, which defines the `pre-fetch.sh` pre-script file located in the `scripts` directory on the AutoYaST server. This information causes the AutoYaST engine to retrieve and execute this script.

This section is one of the few places where customer-specific values must be hard-coded. The IP address of the AutoYaST server is unique to every customer environment and cannot be derived from environment parameters. Therefore it must be explicitly specified in the URL to the `pre-fetch.sh` script in the location directive of the pre-script definition in the default file.

6.2.4 pre-fetch.sh

The first action of this script is to retrieve the `AY_MAIN.txt` system configuration file. This is implemented in the `get_main_config_files()` function that first derives the address of the AutoYaST server to be used from the command line.

The script needs three other pieces of information to be able to retrieve the required files from the installation server:

- ♦ The name of the top-level directory of the installation framework (`PREFIX=autoyast`)
- ♦ The directory where the configuration files are stored (`AY_CONFIG_DIR=configs`)
- ♦ The name of the system configuration file (`MAIN_CONFIG_FILE=AY_MAIN.txt`)

This information must be available from within the script, so it is hard-coded. If you want to use different names for this file or any of the two directories, ensure that you set the correct values here.

All other file names, directory names, and URLs required to retrieve files from the AutoYaST server are always derived from the system configuration file.

The same function also retrieves and sources the `ay_lib.sh` main library (a set of shell functions that can be reused to reduce code fragments) and the `CUSTOMER.txt` customer configuration file (or whatever name is defined for it in `AY_MAIN.txt`).

Certain configuration information such as the default gateway, a list of name servers, or a suffix search list, and also OES-specific information such as the name of the installation user or the IP address of an existing replica server can be specified in three different places:

- ♦ The customer configuration file (default name `CUSTOMER.txt`; mandatory)
- ♦ The eDirectory tree configuration file (must be named `<TreeName>.txt`; mandatory)
- ♦ A location configuration file (can use any name; optional)

The configuration files that need to be incorporated into the installation of a particular server are defined in Field 10 and Field 13 of the `server.txt` server configuration file. `pre-fetch.sh` sources these files in the following sequence:

```
customer configuration file > eDirectory tree configuration file > location configuration file
```

Values defined in global configuration files are overwritten by values for the same variable from more specific configuration files. For example, a list of NTP time servers specified in the customer configuration file as a company-wide default can be overwritten by specifying a different set of NTP servers in a configuration file representing an eDirectory tree. This NTP server list can be further modified by specifying yet another list of NTP servers in a location configuration file representing a particular site.

In some cases, a customer might have a requirement that cannot be accommodated by the `ay_lib.sh` main library. For these cases, the `customer_lib.sh` file has been provided. Code contained in this file is evaluated after code from the main library but before the replacement of the template variables takes place.

One of the central functions executed by `pre-fetch.sh` is the replacement of the placeholders with their real values. The following XML snippet illustrates how placeholders are specified and used:

```
<dns>
  <dhcp_hostname config:type="boolean">false</dhcp_hostname>
  <dhcp_resolv config:type="boolean">true</dhcp_resolv>
  <hostname>%%HOST_NAME%%</hostname>
  <domain>%%DOMAIN%%</domain>
  <searchlist config:type="list">
    <search>%%SEARCH_LIST%%</search>
  </searchlist><nameservers config:type="list">
    <nameserver>%%NAMESERVER%%</nameserver>
  </nameservers>
</dns>
```

At some point after parsing the server-specific line in `server.txt` and processing the specified configuration files, every variable will have its final value assigned. The replacement process now searches for every placeholder (enclosed in %%-signs, such as `%%HOST_NAME%%`) within `/tmp/profile/modified.xml` and replaces the whole string with the value of the corresponding `$HOST_NAME` variable. The replacement itself is using a `sed` function.

Another important function ensures the merging of complete XML snippets into the final control file. The function is using XSLT techniques by utilizing the `/usr/share/autoinstall/xslt/merge.xslt` file, which is part of any installation RAM disk provided by SUSE.

In particular, the partitioning, software, and service XML files are merged completely into the final `/tmp/profile/modified.xml` control file.

The last action performed by `pre-fetch.sh` is a basic error check. First, it checks to see if the `/tmp/profile/modified.xml` control file still contains %%-signs (which implies that “%%” must exclusively be used to identify placeholders. Any other usage, even in comments, results in an error).

Then it checks for a `/tmp/profile/errors.txt` file. This file is created by functions from `ay_lib.sh` when the retrieval of a remote file fails. If this file exists, a pop-up is generated that displays the content of `/tmp/profile/errors.txt` and queries whether the installation process should be continued or aborted.

6.2.5 Libraries

The framework provides two libraries. The main `ay_lib.sh` library developed by Novell Consulting contains nearly all code used by the `pre-fetch.sh` script. This library must not be changed or modified in any way. Any potential update to the Novell Consulting Installation Framework almost certainly results in a newer version of this library that overwrites any customer-specific modifications.

To accommodate customer-specific functionality, the empty `.../autoyast/lib/ customer/ customer_lib.sh` library file is available. It can use functions that are already implemented in the main library, but also can implement new functions to provide additional functionality. The content of this file is sourced and executed by `pre-fetch.sh` just before the replacement of the placeholders takes place. No update to the framework ever touches this particular library.

6.2.6 Configuration Files

Configuration files have been mentioned several times in preceding sections. The purpose of these files, stored in the directory defined by the `AY_CONFIG_DIR` variable in `pre-fetch.sh`, is to separate information that is applicable to multiple servers from server-specific information.

Name servers, time servers, the distinguished name of the UNIX configuration object, or the list of LDAP servers are examples of settings that can be configured.

Configuration file entries provide three pieces of information:

- ♦ The type of the information

This can be a string, an IP address or DNS name, a URL, a file or directory name, a list of values, or any custom definition.
- ♦ A short description
- ♦ The actual definition in the format `VARIABLE="value"`

The following example defines the name of the admin group for Linux User Management:

```
## Type String
## Description: fully distinguished name of the LUM admingroup
##               in LDAP syntax
LUM_ADMIN_GROUP="cn=admingroup,ou=Cluster1,ou=Servers,ou=DUS,ou=EMEA,o=Novell"
```

Although it is not always required in a strictly technical sense, we recommend that you always include values in double quotes. There is one notable exception: Because the password for the root user typically contains "\$", it must be enclosed in single quotes. As a best practice, any setting that is not used in a particular configuration file should be commented.

- ♦ [“AY_MAIN.txt System Configuration File” on page 47](#)
- ♦ [“Environment Configuration Files” on page 48](#)
- ♦ [“Server Configuration File \(server.txt\)” on page 49](#)

AY_MAIN.txt System Configuration File

This is the system configuration file of the Novell Consulting Installation Framework. It is the only configuration file where file and directory names are defined.

IMPORTANT: As explained in [Section 6.2.4, “pre-fetch.sh,” on page 45](#), the name and path of this file must be hard-coded in `pre-fetch.sh`. If you need to use a different name or path for the system configuration file, do not forget to make the appropriate changes in the script.

The framework has been tested only with the default names used in this document and defined in the version of `AY_MAIN.txt` provided by Novell Consulting. If you need to use different names, you can do so at your own risk, making the appropriate changes in `AY_MAIN.txt`.

However, we strongly recommend that you use the default name wherever possible.

Environment Configuration Files

To allow for sufficient flexibility to accommodate a wide variety of environments, the framework supports up to three configuration files where the same set of variables can be defined. They are discussed below in a global-to-local order.

- ♦ [“Customer Configuration File \(CUSTOMER.txt\)” on page 48](#)
- ♦ [“Tree Name Configuration File \(<TREE_NAME>.txt \)” on page 48](#)
- ♦ [“Location Configuration File” on page 48](#)

Customer Configuration File (CUSTOMER.txt)

This file is the most global configuration file. It is mandatory for pure SLES systems as well as for OES systems. Some information is found only here, such as the customer name (informational only) or the information on the system used for software updates. It is also the only place where service types are configured.

Other parameters, such as `GATEWAY`, `REPLICA_SERVER`, `OES_INSTALL_USER`, or the `LUM_ADMIN_GROUP` can be defined in any configuration file.

Tree Name Configuration File (<TREE_NAME>.txt)

Most customers have more than one eDirectory tree in their environment. Each of these eDirectory trees requires its own configuration file.

In small environments with only a few servers with identical settings, the `CUSTOMER.txt` configuration file can provide all required settings. However, the environment file responsible for the eDirectory tree environment must exist (the values can be left empty) if it is intended to install OES servers.

The name of this file is derived from the eDirectory tree name specified in Field 10 of the `server.txt` server configuration file and the suffix “.txt”. For example, if an eDirectory tree is named `MY-COMPANY-TREE`, then `pre-fetch.sh` searches for a `.../autoyast/configs/MY-COMPANY-TREE.txt` file (case-sensitive).

Location Configuration File

Configuration information defined in the customer configuration file or in the tree name configuration file can be overwritten by an optional location configuration file. You can choose any file name you want for this type of configuration file. This file is optional. If it is used, it must be specified in Field 13 of the server configuration file.

“Location” can represent a country, a city, a site, a building, a data center or a server room, or even just a LAN segment—basically anything that defines a group of servers that needs different configuration information.

Server Configuration File (server.txt)

The server configuration file defines the server-specific information for the Novell Consulting Installation Framework.

Each server configuration is represented by a single line, with its fields separated by semicolons. The current server file contains 14 fields explained in the following table:

Table 6-2 Field Descriptions for the server.txt Server Configuration File

No.	Field Name	Meaning	Example	Multivalue/ Required
1	HOST_NAME	Short name of the server being installed.	machine01	No Mandatory
2	IP_ADDRESS/CIDR	IP address and net mask of the server being installed; used as system identifier. Net mask in CIDR notation. For example, 255.255.240.0 = 20	10.10.10.101/24	No Mandatory
3	GATEWAY	IP address of the default gateway	10.10.10.1	No Mandatory
4	SERVER_TYPE	Pure SLES or OES systems, including version, feature pack (FP, OES 11 only) and support pack level. An <code>ovl</code> at the end of the identifier denotes that the system will be installed from the combined SLES/OES installation medium. Server types for different environments can be created by appending them with identifiers such as <code>-DEV</code> or <code>-TEST</code> . IMPORTANT: There must always be a corresponding <code>addon_products-\$SERVER_TYPE.xml</code> file in the <code>../files/addon</code> directory, even when you are installing a pure SLES system.	sles11sp2, oes11fp0sp1, oes11fp0sp1ovl	No Mandatory
5	DEVICE_NAME0	Name of the first system disk device.	/dev/sda, /dev/hda, / dev/vmx, /dev/cciss / c0d0	No Mandatory
6	DEVICE_NAME1	Name of the second system disk device.	/dev/sdb, /dev/hdb, / dev/vmx, /dev/cciss / c0d1	No Optional

No.	Field Name	Meaning	Example	Multivalue/ Required
7	PART_FILE	Partitioning class file. Must be located in .../files/ partitioning.	The file name should identify the type and size of the device for which the partitioning has been defined. For example: part-Xen-20GB.xml part-cciss-146GB.xml	No Mandatory
8	SOFT_FILE	Software class file. Must be located in. .../files/software.	The file name should identify the SERVER_TYPE and the purpose of the server for which the software selection is valid. For example: soft-oes11fp0sp1ov1- login.xml soft-oes11fp0sp1ov1- NCS.xml	No Mandatory
9	ZCM_KEY_LIST	Keys for the registration of the new device with a ZCM server. Multiple keys are possible, separated by ':' The first key is used for registration with the ZCM server,. All other keys are used for subscribing to device groups in ZENworks Configuration Management for managing configuration and software updates.	Examples for valid ZCM keys for a production environment are: Location: PROD_EDIR Config: PROD_EDIR_GRP Update: PROD_OES11FP0SP1_GRP	Yes Optional
10	TREE_NAME	eDirectory tree name. There must be a configuration file <TREE_NAME>.txt in .../configs. The file name is case sensitive.	MyTree	No Mandatory for OES servers
11	TREE_TYPE	Determines whether a new tree will be created or the server will join an existing tree.	existing new	No Mandatory for OES servers
12	SERVER_CONTEXT	Server context in LDAP syntax.	ou=servers, ou=services, o=Novell	No Mandatory for OES servers

No.	Field Name	Meaning	Example	Multivalue/ Required
13	SERVER_LOCATION	<p>Configuration file determining all aspects of the physical server location such as the following:</p> <ul style="list-style-type: none"> ♦ Default gateway ♦ LDAP server list ♦ NTP server list <p>There must be a configuration file <code><SERVER_LOCATION>.txt</code> in <code>.../configs</code>. The file name is case-sensitive.</p> <p>In smaller environments, all of this information can be provided in the configuration file corresponding to the tree name or in <code>CUSTOMER.txt</code>.</p>	<p><code>Provo.txt</code></p> <p><code>Utah.cfg</code></p> <p><code>DC1.info</code></p>	<p>No</p> <p>Optional</p>
14	SERVICE_TYPE	<p>XML profile as defined in <code>CUSTOMER.txt</code>. The file names are case-sensitive.</p> <p>The XML files referenced by the profile must exist in <code>.../files/services/oes11</code> and <code>.../files/services/sles11</code>.</p>	<p><code>OES11FP0SP1_eDir</code></p> <p><code>OES11FP0SP1_NCS</code></p>	<p>No</p> <p>Mandatory</p>

The question sometimes arises about how the installation process determines which line of the server configuration file represents the server that is being installed.

This is another task managed by `pre-fetch.sh` and processed within the `parse_line $ (get_unique_line IP)` function of the main library.

The unique key that is used to identify the correct server configuration line is the IP address of the server being installed. It is compared to the IP address part of Field 02 in the server configuration file, line by line. The first match completes the parse mechanism and the resulting line is used to configure the server.

6.2.7 XML Snippets

The files stored in the directory structure underneath `.../files` have been obtained by retrieving the relevant portion from an `autoinst.xml` file saved after a manual installation and by replacing dynamic information with placeholders.

You should carefully inspect each of these files to ensure that the various settings meet your specific requirements.

- ♦ [“Add-On Products” on page 52](#)
- ♦ [“Partitioning” on page 52](#)
- ♦ [“Software” on page 52](#)
- ♦ [“Services” on page 53](#)
- ♦ [“Tools” on page 53](#)

Add-On Products

Each server installed by the Novell Consulting Installation Framework requires one add-on XML file. This file must be named `addon_products- $\$SERVER_TYPE$.xml`, where $\$SERVER_TYPE$ is the value of Field 04 in `server.txt`.

For pure SLES systems, these files determine which patches (YUM repositories) are deployed as part of the installation process.

For OES systems, they also define the installation source for OES (single CD or combined SLES/OES DVD) as well as the OES patches included in the installation process.

These files can either use the full path to the directory where a YUM repository is stored on the installation server, or you can define additional alias directives in the Apache configuration to shorten these URLs.

Partitioning

Files in this subdirectory contain partitioning information. The sample file implements the following partitions on the first system device with a capacity of 146 GB. This could be a single device or a hardware RAID1:

Table 6-3 Partition Layout for a Server With a 146 GB Disk

Partition	Type	Volume Group	Volume	Mount Point	File System	Size (GB)
1	Linux	n/a	n/a	/boot	ext3	0.2
2	LVM	system	swap	swap	swap	4
2	LVM	system	root	/temp	ext3	5
2	LVM	system	root	/var	ext3	90
2	LVM	system	root	/	ext3	10
2	LVM	system	unused			36

More complex partition layouts can easily be developed by using this sample file as a template.

Software

The files in this subdirectory are referenced in Field 08 of `server.txt` and determine which patterns and packages are installed on the target system. The guideline for software selection should always be to install only what is required.

Carefully plan these files by following the concept “As many as needed, as few as possible.” You might want to define one software selection for dedicated eDirectory/Login/iManager servers and another one for cluster nodes providing print and file services accessed through AFP, CIFS, or NCP. Branch office servers providing additional services such as DHCP and DNS might require yet another software selection.

Services

The configuration of the software that has been installed is defined by the files in two directories.

- ♦ “Services - OES” on page 53
- ♦ “Services - SLES” on page 53

Services - OES

The `oes11fp0sp1_base.xml` file contains all of the configuration settings that are required by each OES11 SP1 server, no matter which additional software might be installed on the system, such as the LDAP servers for the OES services, the eDirectory, SLP, and NTP configuration information, or the configuration of the LUM, NCP Server, and SMS OES services. This file needs to be part of any service type definition in `CUSTOMER.txt`.

The other files in this directory configure individual OES services such as AFP, NSS, or iManager. You cannot configure Novell Cluster Services as part of the initial installation. You must do this manually after you have completed your storage configuration and verified proper storage access.

Services - SLES

Currently the only configuration for a SLES service through an XML snippet is the memory reservation for the `kdump` kernel required to obtain kernel dumps for analysis in case of a software issue.

For SLES 11, a minimum of 128 MB is recommended for servers with up to 12 GB of RAM. However, the total amount of memory required by the `kdump` kernel is also dependent on the number of storage devices assigned to a server. For more information, see [Novell Support TID 3374462](http://www.novell.com/support/kb/doc.php?id=3374462), “Configure kernel core dump capture” (<http://www.novell.com/support/kb/doc.php?id=3374462>).

As a consequence, a memory reservation of 128 MB might be considered a baseline for eDirectory servers and stand-alone servers with a small number of devices, but cluster nodes that have access to a larger number of devices require a higher memory reservation.

Tools

This directory currently only holds the `check_errors.xml` file that is part of the error checking mechanism of the `pre-fetch.sh` script.

6.2.8 Post-Installation Scripts

Post-installation scripts are necessary to perform installations that cannot be covered by AutoYaST. They must be specified within the `<scripts>` section of an AutoYaST control file. The framework uses a separate class-file (`../autoyast/xml/classes/scripts/ scripts.xml`) that specifies where these scripts are retrieved from and execute. For example:

```

<?xml version="1.0"?>
<!DOCTYPE profile>
<profile xmlns="http://www.suse.com/1.0/yast2ns"
xmlns:config="http://www.suse.com/1.0/configns">
  <scripts>
    <!-- init-scripts (during the first boot of the installed system,
                      all services up and running) -->
    <init-scripts config:type="list">
      <listentry>
        <!-- Miscellaneous changes -->
        <filename>post-inst.sh</filename>
        <interpreter>shell</interpreter>
        <location>%%AY_SERVER%%/%%PREFIX%%/scripts/post-inst.sh</location>
      </listentry>
      <listentry>
        <!-- Install ZCM agent -->
        <filename>zcm-install.sh</filename>
        <interpreter>shell</interpreter>
        <location>%%AY_SERVER%%/%%PREFIX%%/scripts/zcm-install.sh</location>
      </listentry>
    </init-scripts>
  </scripts>
</profile>

```

The scripts are executed in the init stage as specified by the `<init-scripts>` tag. The initial release of the Novell Consulting Installation Framework uses two post-installation scripts:

- [“The post-inst.sh Script” on page 54](#)
- [“The zcm-install.sh Script” on page 54](#)

The post-inst.sh Script

This script is used to correct or change configuration settings that have not been configured as expected by AutoYaST. It might be obsolete in future releases. Currently, it takes care of the following configuration settings:

- Disabling IPv6
- Removing “localhost” from the IPv6 loopback entry in `/etc/hosts`
- Enabling X forwarding for SSH connections

The zcm-install.sh Script

The `zcm-install.sh` script is responsible for downloading the ZENworks Configuration Management agent binary from the source defined in `CUSTOMER.txt`, and for installing it and subscribing the new server to the ZENworks Configuration Management system by using the ZENworks Configuration Management keys specified in Field 09 of the server configuration file. This process ensures the creation of the server device object in ZENworks Configuration Management in the desired folder (using the first key specified) and the assignment to server device groups (all other keys). The latter step implements update and software assignments for this server.

As a result of the subscription process configuration, bundles from the ZENworks Configuration Management system are immediately installed on the server and the patch channels become visible after the script has been executed.

7 Miscellaneous

- ♦ Section 7.1, “SSH-Based Installation,” on page 55
- ♦ Section 7.2, “Info File,” on page 56
- ♦ Section 7.3, “Driver Updates,” on page 56
- ♦ Section 7.4, “Boot Parameter y2confirm,” on page 56
- ♦ Section 7.5, “Securely Downloading AutoYaST Control Files via HTTPS,” on page 57
- ♦ Section 7.6, “Troubleshooting and Monitoring,” on page 57
- ♦ Section 7.7, “Documentation,” on page 58

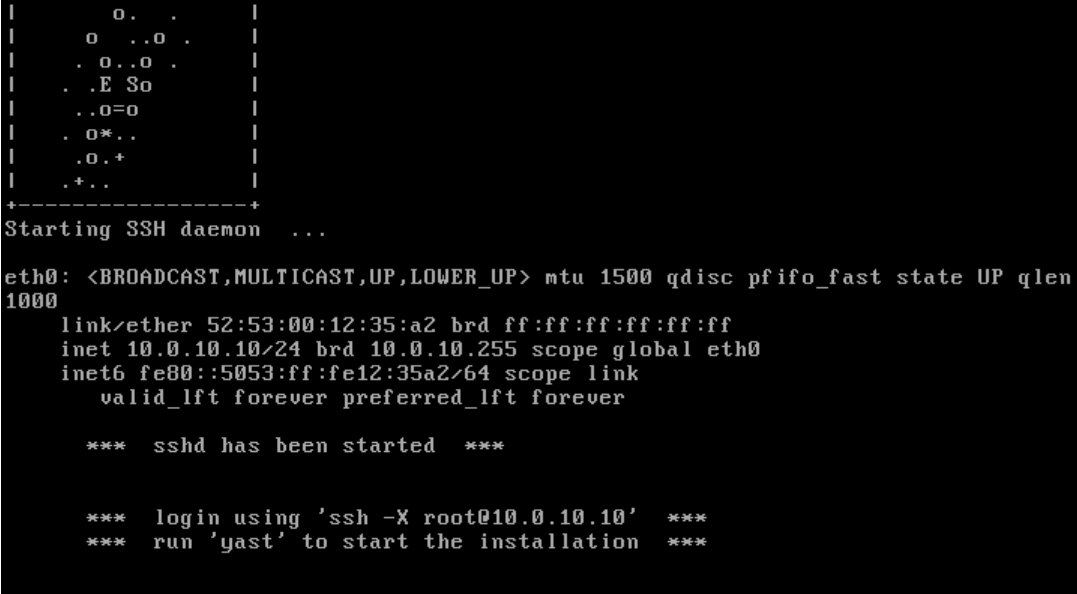
7.1 SSH-Based Installation

Sometimes the installation process gets stuck or the initialization of hardware fails because of hardware-related errors or misconfigurations. In such cases, the ability to inspect the affected system can be very useful.

This can easily be achieved by providing the boot parameter `ssh=1`, which invokes the SSH daemon on the system being installed. The system prompts for a temporary SSH password. As an alternative, this password can also be provided by the `sshpassword=` boot parameter.

When the system is fully initialized, a message with instructions on how to log in to the server and how to invoke the installation appears on the screen.

Figure 7-1 Instructions for SSH-Based Login



```
|      0. . |
|      0  .0 . |
|      . 0..0 . |
|      .E So |
|      .0=0 |
|      . 0*.. |
|      .0.+ |
|      .+.. |
+-----+
Starting SSH daemon ...

eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 52:53:00:12:35:a2 brd ff:ff:ff:ff:ff:ff
    inet 10.0.10.10/24 brd 10.0.10.255 scope global eth0
    inet6 fe80::5053:ff:fe12:35a2/64 scope link
        valid_lft forever preferred_lft forever

    ***  sshd has been started  ***

    ***  login using 'ssh -X root@10.0.10.10'  ***
    ***  run 'yast' to start the installation  ***
```

The system can now be inspected. When the installation is started, a second SSH session to the device is required to monitor the installation process or to analyze any problem. The system from which you log in must be running a X server.

Another reason to use SSH-based installations is because some remote management boards might have difficulty in properly displaying graphical screens.

7.2 Info File

The current version of GRUB does not allow you to pass more than 256 characters at the boot prompt. If many parameters must be specified, some of them can be put into a special file to reduce the number of characters. The file must be specified as kernel parameter as follows:

```
info=<Protocol://path_to_info_file>
```

Not every boot parameter can be configured via info files. For instance, it is not possible to set up network parameters if the info file itself is located on a network repository.

The following example shows an info file:

```
insecure: 1
autoupgrade: 1
dud: http://10.10.10.221/info/au_oes11/unattended_migration.dud
```

7.3 Driver Updates

In rare cases, it might be necessary to provide driver update files to the installation environment. Driver update files contain binaries or configuration files that are needed at the beginning of the installation but are not part of the current initrd. They are mostly used to support very new hardware components or to provide a missing feature via a YaST module.

A driver update can be specified as follows:

```
dud=<Protocol://path_to_dud_or_rpm>
```

A driver update can also be included in the info file as shown in [Section 7.2, “Info File,” on page 56](#).

7.4 Boot Parameter y2confirm

During the development phase of a new installation (that is, a new service type), the ability to inspect the installation proposal created by YaST before the installation is invoked can save a lot of time.

This can be achieved by specifying the y2confirm parameter at the boot prompt.

When this parameter is set, YaST stops when the installation proposal is complete and only when the proposal has been accepted, just as you need to do in a manual installation.

7.5 Securely Downloading AutoYaST Control Files via HTTPS

Until recently it was not possible to protect the control file repository against unauthorized access via HTTP. Configuring the Web server for certificate-based client authentication prevented AutoYaST from accessing this repository.

Since SLES 11 SP2, AutoYaST does support certificate-based authentication. For this purpose, a public key and private key generated by the CA that created the certificate used by the Web server must be inserted into `/etc/ssl/clientcerts` on the initrd of the boot medium. The keys must be named `client-cert.pem` for the public key and `client-key.pem` for the private key.

For more information, see [Novell Support TID 3909888](http://www.novell.com/support/kb/doc.php?id=3909888), “How to modify or customize the installation initrd” (<http://www.novell.com/support/kb/doc.php?id=3909888>).

7.6 Troubleshooting and Monitoring

- ♦ [Section 7.6.1, “Installation Server,” on page 57](#)
- ♦ [Section 7.6.2, “Server Being Installed,” on page 57](#)

7.6.1 Installation Server

Access to installation repositories and control files via HTTP or HTTPS can be monitored in the Apache 2 log files. By default, these files are:

- ♦ `/var/log/apache2/access.log`
- ♦ `/var/log/apache2/error.log`

7.6.2 Server Being Installed

- ♦ [“Installation Stage” on page 57](#)
- ♦ [“Installed System” on page 58](#)

Installation Stage

The inspection of the server being installed can be done via a remote SSH session as described in [Section 7.1, “SSH-Based Installation,” on page 55](#).

Configuration files and temporary files involved in the AutoYaST process are located at two different places:

- ♦ `/tmp/profile`

Control files, configuration files, and the libraries of the Novell Consulting Installation Framework are found here.

- ♦ `/tmp/YaST-nnnnn-xxxxxx`

In the `/tmp` directory of the server being installed, there are at least three directories with a name starting with YaST followed by five random numbers and six alphanumerical characters, such as `YaST-03200-AxfV0b`.

One of them contains a subdirectory `pre-script` that contains the script used in the pre-script stage of the installation. The same directory contains a log subdirectory that holds a log file for every pre-script.

These directories exist only during the installation stage. They are automatically deleted at the end of Stage 1.

All processes invoked by YaST are logged to `/var/log/YaST/y2log`. This file can be inspected during the installation stage as well as after the installation stage.

If the boot parameter `loghost=<IP-Address of syslog server>` is given at the boot prompt of the device being installed, all log files are redirected to the syslog server specified.

The log level can also be specified by using the `loglevel=[0-10]` boot parameter. Of course, the syslog server needs to be configured for remote logging.

Installed System

Information regarding aspects of the AutoYaST process can be found on a running system in the `/var/adm/autoinstall` directory. This directory contains the following subdirectories:

- ♦ `logs`
Contains a log file for every script that has been specified within the `<scripts>` tag.
- ♦ `scripts`
Contains the scripts themselves.
- ♦ `cache`
Contains the initial control file named `pre-autoinst.xml` and the final control file named `installedSystem.xml`.
- ♦ `files`
Contain all files provided via the `<files>` tag of the control file.

7.7 Documentation

AutoYaST-specific documentation from the developer can be found on the [SUSE home page \(http://users.suse.com/~ug/\)](http://users.suse.com/~ug/).

Using ZENworks 11 to Manage Open Enterprise Server 11 or Later



Novell Consulting has developed a methodology to configure and manage SUSE Linux Enterprise Servers (SLES) and Novell Open Enterprise (OES) servers through the ZENworks Configuration Management framework.

- ♦ [Chapter 8, “ZENworks Configuration Management Introduction,” on page 61](#)
- ♦ [Chapter 9, “Server Installation,” on page 63](#)
- ♦ [Chapter 10, “Server Configuration,” on page 79](#)
- ♦ [Chapter 11, “Managing ZENworks Configuration Management,” on page 89](#)
- ♦ [Chapter 12, “Managing Linux Bundles,” on page 155](#)
- ♦ [Chapter 13, “Agent Deployment,” on page 181](#)
- ♦ [Chapter 14, “Agent Commands,” on page 191](#)
- ♦ [Chapter 15, “Fault Diagnostics and Debugging,” on page 193](#)

8 ZENworks Configuration Management Introduction

ZENworks Configuration Management is a component of a solution portfolio that provides centralized patch management together with configuration management, imaging functions, reporting functions, and Linux server and workstation remote management.

Patch management involves obtaining patches from external sources (Novell, Red Hat Network, or others) and distributing them to servers or workstations internal to the company. To mirror the patches from Novell or Red Hat mirror servers, an appropriate registration code from the supplier must be available. In-house applications and patches can also be managed and distributed by using ZENworks Configuration Management.

Patches can either be distributed by a push process (distribution of patches predetermined by the ZCM server) or by a pull process (installation of patches controlled by the ZENworks Configuration Management agent component from the server being patched).

Imaging makes the Image and Restore functions available for partitions or whole hard drives. Imaging and Restore can follow specific policies automatically or can be performed manually. In addition to the standardized processes, scripted Image and Restore solutions can also be set up with ZENworks Configuration Management.

In addition to patch management via RPM packages, ZENworks Configuration Management has the ability to deploy configuration files or archives. Several actions can be combined with remote execution of binary files, scripts, or Java applications on the devices to be managed.

Novell Consulting has developed a methodology to configure and manage SUSE Linux Enterprise Servers (SLES) and Novell Open Enterprise (OES) servers by using the ZENworks Configuration Management framework.

Special attention has been given to questions that arise in every project situation, such as:

- ♦ How to manage different patch levels (frozen patch level)
- ♦ How to manage different staging areas such as test, development, and production
- ♦ How different or unique configuration settings can be managed across many servers
- ♦ Which mechanisms are available to remotely execute tasks on one or more devices

In ZENworks Configuration Management, a Management Zone consists of one or more Primary Servers, optional Satellite Servers, and manageable devices (Windows or Linux).

Because managing Linux servers in typical environments is easily achieved with a single Primary Server, this document does not cover ZENworks Configuration Management environments with multiple Primary Servers. More details on such designs can be found in the *System Planning, Deployment, and Best Practices Guide* (http://www.novell.com/documentation/zenworks11/zen11_cm_deployment_bp/data/index.html) and corresponding TIDs.

Nor does this document cover how to install or configure ZENworks Configuration Management on operating systems other than SLES. An overview of operating systems that are suitable as ZCM servers is found “Primary Server Requirements” (http://www.novell.com/documentation/zenworks11/zen11_installation/?page=/documentation/zenworks11/zen11_installation/data/bon1pl2.html) in the *ZENworks 11 SP2 Server Installation Guide*.

9 Server Installation

- ♦ [Section 9.1, “Prerequisites and Planning,” on page 63](#)
- ♦ [Section 9.2, “Server Installation Procedure,” on page 69](#)

9.1 Prerequisites and Planning

- ♦ [Section 9.1.1, “Operating System,” on page 63](#)
- ♦ [Section 9.1.2, “Quantity Structures,” on page 63](#)
- ♦ [Section 9.1.3, “Network,” on page 65](#)
- ♦ [Section 9.1.4, “Databases,” on page 66](#)
- ♦ [Section 9.1.5, “Virtualization,” on page 67](#)
- ♦ [Section 9.1.6, “Installation Worksheet for a Primary Server,” on page 67](#)
- ♦ [Section 9.1.7, “Installation Worksheet for the Sybase Database,” on page 68](#)

9.1.1 Operating System

When you install ZENworks Configuration Management on SUSE Linux Enterprise Server (SLES), Novell Consulting recommends that you use the newest SLES version and patch level, which is currently SLES 11 SP2. Other supported SLES versions are SLES 10 SP3/SP4 and SLES 11 SP1.

Installing on 64-bit architecture is highly recommended to gain all the advantages of that architecture.

The ZCM server must not host Novell eDirectory or a Novell Client.

9.1.2 Quantity Structures

- ♦ [“Disk Capacity” on page 63](#)
- ♦ [“Memory” on page 64](#)
- ♦ [“CPU” on page 64](#)
- ♦ [“Network Load” on page 64](#)

Disk Capacity

- ♦ [“Content Repository” on page 63](#)
- ♦ [“Partitioning” on page 64](#)

Content Repository

The disk capacity required for the content repository can vary, depending on the life span of the server and the number of distributions to be managed.

- ♦ Every patch level of any package must be retained on the server from initial release up to the present date. Over time, 10–20 different kernel versions can be managed by a ZCM server.

- ♦ As a general rule, multiple architectures (such as i586, x86_64, and ia64) must be managed per distribution, which in turn implies doubling or tripling disk space requirements
- ♦ An online catalog with all packages for the distribution media must be made available for resolution of package dependencies (2–4 GB per architecture and distribution)
- ♦ New service packs or new distribution versions should be expected every 1–2 years

For simple environments, 30–50 GB of disk space are required. For larger environments, a minimum of 100 GB should be reserved.

Partitioning

Novell Consulting recommends the use of an LVM volume group that contains LVM volumes for all system file systems except `/boot`. A capacity of 10 GB is usually sufficient for the `/` partition of a SLES server. 4 GB is sufficient for the `/swap` partition. The `/opt` directory should be partitioned separately with a capacity of at least 8 GB because 6 GB is temporarily required by the installation engine (ZEROConf).

When partitioning, the `/var/opt/novell/zenworks` directory should be located on its own LVM logical volume *zenworks* in the LVM volume group *zenworks* with a size of 100 GB. The database (if it is embedded) and content repository are stored below this directory.

The partition and file system layout recommended by Novell Consulting are summarized in the following table:

Table 9-1 Partition Layout

Partition	Type	Volume Group	Volume	Mount Point	File System	Size (GB)
1	Linux	n/a	n/a	<code>/boot</code>	ext3	0.2
2	LVM	system	swap	swap	swap	4
2	LVM	system	root	<code>/</code>	ext3	10
2	LVM	system	opt	<code>/opt</code>	ext3	8
3	LVM	zenworks	zenworks	<code>/var/opt/novell/zenworks</code>	ext3	100

Memory

The ZENworks Configuration Management components are largely based on programs written in Java and, in part, in Mono. This means that the more memory you have, the better the performance will be. A minimum of 4 GB RAM must be available. An installation on servers with less than 2 GB memory is declined. Novell Consulting recommends that you provide at least 8 GB RAM.

CPU

At a minimum, a server-class CPU such as an AMD Opteron or Intel Xeon processor is required. If the Primary Server is running on a virtual machine, you should use a dual-core processor.

Network Load

- ♦ [“WAN” on page 65](#)
- ♦ [“LAN” on page 65](#)

WAN

Depending on the number of distributions and catalogs to be managed, a lengthy initial download time can be expected. After that, only the latest patches are retrieved from the mirror server at the interval you prefer.

LAN

The amount of LAN traffic depends on the number of managed devices and how patch deployment takes place. Manual distributions of patches typically configured in small and medium-sized environments do not impact network load. Automatic patch deployments to many devices can heavily influence network load in large environments.

9.1.3 Network

- ♦ [“IP Address” on page 65](#)
- ♦ [“TCP Ports” on page 65](#)
- ♦ [“UDP Ports” on page 66](#)
- ♦ [“Name Resolution” on page 66](#)
- ♦ [“Time Synchronization” on page 66](#)
- ♦ [“Software” on page 66](#)

IP Address

The server must have a static IP address or permanent lease DHCP IP address.

TCP Ports

The following ports must be open on the server for inbound connections:

- ♦ 80 - HTTP
- ♦ 443 - HTTPS

The following ports must be open on the server for outbound connections:

- ♦ 80 - HTTP
- ♦ 443 - HTTPS
- ♦ 2645 - CASA

Opening this port allows ZENworks to manage devices outside of the firewall

- ♦ 5550 - Remote Management Listener
- ♦ 5750 - Used by the Remote Management Proxy
- ♦ 5950 - Used by the Remote Management Service
- ♦ 7628 - Used by the Adaptive Agent for quick tasks
- ♦ 8009 - Used by the Tomcat AJP connector

The following ports should be opened optionally:

- ♦ 9971 - Used by the AMT Hello Listener to discover AMT devices

UDP Ports

The following ports must be opened if the services are used:

- ♦ 67 - Used by DHCP Proxy if the server is not a DHCP server
- ♦ 69 - Used by the Imaging TFTP
- ♦ 997 - Used by the Imaging Server for multicasting
- ♦ 1761 - Used to forward subnet-oriented broadcast magic packets for Wake-on-LAN.
- ♦ 4011 - Used for Proxy DHCP
- ♦ 13331 - Used by the zmgpreboot policy

Name Resolution

Before installing ZENworks Configuration Management, the functionality of DNS and `/etc/hosts` must be checked. We highly recommend that both services are configured.

Time Synchronization

NTP must always be configured with at least one additional failover time source.

Software

When you are installing on x86_64 systems, the `pam-32bit` package must be installed because of a dependency with the CASA packages. In addition, `libgdiplus0` and `mono-core` are needed at the Primary Server.

ZENworks Configuration Management 11 SP2 can be downloaded from the [Novell Downloads Web site](http://download.novell.com/Download?buildid=yO8R08QTXEM~) (<http://download.novell.com/Download?buildid=yO8R08QTXEM~>)

9.1.4 Databases

- ♦ “Database Products” on page 66
- ♦ “External versus Internal Sybase Database Server” on page 67

Database Products

ZENworks Configuration Management supports various versions of Sybase SQL Anywhere, Oracle 11g, and Microsoft SQL Server as database back ends. More details about the supported databases can be found in “Database Requirements” (http://www.novell.com/documentation/zenworks11/zen11_installation/data/bon1pn4.html) in the *ZENworks 11 SP2 Server Installation Guide*. The decision to use a certain database to use depends on several factors:

- ♦ Number of managed devices:
 - ♦ The Sybase database delivered with ZENworks Configuration Management fulfills performance requirements for up to 1500 managed devices (Windows + Linux)
 - ♦ If more than 1500 devices must be managed, an external Oracle or MS-SQL database must be used

- ♦ Available database know-how. If in-house know-how for one of these databases is already available, it should be taken into account when choosing the database
- ♦ Existing database environments. If the company is already managing large Oracle or MS-SQL database instances, you might want to use the advantage of existing management processes such as database backup and database optimization procedures.

External versus Internal Sybase Database Server

An internal Sybase database is preferred in following environments:

- ♦ The number of manageable devices is not expected to exceed 3000 devices in the future
- ♦ Devices that must be managed are preferably Linux devices
- ♦ The ZENworks Configuration Management zone consists of only one Primary Server. No further Primary or Satellite servers are planned to be installed in future

An external Sybase database should be considered under the following conditions:

- ♦ A growing number of manageable devices are expected in the future
- ♦ Devices with Windows operating systems must be managed from the same ZENworks Configuration Management zone
- ♦ Load balancing, higher availability, and separate management is a requirement

9.1.5 Virtualization

Although virtualization of the ZENworks Configuration Management components is supported (see “Primary Server Requirements” (http://www.novell.com/documentation/zenworks11/zen11_installation/data/bon1pl2.html) in the *ZENworks 11 SP2 Server Installation Guide*, Novell Consulting recommends that you install the primary ZCM server that hosts the database on physical hardware if you plan to manage more than 500 devices. Servers hosting the management interface or acting as content servers can be running on virtualized machines.

9.1.6 Installation Worksheet for a Primary Server

The following table summarizes requirements for the installation of a Primary Server.

Component	Value
Prerequisites	<p>Software:</p> <ul style="list-style-type: none"> ♦ Mono ♦ pam-32bit ♦ CASA ♦ ISO image that contains the current ZENworks Configuration Management software <p>Hardware:</p> <ul style="list-style-type: none"> ♦ Minimum 4 GB RAM

Component	Value
Partitioning/File system configuration	<ul style="list-style-type: none"> ♦ Linux partition mounted on <code>/boot</code>, 200 MB, maximum 2 GB, ext3 ♦ LVM partition for volume group <code>zenworks</code> with volume <code>zenworks</code> mounted on <code>/var/opt/novell/zenworks</code>, 100 GB, ext3 ♦ LVM partition for volume group <code>system</code> with the following volumes: <ul style="list-style-type: none"> ♦ <code>swap</code>, <code>swap</code>, 4 GB, ext3 ♦ <code>root</code>, <code>/</code> mounted, 10 GB, ext3 ♦ <code>opt</code>, <code>/opt</code> mounted, 8GB, ext3
DNS	All ZCM back-end servers must have valid DNS entries for forward and reverse lookup.
NTP	NTP must be configured for all back-end systems.
ZENworks Configuration Management Zone Name (new zone)	The ZENworks Configuration Management Zone name is limited to 20 characters. No special characters are allowed for except dashes and underscores. Digits count as normal characters.
External Database	<ul style="list-style-type: none"> ♦ DNS name ♦ Database port (defaults – 2638 remote Sybase SQL, 1433 MS-SQL) ♦ Database name ♦ Database user ♦ Password for database user
License Key	Optional

9.1.7 Installation Worksheet for the Sybase Database

The following table summarizes requirements for the installation of an external database server.

Component	Value
Prerequisites	<p>Software:</p> <ul style="list-style-type: none"> ♦ ISO image that contains the current ZENworks Configuration Management software <p>Hardware:</p> <ul style="list-style-type: none"> ♦ Minimum 4 GB RAM
Database Port	2638. Communication on the database port must be allowed between the Primary Server and the database server.
WAN consideration	Primary Servers and the ZENworks database must reside on the same network segment.
Default Character Set	For Sybase, the UTF-8 character set is required.

Component	Value
Database properties	<p>The following properties must be configured during database setup:</p> <ul style="list-style-type: none"> ♦ Database name ♦ Database administrator ♦ Database administrator password <p>Make note of these values, because they are needed during installation of the Primary Server.</p>

9.2 Server Installation Procedure

After downloading the `ZENworks11SP2.iso`, the MD5 checksum should be computed and compared with the MD5 checksum on the Novell download site.

The image is then loop mounted. For example:

```
mount -o loop ZENworks11SP2.iso /mnt
```

The installation program must be executed as user `root` and an X server should be running for GUI-based installations.

On Linux platforms, the administrator can choose among different installation methods:

- ♦ GUI installation

```
sh /<path_to_mountpoint>/setup.sh
```

For example:

```
sh /mnt/setup.sh
```

- ♦ Installation with an external database

```
sh /<path_to_mountpoint>/setup.sh -o
```

- ♦ Command line installation

```
sh /<path_to_mountpoint>/setup.sh -e
```

The following section explains the standard installation method via the Java GUI.

9.2.1 ZCM Primary Server Installation in GUI Mode

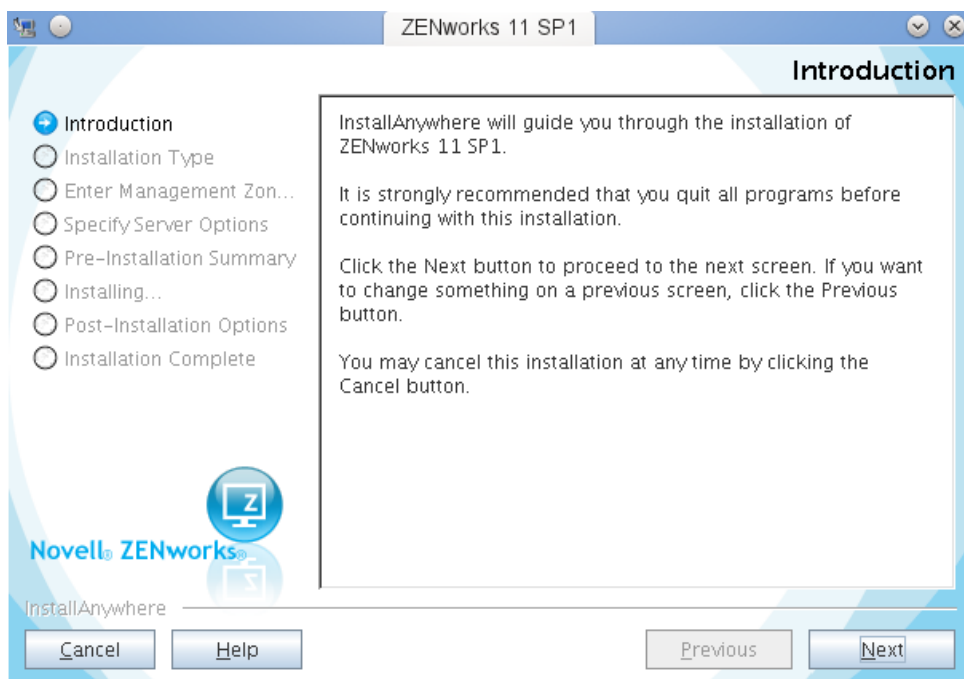
The following screen shot shows the first installation screen. The language settings are automatically determined. Novell Consulting recommends that you always use English as the installation language. To achieve this in a command line installation, execute the `unset LANG` command at the shell prompt.

Figure 9-1 ZCM Primary Server Installation (1)



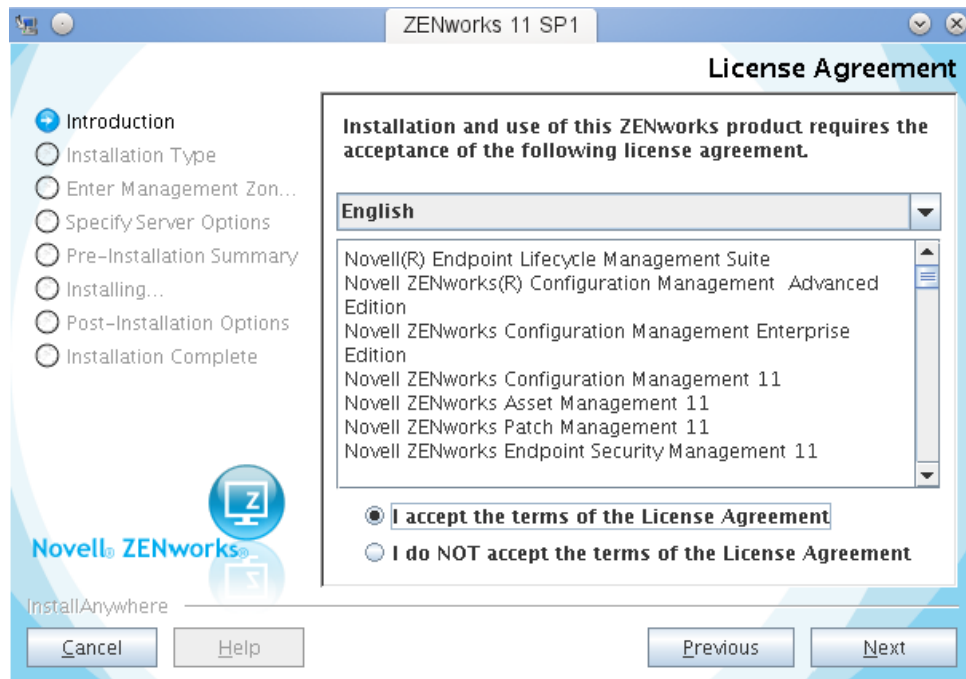
The introduction screen is shown next:

Figure 9-2 ZCM Primary Server Installation (2)



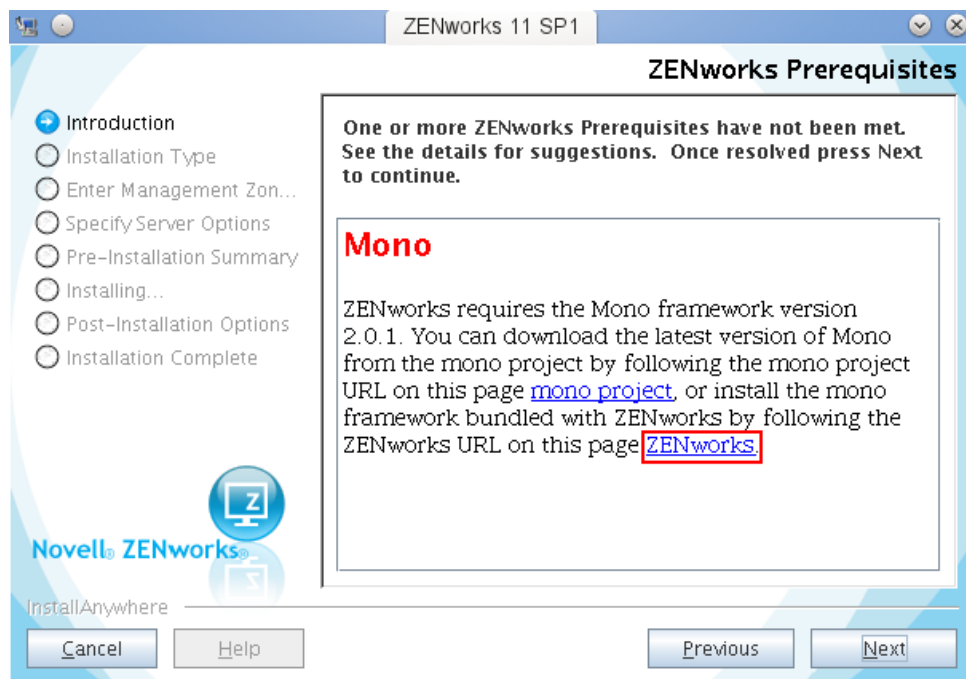
The License Agreement must be accepted:

Figure 9-3 ZCM Primary Server Installation (3)



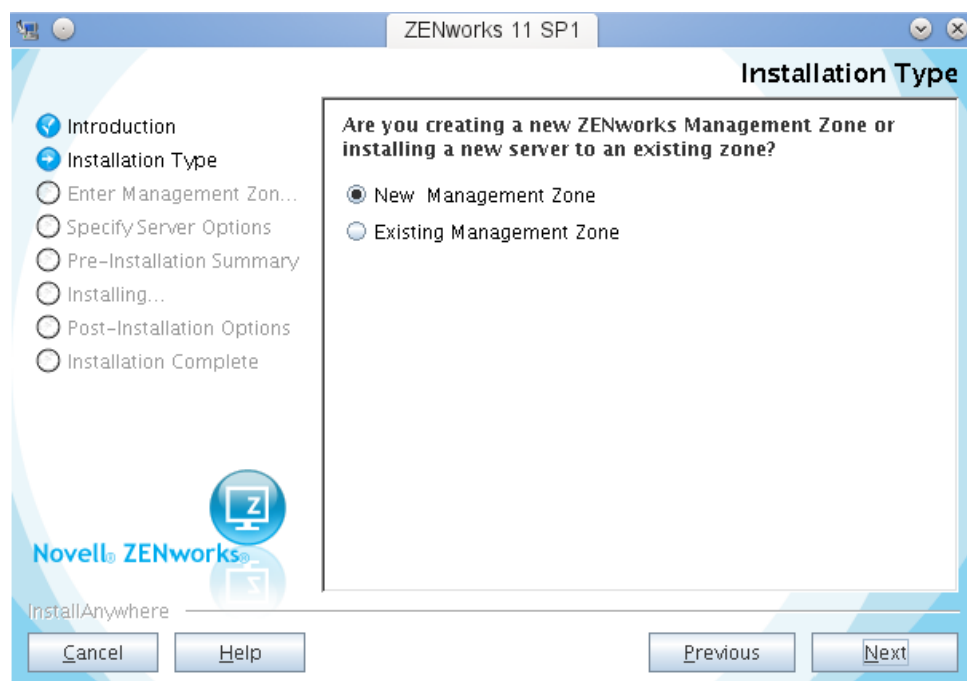
If Mono has not been previously installed, select the *ZENworks* link to install Mono from the ISO image itself:

Figure 9-4 ZCM Primary Server Installation (4)



The Primary Server is installed either into a new Management Zone or into an existing Management Zone:

Figure 9-5 ZCM Primary Server Installation (5)



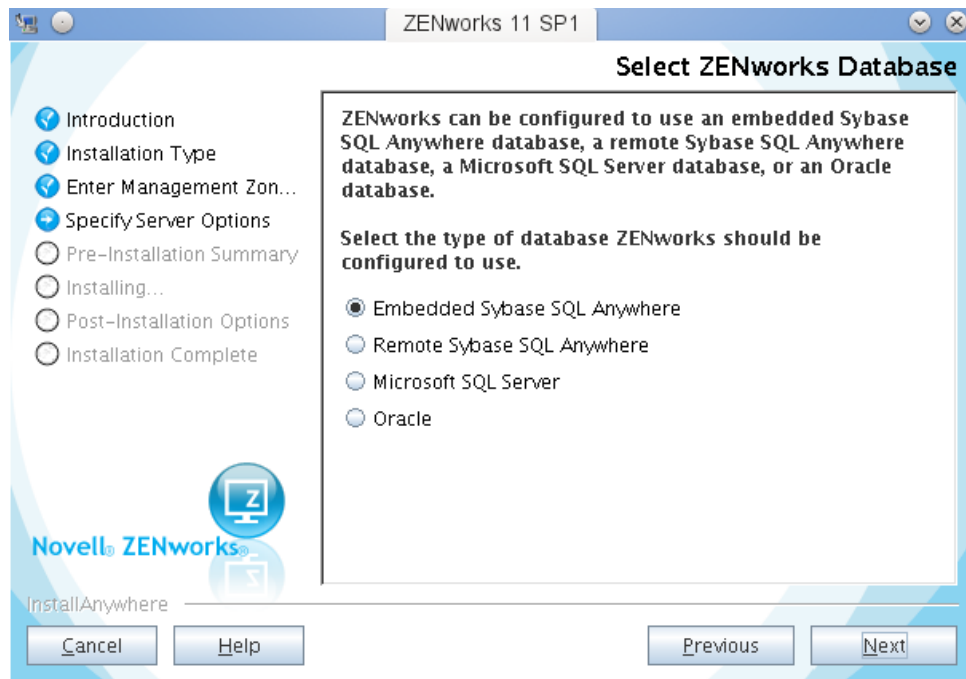
A zone name and the password for the zone Administrator must be specified in the next dialog box. The Administrator account is always Administrator (case-insensitive):

Figure 9-6 ZCM Primary Server Installation (6)



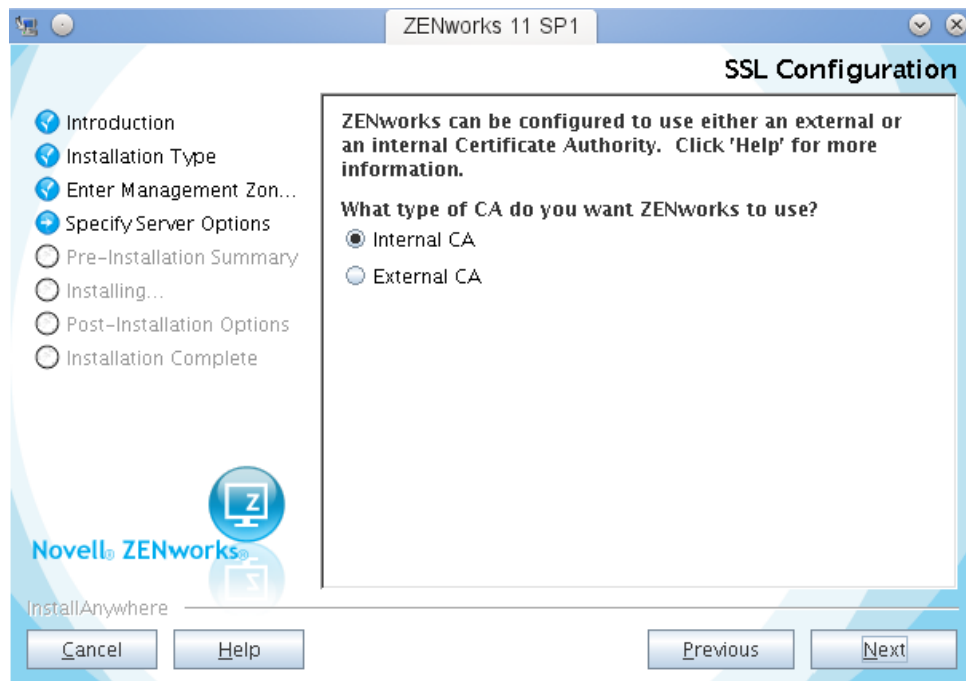
This installation example presumes the simplest case with an embedded Sybase SQL Anywhere database server:

Figure 9-7 ZCM Primary Server Installation (7)



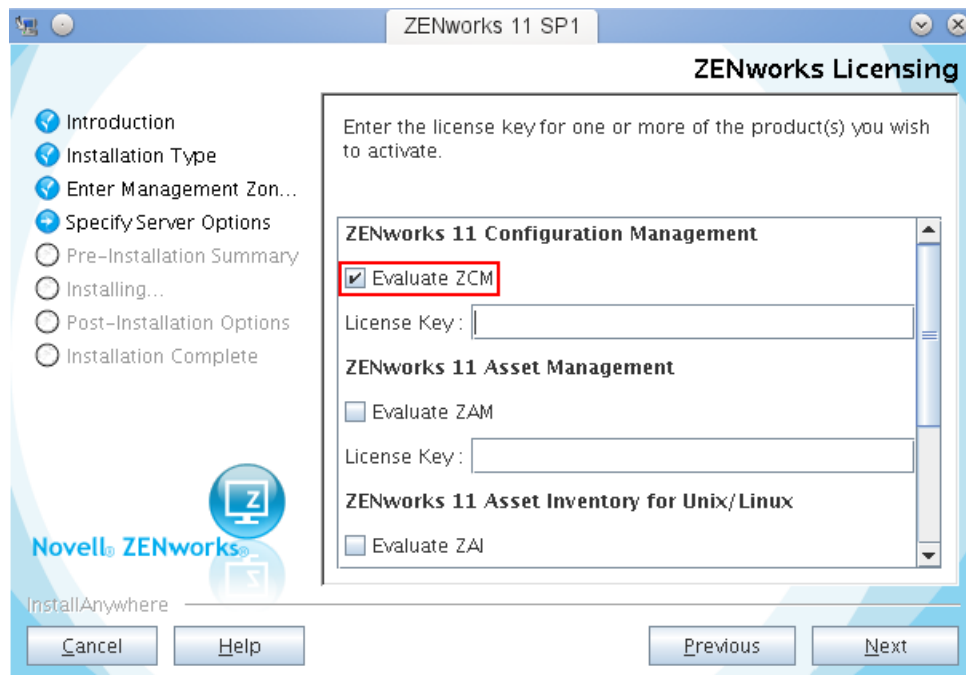
In most cases, an internal certificate authority (CA) is used for ZENworks Configuration Management. However, it is also possible to use an external CA. Information about how to configure an external CA is described in the Novell ZENworks Configuration Management installation documents in detail.

Figure 9-8 ZCM Primary Server Installation (8)



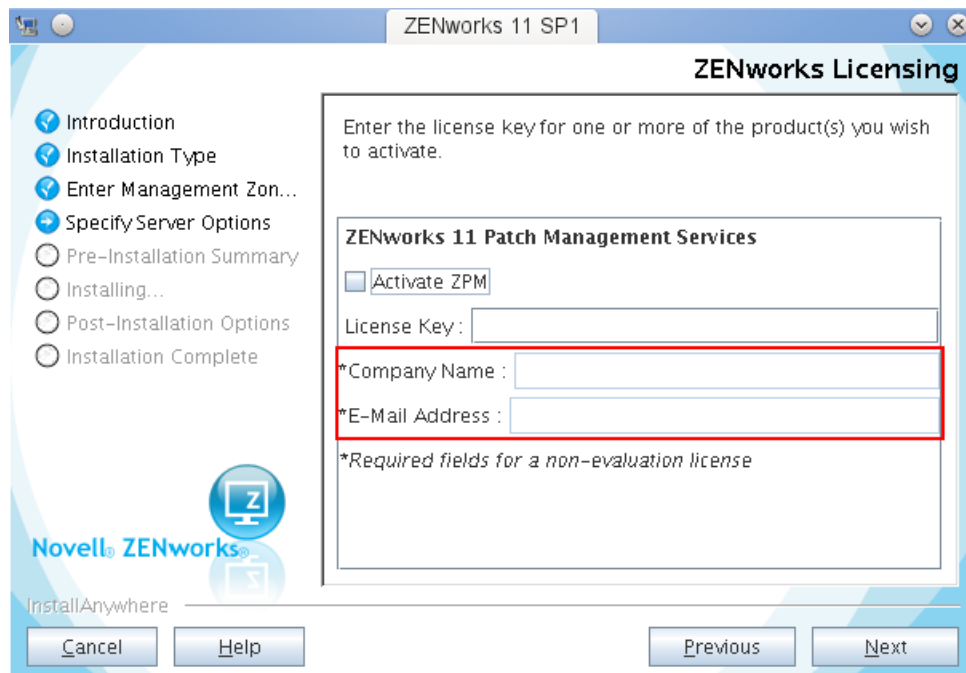
Managing SUSE Linux Enterprise Server and Open Enterprise Server requires only Configuration Management. Therefore, only one license key is needed:

Figure 9-9 Figure 9: ZCM Primary Server Installation (9)



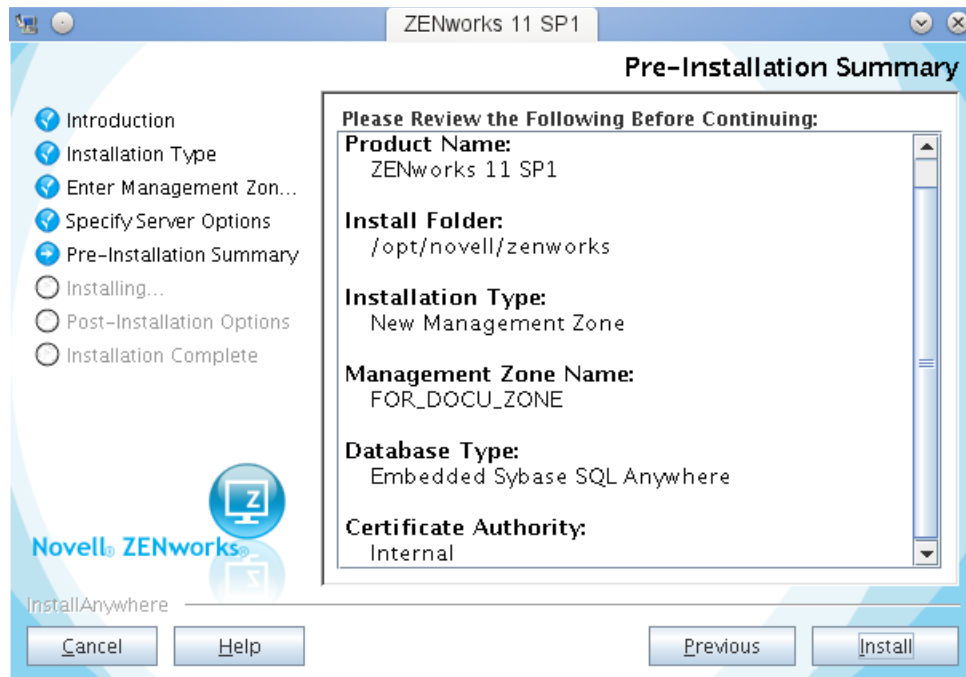
After the license has been supplied, the next dialog window requires your company name and an email address to be able to activate the selected components:

Figure 9-10 ZCM Primary Server Installation (10)



A summary is displayed before the installation begins:

Figure 9-11 ZCM Primary Server Installation (11)



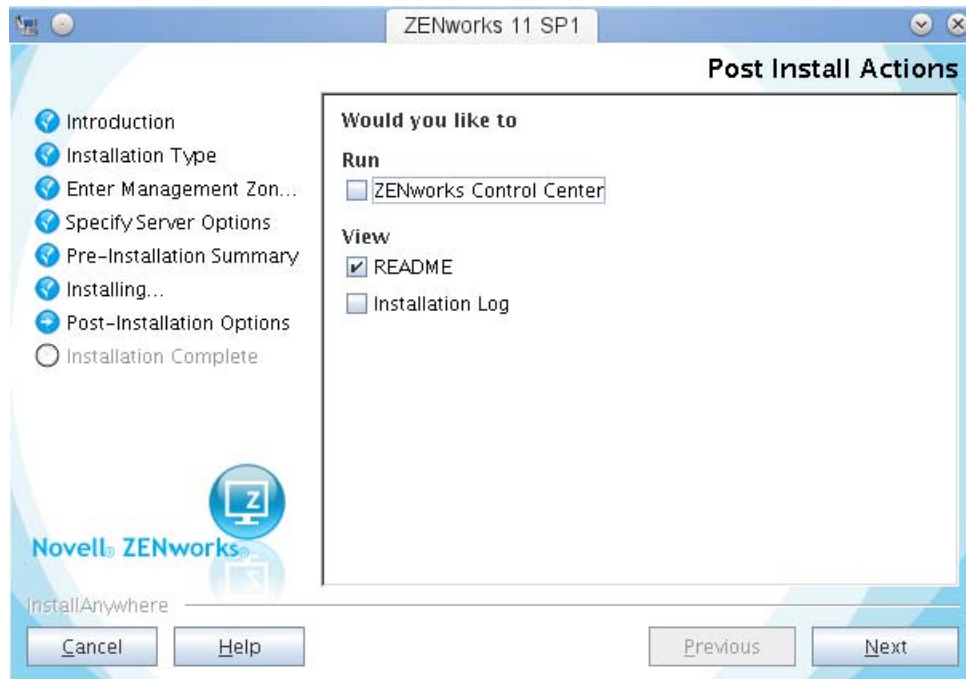
The installation needs some time. A progress bar gives information about the installation state:

Figure 9-12 ZCM Primary Server Installation (12)



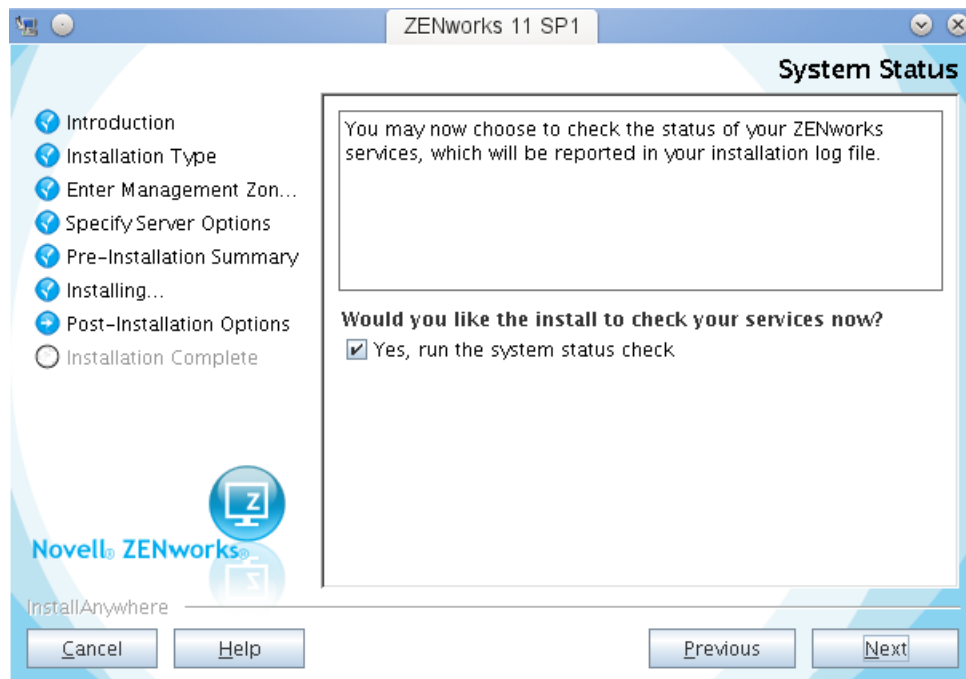
After the installation has finished, ZENworks Control Center can be started and you can view the Readme and the installation log:

Figure 9-13 ZCM Primary Server Installation (13)



Finally, you can execute a system status check to be sure that everything went well:

Figure 9-14 ZCM Primary Server Installation (14)



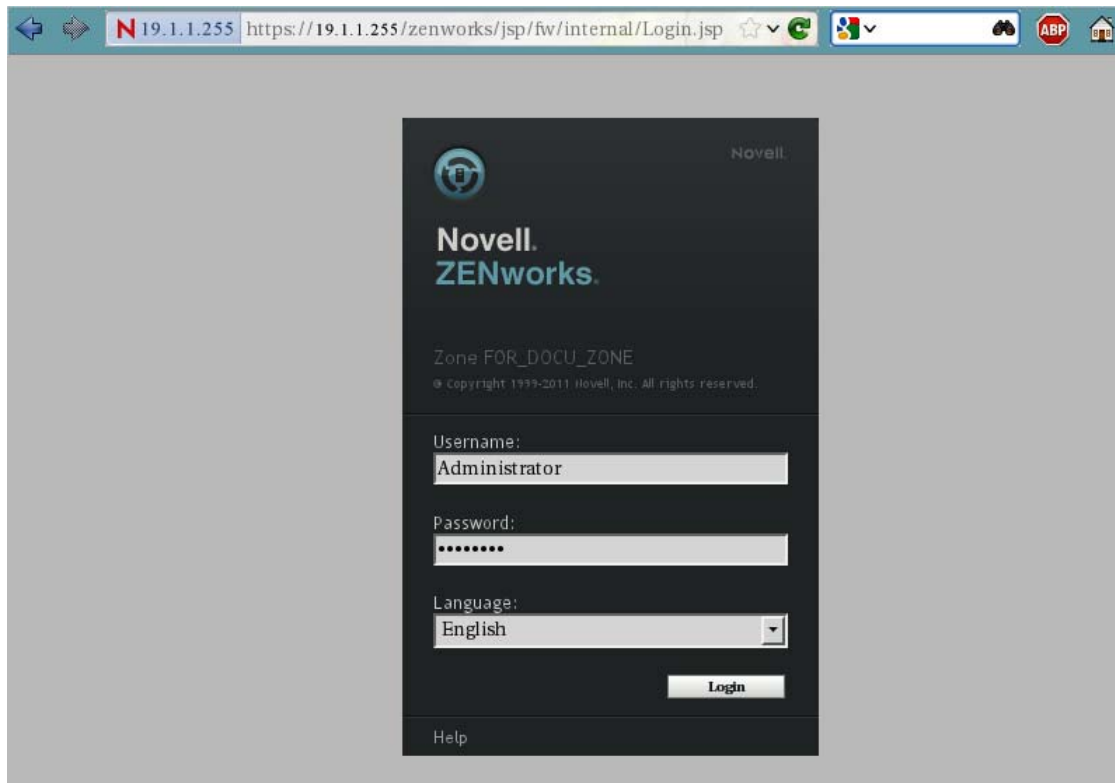
The final screen after a successful installation is shown below:

Figure 9-15 ZCM Primary Server Installation (15)



After installation, you can log in to the Configuration Management interface by pointing to the IP address of the server via a Web browser:

Figure 9-16 Login Page for the ZCM Server



10 Server Configuration

After the initial installation stage, Novell Consulting recommends that you change several default settings to suitable values. Most of them will be set to values that have less performance impact. The appropriate settings are described in this section.

- ♦ [Section 10.1, “Increasing the Content Replication Schedule,” on page 79](#)
- ♦ [Section 10.2, “Configuring Content Primary Server Replication,” on page 80](#)
- ♦ [Section 10.3, “Configuring the Inventory Service,” on page 81](#)
- ♦ [Section 10.4, “Configuring the Inventory Schedule,” on page 83](#)
- ♦ [Section 10.5, “Configuring the YUM Service Settings,” on page 84](#)
- ♦ [Section 10.6, “Locations and Network Environments,” on page 84](#)

10.1 Increasing the Content Replication Schedule

The content replication schedule determines how much time elapses before replication between Primary Servers and/or to Satellite servers. You change this setting in *Configuration > Content Replication*. The default value is 1 hour. You should increase it to 4 hours:

Figure 10-1 Content Replication Schedule

The screenshot shows the Novell ZENworks Configuration console. The left sidebar contains a navigation menu with options: Home, Devices, Users, Policies, Bundles, Deployment, Reports, Configuration, Configuration Tasks, Manage Licenses, and Message Cleanup. The main content area is titled 'Configuration > Content Replication'. Below this, there is a section for 'Content Replication' with a description: 'Configure the settings for content replication schedules and throttle rates.' The settings are as follows:

Setting	Value
Primary Server recurring content replication schedule:	0 Days, 1 Hours, 0 Minutes
Primary Server output throttling in KB/sec:	None
Agent Content Checksum:	On
Satellite Content Checksum:	On
Default Satellite Server replication throttling in KB/sec:	None
Default Satellite Server content replication Schedule Type:	Recurring

10.2 Configuring Content Primary Server Replication

This setting is of interest only in environments where there are multiple Primary Servers; however, it is discussed here because it affects disk space and network load.

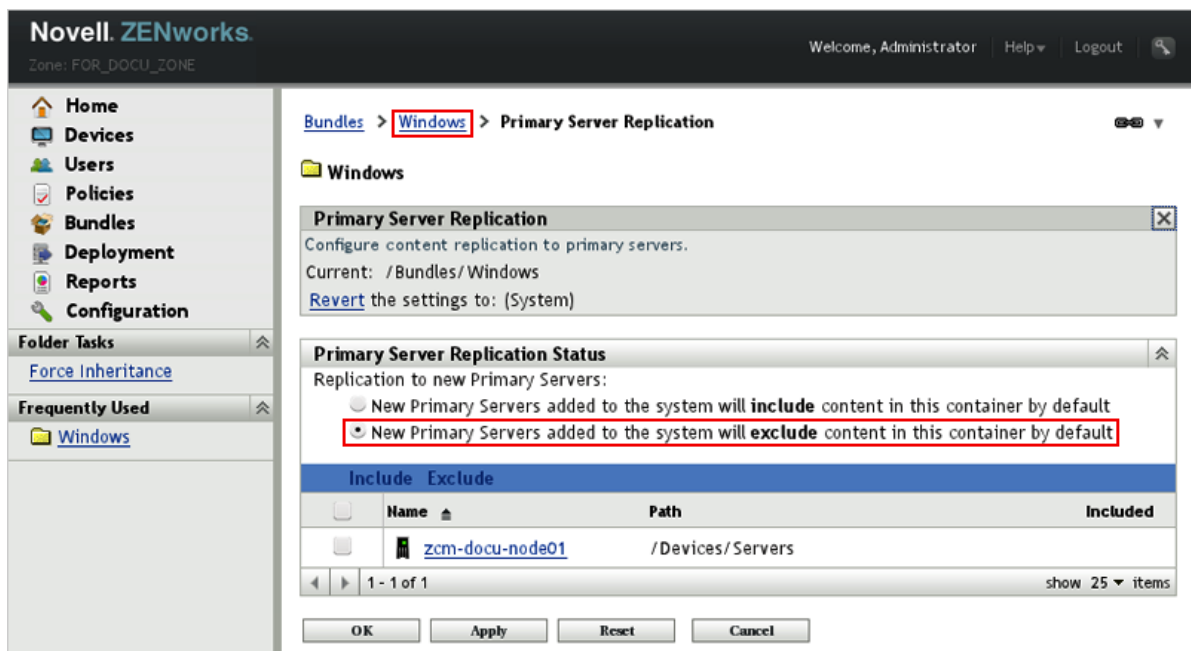
The setting determines if a Primary Server automatically receives content from other Primary Servers or if content is not delivered automatically to this server. This rule is configured with a default include value at the *Configuration > Primary Server Replication* menu, but it can be overwritten at various deeper folder levels.

Figure 10-2 Content Primary Server Default Rule



In environments with multiple Primary Servers, it is often desirable to keep one particular server for Linux patches only and to not replicate Windows patches to that server. If this is a requirement for you, you should exclude the server from the list of Primary Server replication at the Windows folder level. You might also want to avoid replicating all Linux patches to each Satellite Server.

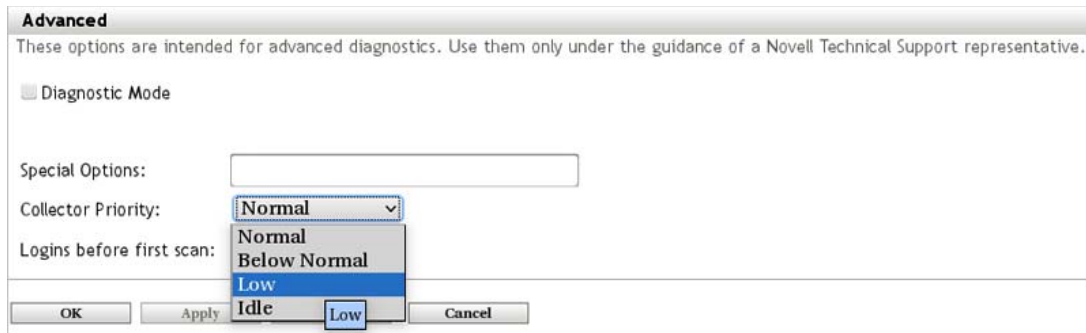
Figure 10-3 Content Replication at the Folder Level



10.3 Configuring the Inventory Service

The Collector Priority of the Inventory Service should be a matter of interest only in debugging situations; however, the default value is set to *Normal*. For performance reasons, you should set it to *Low*. The setting is configured in *Configuration > Inventory > Inventory* in the last option at the bottom of the page.

Figure 10-4 *Configuration > Inventory > Advanced > Collector Priority*



In addition, some directories must be excluded from the inventory scan because they contain content that is not of interest from an inventory perspective:

- ♦ /proc
- ♦ /sys
- ♦ /tmp
- ♦ /dev
- ♦ /media
- ♦ /mnt

Figure 10-5 *Inventory > Inventory > Software Files*

82 Novell Consulting Best Practices Guide: Automated Installation, Configuration, and Update for OES 2015 SP1

10.4 Configuring the Inventory Schedule

The next setting you should change is the *Configuration > Inventory Schedule*, which determines when the inventory scan at servers and workstations takes place. The default value of the Inventory Schedule is set to the first day of the month.

To avoid simultaneous inventory scans on multiple devices, Novell Consulting recommends that you configure a random start and end time on a working day (preferably in the middle of the week) to ensure that devices are online:

Figure 10-6 *Configuration > Inventory Schedule*

Scan Schedule

Specify the schedule the device inventory scanner should run on:

Schedule Type:
Recurring

☐ When a device is refreshed

☐ Delay execution after refresh: 0 Days 0 Hours 0 Minutes

Days of the week

Sun Mon Tue Wed Thu Fri Sat

☐ ☐ ☐ ☒ ☒ ☐ ☐

Start Time: 11 : 00

[Hide Options](#)

☐ Process immediately if device unable to execute on schedule

☐ Use Coordinated Universal Time (Current UTC 12:48 PM)

☒ Start at a random time between Start and End Times

End Time: 15 : 00

☐ Restrict schedule execution to the following date range:

Start Date: 1/16/2012

End Date: 1/16/2012

☐ Monthly

☒ Day of the month: 1

☐ Last day of the month

☐ First Sunday

Start Time: 1 : 00

[More Options](#)

☐ Fixed Interval

0 Months 0 Weeks 0 Days 0 Hours 0 Minutes

10.5 Configuring the YUM Service Settings

By default, the ZCM server updates the YUM service on the target devices every Saturday at 01:00. You can change this in the Infrastructure Management section of the Configuration page.

The following example configures the YUM Service refresh to happen daily at 04:00.

Figure 10-7 YUM Service Refresh

[Configuration](#) > YUM Service Settings

YUM Service Settings

Configure the YUM Service Refresh Schedule

Bundle Name	Relative URL	Auto Update	Can Refresh
OES11FP0SP1_PROD	/zenworks-yumrepo/OES11FP0SP1_PROD	false	Yes
OES2SP3_PROD	/zenworks-yumrepo/OES2SP3_PROD	false	Yes
SLES10SP3_PROD	/zenworks-yumrepo/SLES10SP3_PROD	false	Yes
SLES11-SP1-Pool	/zenworks-yumrepo/SLES11-SP1-Pool	false	Yes
SLES11-SP1-Updates-DEV	/zenworks-yumrepo/SLES11-SP1-Updates-DEV	true	Yes

1 - 5 of 8 show 5 items

YUM Service Refresh Schedule

Click [here](#) to Update YUM Services Now

Schedule Type:
Recurring

Days of the week

Sun	Mon	Tue	Wed	Thu	Fri	Sat
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Start Time: 4 : 00

[More Options](#)

Fixed Interval

0	Months	0	Weeks	0	Days	0	Hours	0	Minutes
---	--------	---	-------	---	------	---	-------	---	---------

Start Date: 3/21/2015 Start Time: 1 : 00

[More Options](#)

OK

Apply

Reset

Cancel

10.6 Locations and Network Environments

Any client that registers to a ZCM Zone automatically applies to a certain location. If no location is defined, it will be assigned to the ~unknown~ location. Because locations define which Primary Servers are contacted by the client, it is important to know how locations and their associated network environments are configured in existing ZENworks Configuration Management setups.




The settings described in the following section are not of any interest in environments with only one ZENworks Configuration Management server.

- [Section 10.6.1, “Locations,” on page 85](#)
- [Section 10.6.2, “Network Environment,” on page 87](#)

10.6.1 Locations

The main purpose of a location is to define content, authentication, configuration, and collection servers that can be contacted by all clients of the associated network environment. Rules defined here need to ensure that clients always contact their closest servers from a LAN/WAN perspective and also guarantee high availability if servers defined here are temporarily not reachable. The menu for creating or editing locations is shown in the following figure:


Figure 10-8 Locations Tab

Configuration	Registration	System Information	Asset Inventory	System Updates	Locations	Subscriptions
Locations						⌵
New Rename Delete Move Up Move Down						<input type="text"/> 🔍 ↻
<input type="checkbox"/>	Name			Reference Count		
<input type="checkbox"/>	 DUS				<u>0</u>	
<input type="checkbox"/>	 ~unknown~				<u>0</u>	
⏪ ▶ 1 - 2 of 2					show 5 ▼ items	
Network Environments						⌵
New Rename Delete						<input type="text"/> 🔍 ↻
<input type="checkbox"/>	Name <small>⌵</small>		Reference Count		Location Reference Count	
<input type="checkbox"/>	 DUS NET			<u>0</u>	<u>1</u>	
⏪ ▶ 1 - 1 of 1					show 5 ▼ items	

The *Details* tab displayed in the following figure shows the network environment assigned to the location DUS (DUS NET):

Figure 10-9 Details Tab

Locations > DUS

 DUS

Details Servers Relationships

General

GUID:

e6ae9c65fa7ee63efa379903440b9582

Description:

Main department of our company

Throttle Rate (in kbps):

0

Last Modified:

4/3/2012 7:38 AM

Assigned Network Environments

Add

Remove


Move Up

Move Down

☐

Name

☐

 DUS NET

1 - 1 of 1

show 5 items

A location can consist of several physical subnets, so it is not unusual that a location has multiple network environments assigned. The *Servers* tab of the *Location* menu is intended to configure the servers or server groups associated to a location.

To avoid any uncontrolled device assignments, you should also exclude the Closest Server Default Rule from all location objects:

Figure 10-10 Servers Tab

Locations > DUS

DUS

Details Servers Relationships

Location Closest Servers

Configure the setting for how managed devices determine their closest servers using the location

☒ Exclude the Closest Server Default Rule

Collection Servers:

Move Up Move Down Add Remove Groups L4 Switch

Name
DUS Collection Server Group
/Devices/Servers/zcm-docu-node01

Content Servers:

Move Up Move Down Add Remove Groups L4 Switch

Name
DUS Content Server Group
/Devices/Servers/zcm-docu-node01

Configuration Servers:

Move Up Move Down Add Remove Groups L4 Switch

Name
DUS Configuration Server Group
/Devices/Servers/zcm-docu-node01

Authentication Servers:

Move Up Move Down Add Remove Groups L4 Switch

Name
DUS Authentication Server Group
/Devices/Servers/zcm-docu-node01

Limit Servers Returned to Agent: ☒ Unlimited ☐ Limit to servers per list

10.6.2 Network Environment

A device uses its current network environment to determine its configuration location. A network environment defines several properties such as DNS servers, DHCP servers, gateways, and subnets, as well as others that the agent uses to determine the location it belongs to.

Figure 10-11 Network Environment

The screenshot shows the 'Locations' and 'Network Environments' sections of a configuration tool. The 'Locations' section has a table with columns 'Name' and 'Reference Count'. It lists 'DUS' and '-unknown-' with a reference count of 0. The 'Network Environments' section has a table with columns 'Name', 'Reference Count', and 'Location Reference Count'. It lists 'DUS NET' with a reference count of 0 and a location reference count of 1.

Locations	
Name	Reference Count
DUS	0
-unknown-	0

Network Environments		
Name	Reference Count	Location Reference Count
DUS NET	0	1

The following figure demonstrates how to set up the different properties that define a unique network environment. The configuration takes place at *Configuration > Locations > Network Environments*, where you can create a new network environment or edit an existing one.

In this example, clients identify their environment via the gateway and its IP address properties.

Figure 10-12 Network Environment - Details

The screenshot shows the 'Details' tab for the 'DUS NET' network environment. It includes fields for GUID, Description, Throttle Rate, and Last Modified. It also shows a list of associated locations. The 'Network Environment Settings' section includes a dropdown for 'Limit to Adapter Type' and a 'Minimum Match' field. The 'Gateways' section has a table with columns 'Operator', 'IP Address', 'MAC Address', and 'Match Required'. The 'IP Address' field is highlighted with a red box.

Network Environments > DUS NET

DUS NET

Details Servers Relationships

General

GUID: 92ab9e2415982a04e68d1d10e8797f9b

Description:

Throttle Rate (in kbps): 0

Last Modified: 4/3/2012 8:04 AM

References

Associated Location(s): /Locations/DUS

Network Environment Settings

Limit to Adapter Type: All

Minimum Match: 1

Gateways

DNS Servers DHCP Servers WINS Servers Dial-up Connections Adapters Access Points Client IP Address Client DNS Settings

Add Edit Remove

Operator	IP Address	MAC Address	Match Required
=	192.168.1.255		✓

1 - 1 of 1 show 5 items

Apply Reset

11 Managing ZENworks Configuration Management

ZENworks Configuration Management provides numerous possibilities and paths to manage Linux devices. Areas like patch management, configuration management, and remote management can be administered in various different ways. In practice, the number of possibilities often leads to inefficient or inoperational management strategies.

In this section, Novell Consulting demonstrates how to design and administer the ZENworks Configuration Management framework to perform recurring operations required in enterprise environments, such as provisioning different patch levels, managing different staging areas, and assigning patches or configuration items to multiple devices.

In addition, we discuss topics like unique naming standards, specific configuration bundles for Open Enterprise Server (OES) 11 devices, and administration of the ZENworks Configuration Management agent on Linux.

- ♦ [Section 11.1, “Overview,” on page 89](#)
- ♦ [Section 11.2, “Folders,” on page 90](#)
- ♦ [Section 11.3, “Devices,” on page 93](#)
- ♦ [Section 11.4, “Bundles,” on page 109](#)
- ♦ [Section 11.5, “Subscriptions,” on page 142](#)
- ♦ [Section 11.6, “zman,” on page 154](#)

11.1 Overview

The ZENworks Configuration Management management methodology developed by Novell Consulting is based on the following principles:

- ♦ The ZENworks Configuration Management environment is configured to manage three different staging areas: Development, Test, and Production (referred to as DEV, TEST, and PROD). These stages are considered in the device and key folder structures and their associated naming standards.
- ♦ Any device managed via ZENworks Configuration Management is registered during its installation and is assigned to different device groups based on the product and support pack being installed, and on its function. Up to three registration keys are used:
 - ♦ **Location key:** Determines the placement of the device object in the ZENworks Configuration Management folder structure.
 - ♦ **Update Group key:** Determines membership in device groups that govern patch assignments.
 - ♦ **Configuration Group key:** Determines membership in device groups that manage deployment of configuration settings.

- ♦ Software updates (patches) are managed in the following ways:
 - ♦ A regular update bundle from the Novell patch channel for a specific product and service pack (such as SLES 11 SP1) is copied at a specific time to obtain a frozen version that is not updated again from this point in time.

The bundle is initially intended for the DEV stage and is named according to a defined naming standard. For more information, see [Section 12.1.1, “Naming Standards for Frozen Patch Levels,” on page 155.](#)
 - ♦ The assignment of the frozen bundle takes place through a bundle group that is assigned to a device group with a development purpose.

When the bundle has proven its stability within the DEV environment, it is assigned to a bundle group representing the next stage. After passing all verifications in the TEST environment, it is assigned to the final production stage. A number of older frozen patch levels and patch levels that might be required for software components of other vendors (such as Oracle) or for other needs are retained.
 - ♦ When a new frozen patch level needs to be deployed, only the old bundle is removed from the bundle group and the new bundle for the new frozen patch level is added to it. The assignment of the bundle group to the device group is not changed.
 - ♦ In addition to the frozen patch level for updates assigned through bundle groups, the pool bundles (for OES 11, SLES 11 and SLES 11 SP1) and the core bundles (SLES 11 SP2 and later) are directly assigned to the device groups for dependency resolution purposes. Because these bundles never change, no bundle groups are required.

Core bundles have been introduced as a consequence of a recent change to the SUSE Linux Enterprise 11 Maintenance Model.
- ♦ Configuration bundles that manage configuration settings on the target systems or that deploy software from other vendors are also assigned through bundle groups to devices and device groups.

11.2 Folders

Folders are a main concept in ZENworks Configuration Management in order to organize the administration of different types of objects. Folders are present in nearly every menu page.

- ♦ [Section 11.2.1, “General Rules for Folder Names,” on page 90](#)
- ♦ [Section 11.2.2, “Creating a New Folder,” on page 91](#)
- ♦ [Section 11.2.3, “Deleting a Folder,” on page 92](#)
- ♦ [Section 11.2.4, “Renaming a Folder,” on page 92](#)

11.2.1 General Rules for Folder Names

To ensure similar naming standards for folders in the different menus, you need to follow certain naming rules for folder names. The following rules apply for menus in ZENworks Control Center:

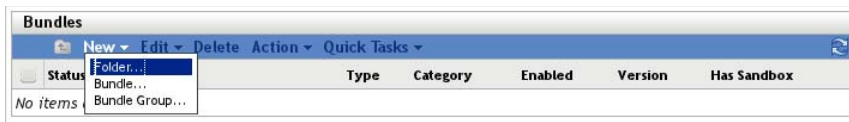
- ♦ Folder names use uppercase
- ♦ Folder names must be self-explanatory
- ♦ When you create a new folder, you must provide an explanatory description

In addition to these general rules, additional naming schemes are used for particular menus. These naming schemes are discussed in the sections for the menus.

11.2.2 Creating a New Folder

In any menu, select *New > Folder*, fill in the fields in the New Folder dialog box, then click *OK*.

Figure 11-1 Creating a New Folder (1)



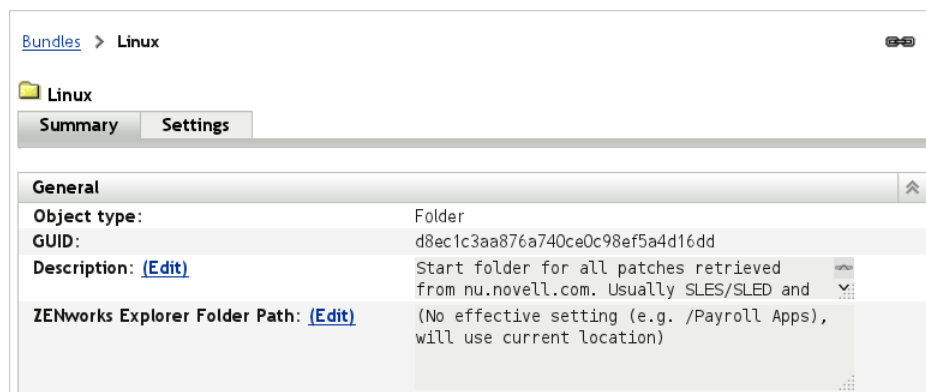
Novell Consulting strongly recommends that you use the *Description* field to clearly describe the purpose of the object. The description and folder name can be changed after folder creation.

Figure 11-2 Creating a New Folder (2)

A screenshot of the 'New Folder' dialog box. It has a title bar with a question mark and close button. The 'Name' field contains 'Linux'. The 'Folder' field contains '/Bundles'. The 'Description' field contains the text: 'Start folder for all patches retrieved from nu.novell.com. Usually SLES/SLED and OES patches.' Below the fields is a note: '* Fields marked with an asterisk are required.' At the bottom are 'OK' and 'Cancel' buttons.

The new folder is ready for use.

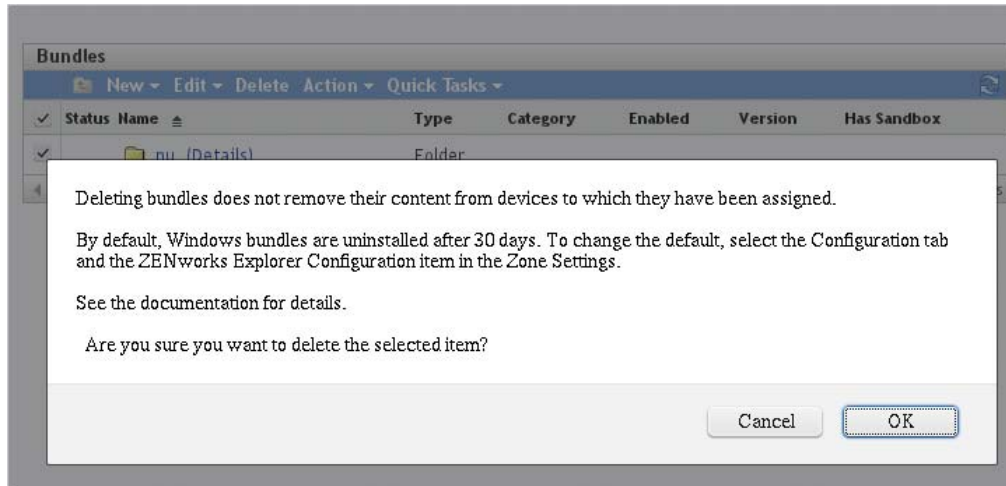
Figure 11-3 A New Folder



11.2.3 Deleting a Folder

If one or more folders must be deleted, select the check boxes to the left of the folder icons and click *Delete*. A warning pops up for you to confirm the deletion. When you click *OK*, the folders and all of their contents are deleted.

Figure 11-4 *Deleting a Folder*

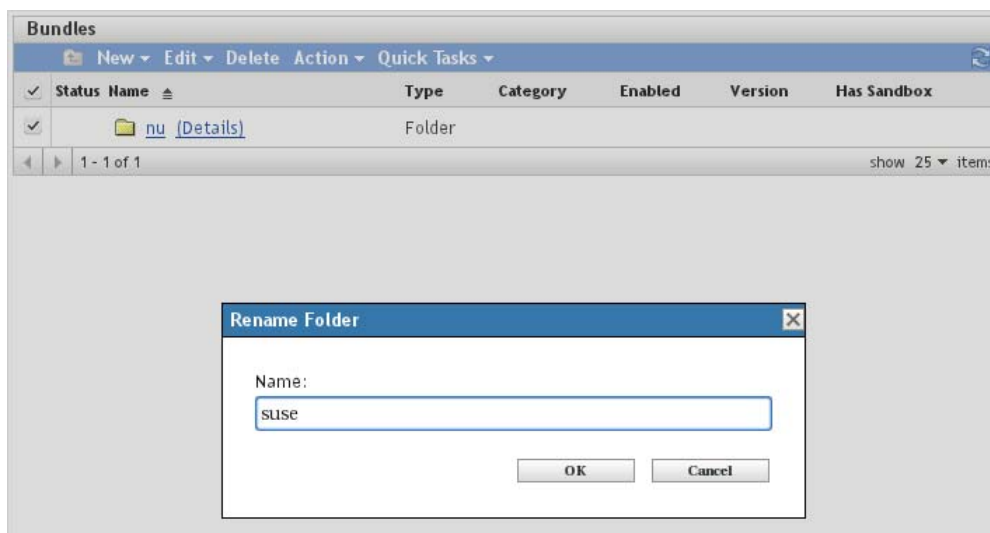


11.2.4 Renaming a Folder

Folder names in ZENworks Configuration Management are not case sensitive. Internally, folder objects are represented by a globally unique identifier (GUID). Renaming a folder does not change the GUID. A folder can be renamed at any time without breaking any assignment or changing folder content. If a folder must be renamed (for example from all uppercase to all lowercase to comply with a naming standard) it must first be renamed to a different name.

To rename a folder, select the check box to the left of the folder icon and click *Edit > Rename*.

Figure 11-5 *Renaming a Folder*



11.3 Devices

The device menu contains all device objects and device group objects that are registered at the ZENworks Configuration Management server. It is subdivided into *Servers* and *Workstations*, which are defaults that cannot be changed. In this document, only the *Servers* are discussed.

After a successful configuration of the ZENworks Configuration Management server, the *Devices/Servers* branch contains an object for the server itself and several dynamic server groups, as illustrated in [Figure 11-7 on page 94](#). For administrative purposes, you should introduce a folder structure below the *Servers* branch to represent your server environment. This helps to ease the administration of large infrastructures.

Figure 11-6 Devices Menu



Name	Type
Servers (Details)	Folder
Workstations (Details)	Folder

11.3.1 Folders in the Device Menu

Every device is represented in ZENworks Configuration Management by an object. Every device object can only exist in one place (folder) in ZENworks Configuration Management. This place can be the first folder level (*Devices/Servers*) in ZENworks Configuration Management, but it can also be at any other level within the ZENworks Configuration Management device folder structure.

The location of the device object should have no functional meaning. When a certain type of software (bundle, policy, and so forth) is assigned to folders, the assignment is inherited by all objects located below that folder, including any subfolder. However, this kind of assignment should be avoided; instead, you should use device group assignments for administrative reasons.

The folder structure itself does not depend on technical requirements. Novell Consulting recommends that you develop a folder hierarchy that is based on functional and organizational aspects of administration. If you are managing different operating systems, you might want to create a folder for each of them inside the *Servers* folder.

It is also a good practice to implement naming standards for any type of object that is created in ZENworks Configuration Management. Novell Consulting recommends that you name folders in all uppercase.

Figure 11-7 Device Folder Structure

Devices > Servers

Devices					
New ▾ Edit ▾ Delete Action ▾ Quick Tasks ▾					
<input type="checkbox"/>	Status	Name	Type	Operating System	Last Contact Retired
<input type="checkbox"/>		GROUPS (Details)	Folder		
<input type="checkbox"/>		NOKEY (Details)	Folder		
<input type="checkbox"/>		SERVERS (Details) View Details	Folder		
<input type="checkbox"/>		Open Enterprise Server 2	Dynamic Server Group		
<input type="checkbox"/>		Red Hat Enterprise Linux Server 4	Dynamic Server Group		
<input type="checkbox"/>		Red Hat Enterprise Linux Server 5	Dynamic Server Group		
<input type="checkbox"/>		Red Hat Enterprise Linux Server 6	Dynamic Server Group		
<input type="checkbox"/>		SUSE Linux Enterprise Server 10	Dynamic Server Group		
<input type="checkbox"/>		SUSE Linux Enterprise Server 11	Dynamic Server Group		
<input type="checkbox"/>		Windows Server 2003	Dynamic Server Group		
<input type="checkbox"/>		Windows Server 2008	Dynamic Server Group		
<input type="checkbox"/>		Windows Server 2008 R2	Dynamic Server Group		
<input type="checkbox"/>		zcm-docu-node01	Server	sles-11-x86_64	10:15 AM

1 - 13 of 13 show 25 items

The folders shown in this figure have the following purposes:

- ♦ **SERVERS:** Every device that will be registered at the ZENworks Configuration Management server is represented as a unique object within the ZENworks Configuration Management framework. These objects are placed in the `Devices/Servers/SERVERS` branch.
- ♦ **NOKEYS:** The `Devices/Servers/NOKEYS` folder collects devices registered accidentally with a key created for group assignment only.
- ♦ **GROUPS:** Most configuration item assignments take place between groups of devices and certain objects like bundles and policies. For this reason, we have decided to create a parent `GROUPS` folder that collects all device group objects.

This folder is divided into subfolders for the DEV, TEST, and PROD environments. Each of these folders holds two folders for configuration and update objects.

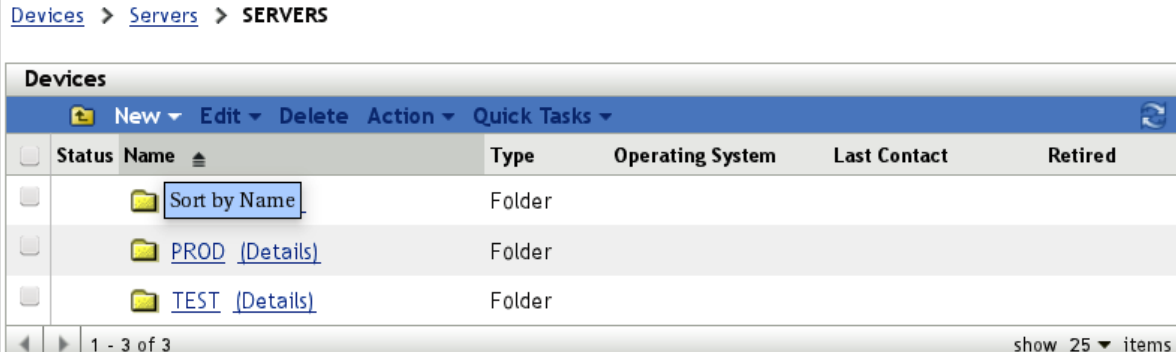
The folder structure for devices is illustrated in the following sections:

- ♦ [“The SERVERS Folder” on page 95](#)
- ♦ [“The NOKEYS Folder” on page 96](#)
- ♦ [“The GROUPS Folder” on page 96](#)




The SERVERS Folder

The suggested structure of this branch combines functional aspects with an existing eDirectory tree or location structure. A new device is placed below the DEV, TEST, or PROD folder, depending on whether it is a development, production, or test machine.

Figure 11-8 SERVERS Folder

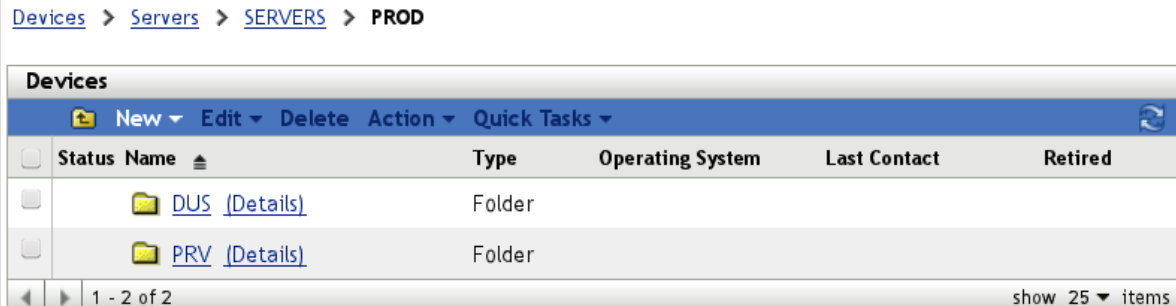


The screenshot shows the 'SERVERS' folder selected in the navigation pane. The main area displays a table of devices. The table has columns for Status, Name, Type, Operating System, Last Contact, and Retired. There are three rows, each representing a folder: 'Sort by Name', 'PROD (Details)', and 'TEST (Details)'. The 'Type' column for all rows is 'Folder'. The 'Status' column has a checkbox for each row. The 'Name' column has a folder icon and a link to details. The 'Type' column has the text 'Folder'. The 'Operating System', 'Last Contact', and 'Retired' columns are empty. The bottom of the table shows '1 - 3 of 3' and 'show 25 items'.

Status	Name	Type	Operating System	Last Contact	Retired
<input type="checkbox"/>	 Sort by Name	Folder			
<input type="checkbox"/>	 PROD (Details)	Folder			
<input type="checkbox"/>	 TEST (Details)	Folder			

The device object is also placed in a folder representing its physical location (and its eDirectory context) and finally in a folder indicating its main function, such as an eDirectory server (EDIR) or a member of a specific cluster (DUSCL01). This structure is illustrated in the following figures:

Figure 11-9 PROD Folder in the Servers Menu



The screenshot shows the 'PROD' folder selected in the navigation pane. The main area displays a table of devices. The table has columns for Status, Name, Type, Operating System, Last Contact, and Retired. There are two rows, each representing a folder: 'DUS (Details)' and 'PRV (Details)'. The 'Type' column for all rows is 'Folder'. The 'Status' column has a checkbox for each row. The 'Name' column has a folder icon and a link to details. The 'Type' column has the text 'Folder'. The 'Operating System', 'Last Contact', and 'Retired' columns are empty. The bottom of the table shows '1 - 2 of 2' and 'show 25 items'.



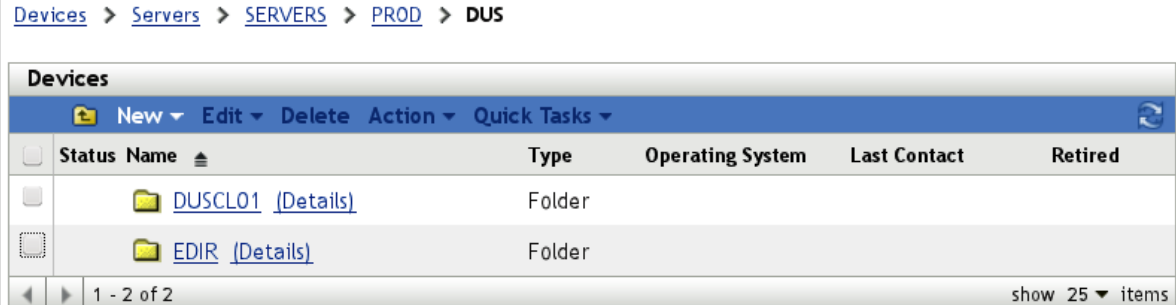


Status	Name	Type	Operating System	Last Contact	Retired
<input type="checkbox"/>	 DUS (Details)	Folder			
<input type="checkbox"/>	 PRV (Details)	Folder			

Figure 11-10 Location Folder in the Servers Menu



The screenshot shows the 'DUS' folder selected in the navigation pane. The main area displays a table of devices. The table has columns for Status, Name, Type, Operating System, Last Contact, and Retired. There are two rows, each representing a folder: 'DUSCL01 (Details)' and 'EDIR (Details)'. The 'Type' column for all rows is 'Folder'. The 'Status' column has a checkbox for each row. The 'Name' column has a folder icon and a link to details. The 'Type' column has the text 'Folder'. The 'Operating System', 'Last Contact', and 'Retired' columns are empty. The bottom of the table shows '1 - 2 of 2' and 'show 25 items'.

Status	Name	Type	Operating System	Last Contact	Retired
<input type="checkbox"/>	 DUSCL01 (Details)	Folder			
<input type="checkbox"/>	 EDIR (Details)	Folder			

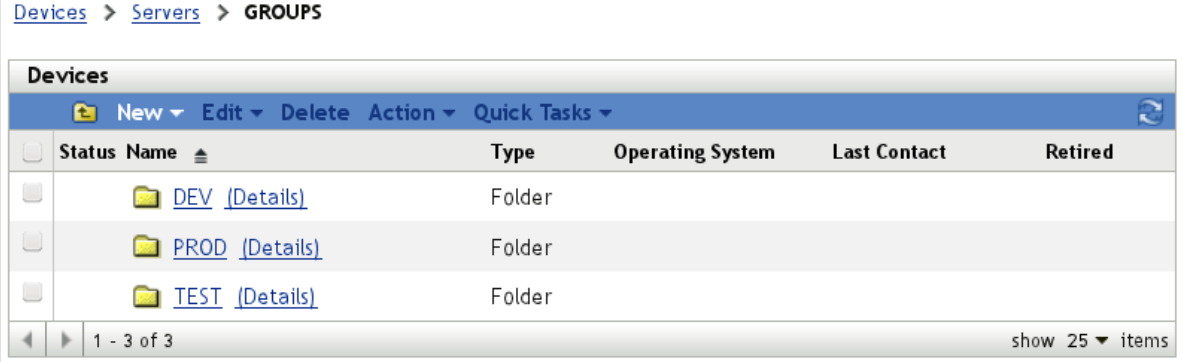
The NOKEYS Folder

This folder serves as a container for all devices registered incidentally with a key that is not intended for registration but for group assignment. It is configured as a target folder to all keys that are intended only for group assignment. In an ideal scenario, this folder remains empty.

The GROUPS Folder

For the `Server` folder, the first level below the `Groups` folder represents the environment for which the server groups are intended:

Figure 11-11 GROUPS Folder in the Device Menu

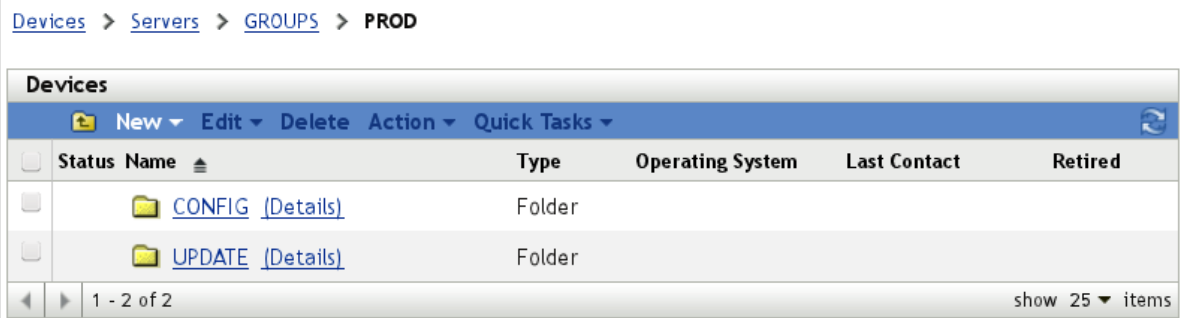


The screenshot shows the 'GROUPS' folder in the 'Servers' menu. The breadcrumb path is 'Devices > Servers > GROUPS'. The table lists three folders: DEV, PROD, and TEST, each with a '(Details)' link. The table has columns for Status, Name, Type, Operating System, Last Contact, and Retired. The bottom of the table shows '1 - 3 of 3' items and a 'show 25 items' option.

Status	Name	Type	Operating System	Last Contact	Retired
	DEV (Details)	Folder			
	PROD (Details)	Folder			
	TEST (Details)	Folder			

Each of these environment folders has a folder to store group objects for configuration or update assignments:

Figure 11-12 CONFIG and UPDATE Folder in the Device Menu



The screenshot shows the 'PROD' folder in the 'GROUPS' menu. The breadcrumb path is 'Devices > Servers > GROUPS > PROD'. The table lists two folders: CONFIG and UPDATE, each with a '(Details)' link. The table has columns for Status, Name, Type, Operating System, Last Contact, and Retired. The bottom of the table shows '1 - 2 of 2' items and a 'show 25 items' option.

Status	Name	Type	Operating System	Last Contact	Retired
	CONFIG (Details)	Folder			
	UPDATE (Details)	Folder			

11.3.2 Server Group Objects in the Device Menu

Server groups are objects that represent multiple server objects. They are placed at folder level 5. You can use this object type to more easily handle associations between tasks, software, and devices.

For instance, if a certain bundle must be delivered to many servers, the task can be achieved by assigning the bundle group containing this bundle to every server object individually. However, if the assignment needs to be deleted later, you would need to touch every server object.






A much better solution is to make the appropriate server objects members of a server group and to assign the desired bundle group to this group. This delivers the bundle to all members of the server group, and if a bundle association is removed from the bundle group, it is automatically removed from all members of the device group.

Novell Consulting distinguishes between two different roles for device groups:

- ♦ **Update Groups:** Servers that are on the same product, version and support pack, such as all SLES 11 SP1 servers, are combined into one server group. The following bundles are assigned to these groups:
 - ♦ Pool/core bundles for dependency resolution
 - ♦ Bundle group providing the frozen patch level

Figure 11-13 Server Groups below the UPDATE Folder

Devices > Servers > GROUPS > PROD > UPDATE

Devices					
New Edit Delete Action Quick Tasks					
Status	Name	Type	Operating System	Last Contact	Retired
	 PROD_OES11GM	Server Group			
	 PROD_OES2SP3	Server Group			
	 PROD_SLES10SP3	Server Group			
	 PROD_SLES10SP4	Server Group			
	 PROD_SLES11SP1	Server Group			



1 - 5 of 5 show 25 items

For more information, see [Section 11.4.2, “Bundles and Bundle Groups,” on page 113](#).

- ♦ **Configuration Groups:** Devices with identical configuration requirements are combined into configuration device groups. Application packages such as antivirus software or configuration files such as `multipath.conf` are assigned to these device groups through a corresponding bundle group. Device group objects for dedicated eDirectory servers and Xen hosts are examples.

Figure 11-14 Server Groups below the CONFIG Folder

Devices > Servers > GROUPS > PROD > CONFIG

Devices					
New Edit Delete Action Quick Tasks					
Status	Name	Type	Operating System	Last Contact	Retired
	 PROD_EDIR	Server Group			
	 PROD_XEN	Server Group			

1 - 2 of 2 show 25 items

A configuration device group can have multiple bundle groups assigned. For example, eDirectory servers can be running either physically or virtually on VMware or XEN hosts. These servers must be a member of a group that provides configurations relevant to eDirectory, such

as `PROD_EDIR`. They must also be a member of a group that provides software or configurations that are necessary to install or configure aspects of the virtual machine, such as the configuration group `PROD_VMWARE` that assigns `VMware-Tools` to VMware virtual machines.

- ♦ [“Server Group Location” on page 98](#)
- ♦ [“Naming Standards for Server Groups” on page 98](#)
- ♦ [“Server Group Creation” on page 99](#)

Server Group Location

Server groups should be placed below the `Devices/GROUPS` folder within their associated environment folder (`DEV`, `TEST`, or `PROD`). Depending on whether they are of the Update Group or Configuration Group type, they are placed into the corresponding folders.

Naming Standards for Server Groups

In general, Novell Consulting distinguishes between two types of server groups and therefore uses two different naming schemes:

- ♦ [“Naming Scheme for Update Groups” on page 98](#)
- ♦ [“Naming Scheme for Configuration Groups” on page 98](#)

Naming Scheme for Update Groups

`%ENVIRONMENT%_%PRODUCT%VERSION%SERVICE PACK%`

Name	Description
ENVIRONMENT	DEV TEST PROD This part of a group name identifies the environment that is relevant for an update group.
PRODUCT	SLES OES This part of the naming standard identifies the product for which the update group provides patches and dependency bundles.
VERSION	2 10 11 ... This part of a group name identifies the product version for which the update group is intended.
SERVICE PACK	GA SP1 SP2 SP3 ... The final part of a group name identifies the service pack level that is installed on the members of the device group. If no service pack is installed, Novell Consulting recommends that you use the term GA, which stands for General Availability.

Naming Scheme for Configuration Groups

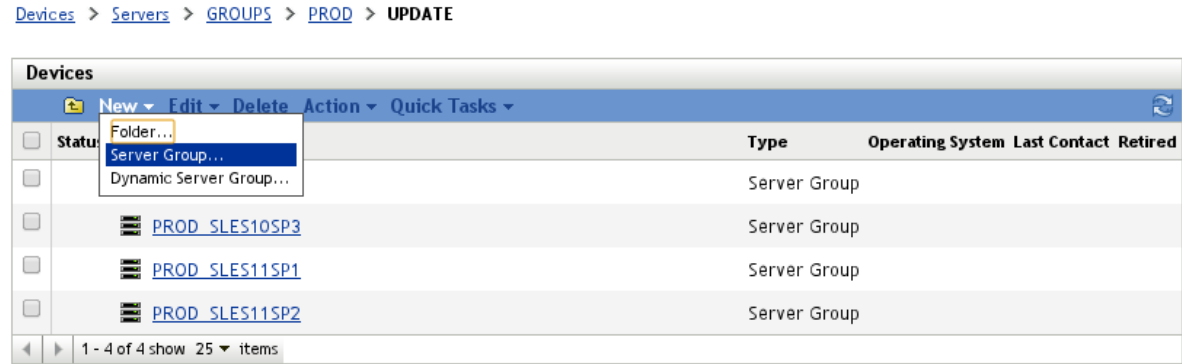
`%ENVIRONMENT%_%CONFIG_GROUP%`

Name	Description
ENVIRONMENT	DEV TEST PROD This part of a group name identifies the environment that is relevant for a configuration group.
CONFIG_GROUP	EDIR CLUST01 XEN ... This part of the key name identifies the function of the members of the configuration group.

Server Group Creation

The necessary steps to create a server group are shown in the following figures:

Figure 11-15 Creating a Server Group



The server group names in the following figure include environment, product, and service pack:


Figure 11-16 Server Group Creation - Basic Information

[Devices](#) > [Servers](#) > [GROUPS](#) > [PROD](#) > [UPDATE](#) > Create New Group

Create New Group

Step 1: Basic Information

Group Name: *

Folder: *
 

Description:

Members of this group get the bundle group containing the frozen update bundle for sles10-sp4 and the sles10-sp4 pool bundle assigned

Fields marked with an asterisk are required.

<< Back Next >> Cancel

There are no members added at this time. They will be added automatically via AutoYaST or they can be added manually in a later step.

Figure 11-17 Server Group Creation - Group Members List

[Devices](#) > [Servers](#) > [GROUPS](#) > [PROD](#) > [UPDATE](#) > Create New Group

Create New Group

PROD_SLES10SP4

Step 2: Add Group Members

Specify the members for this group:

Add Remove	
<input type="checkbox"/>	Name In Folder
No items selected, click add to select items	

<< Back Next >> Cancel

The following figure shows a summary of all selections made to this point:

Figure 11-18 Server Group Creation - Summary

[Devices](#) > [Servers](#) > [GROUPS](#) > [PROD](#) > [UPDATE](#) > **Create New Group**

Create New Group

PROD_SLES10SP4

Step 3: Summary

☐ Define Additional Properties

<< Back

Finish

Cancel

A message about the successful creation of the new server group appears in the final dialog box:

Figure 11-19 Server Group Creation - Success Message

Success: The Group has been created successfully.

[Devices](#) > [Servers](#) > [GROUPS](#) > [PROD](#) > **UPDATE**

Devices						
New Edit Delete Action Quick Tasks						
<input type="checkbox"/>	Status	Name		Type	Operating System	Last Contact Retired
<input type="checkbox"/>		PROD_OES11SP1		Server Group		
<input type="checkbox"/>		PROD_SLES10SP3		Server Group		
<input type="checkbox"/>		PROD_SLES10SP4		Server Group		
<input type="checkbox"/>		PROD_SLES11SP1		Server Group		
<input type="checkbox"/>		PROD_SLES11SP2		Server Group		

1 - 5 of 5 show 25 items

11.3.3 Device Registration

Device objects are automatically created during the initial registration or agent deployment. It is not possible to manually create any device object .

- ♦ “[Naming Standards for Device Objects](#)” on page 102
- ♦ “[Device Object Location](#)” on page 102

- ♦ [“Registration Rules” on page 102](#)
- ♦ [“Registration Keys” on page 103](#)

Naming Standards for Device Objects

Device objects are automatically named after their host name during registration. Although devices can be renamed and some prefixes or suffixes can be added, Novell Consulting recommends that you accept the default naming scheme.

Device Object Location

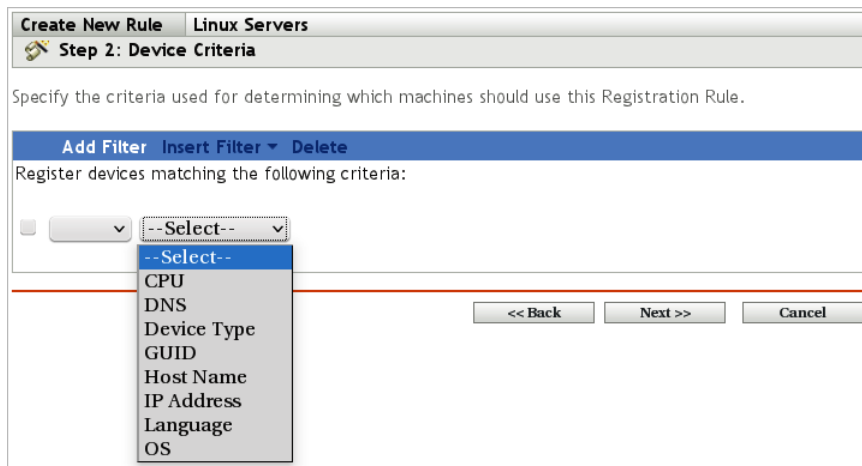
The device object location depends on how the device has been registered:

- ♦ The device was registered without a ZENworks Configuration Management key. If a registration without a key has been executed, the device object appears at the first level in the `/Devices/Servers` folder .
- ♦ The device was registered with a ZENworks Configuration Management key. In this case, the device object is placed into the folder that was defined when the key was created.
- ♦ The device was registered by one of the methods described above and moved manually to the destination folder.

Registration Rules

Registration rules provide the ability to register a device at a certain folder and with certain groups without specifying a key. Device location and group assignments can be determined by filters instead of by keys.

Figure 11-20 Registration Rules - Filters



However, the provided filters are not specific enough to achieve the same goals as the registration keys introduced by Novell Consulting (see [“Registration Keys” on page 103](#)). Therefore, Novell Consulting has decided not to use registration rules, pending further investigation.

Registration Keys

Registration keys play an important role in organizing where Linux devices are placed within ZENworks Configuration Management.

The first and mandatory purpose of a key is to place a device object during its first registration into a certain device folder. If a key is not specified, the device object is created in the `Devices/Servers` folder.

Although device objects can be manually moved within the ZENworks Configuration Management folder structure, Novell Consulting recommends that you use registration keys to determine the final location during initial registration. This type of key is called a “location key.” This assignment method allows for easier administration and supports automatic registration via AutoYaST during installation.

The second and optional purpose of a registration key is to subscribe the device to one or more server groups. This must be configured during key creation or by later editing a key object to determine the device group or device groups the device is assigned to.

Novell Consulting recommends that you distinguish between keys intended to determine the target folder of a device object and keys intended to register the device object to device groups. This simplifies administering and maintaining objects.

- ♦ [“Naming Standards for Registration Keys” on page 103](#)
- ♦ [“ZCM Device Group Registration Keys” on page 104](#)
- ♦ [“Folders in the Registration Keys Menu” on page 104](#)
- ♦ [“Creating a Registration Key” on page 106](#)

Naming Standards for Registration Keys

To avoid any confusion, you must be able to clearly identify the purpose for a key. Using the Novell Consulting naming scheme helps you to accomplish this.

Common Rules: The first component (`%ENVIRONMENT%`) applies to all ZCM keys and identifies the environment in which the managed device is operated:

Name	Description
DEV	Development system
TEST	Test system
PROD	Production system

ZCM Location Keys: Novell Consulting recommends that you restrict the ZENworks Configuration Management location keys to the initial registration of managed devices. The naming scheme for location keys is defined as follows:

`%ENVIRONMENT%_%LOCATION%_%FUNCTION%`

Name	Description
LOCATION	<p>LOC1 LOC2 LOC3 ...</p> <p>This part of the naming standard identifies the physical location of a managed device. This can be a room, a building, a branch office, or even a city. Each of these locations is represented by its own subfolder under the device menu of ZENworks Configuration Management.</p>
FUNCTION	<p>EDIR CLUSTER01 ...</p> <p>This part of a name indicates the primary function of a managed device, such as an eDirectory server or a member of a certain NCS cluster.</p> <p>This part of the key name also defines the name of the subfolder in the devices menu where the objects for the devices will be placed.</p> <p>In well-designed OES environments, this folder name typically matches the name of the corresponding OU in the eDirectory structure where the NCP Server objects for these devices are located.</p>

ZCM Device Group Registration Keys

Novell Consulting suggests that you use the same naming standards for update and configuration group registration keys as defined for server groups in [“Naming Standards for Server Groups” on page 98](#). This results in an easy-to-understand relationship between registration keys and the server groups to which they add the server objects.

Folders in the Registration Keys Menu

Novell Consulting recommends the folder structure as shown in the following table. Level 2 folders must be created underneath each Level 1 folder. The structure is tightly related to the different key purposes as discussed earlier.

ZCM Key Folder for Novell Consulting

Name	Folder Level	Description
DEV	1	Keys that are used within development environments
TEST	1	Keys that are used within test environments
PROD	1	Keys that are used within production environments
CONFIG	2	Keys that have configuration purposes and are used only for group assignment
UPDATE	2	Keys that have update/patch purposes and are used only for group assignment
LOCATION	2	Keys that are used only for initial registration at the ZENworks Configuration Management server; they determine where the appropriate devices object is placed

The folder structure for registration keys is illustrated in the following figures:

Figure 11-21 Registration Keys Menu – Level 1

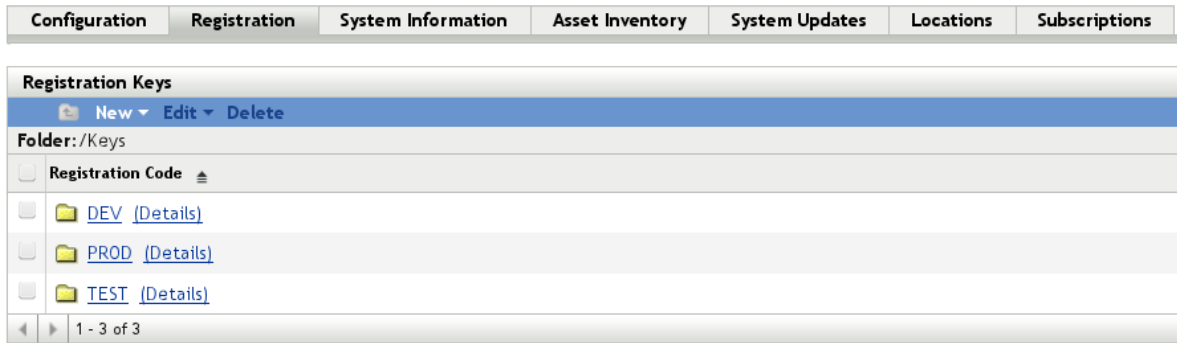
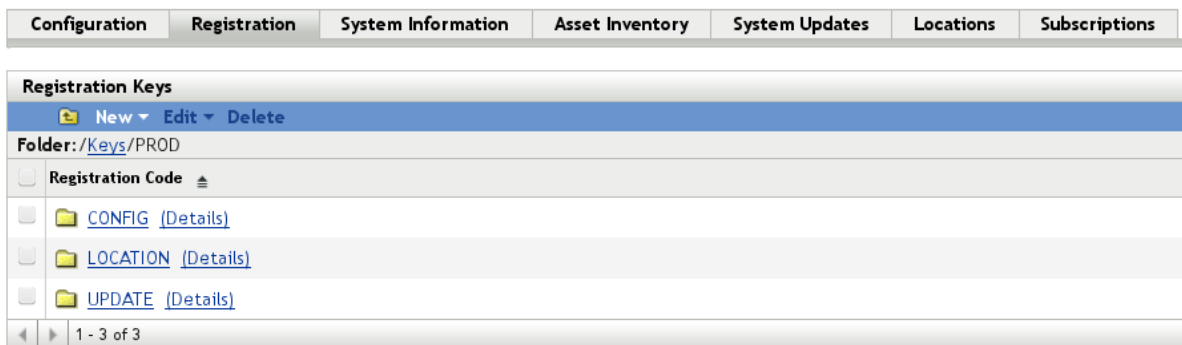


Figure 11-22 Registration Keys Menu – Level 2



The DEV and TEST folders have the same structure as displayed for the PROD folder in Figure 11-22.

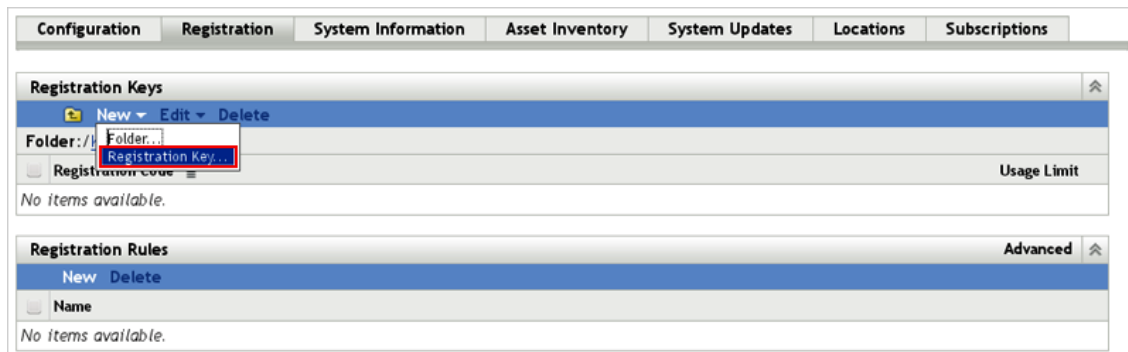
The following table lists some possible registration keys that are based on the naming schema outlined in “Naming Standards for Registration Keys” on page 103:

Name	Role	Key Folder
PROD_DUS_EDIR	Location Key	/Keys/PROD/LOCATION
DEV_DUS_DUSCL01	Location Key	/Keys/DEV/LOCATION
PROD_SLES11SP1	Update Group Key	/Keys/PROD/UPDATE
TEST_OES11GA	Update Group Key	/Keys/TEST/UPDATE
PROD_EDIR	Configuration Group Key	/Keys/PROD/CONFIG
TEST_XEN	Configuration Group Key	/Keys/TEST/CONFIG

Creating a Registration Key

Select *Registration Key* from the *New* menu.

Figure 11-23 Key Creation



Name the key based on the naming scheme described in [“Naming Standards for Device Objects”](#) on [page 102](#). Select the appropriate folder to store the registration key and enter a description explaining the purpose of the key as illustrated below.

Don't press the *Generate* button during this step of key creation. Doing this overwrites the inserted key name with a self-generated key name.

Figure 11-24 Key Creation - Basic Information

The screenshot shows a dialog box titled 'Create New Registration Key' with a sub-header 'PROD_EDIR'. Below the title bar, there is a section titled 'Step 1: Basic Information'. The main text in this section reads: 'Supply the name, description, and the limit for the new registration key. A unique name can be generated by clicking on the "Generate" button.' Below this text, there are three main input areas. The first is 'Key Code: *' with a text input field containing 'PROD_EDIR' and a 'Generate' button to its right, which is highlighted with a red rectangle. The second is 'Folder: *' with a text input field containing '/Keys/PROD/CONFIG' and a folder selection icon to its right. The third is 'Description:' with a text area containing the text: 'For group assignment only. Registers a device as a member to Servers/GROUPS/PROD/CONFIG /PROD_EDIR'. Below the description area, there is a section titled 'Number of times this key can be used:' with two radio buttons: 'Unlimited' (which is selected) and 'Limit to:' followed by a text input field. At the bottom of the dialog box, there is a red line and a note: '* Fields marked with an asterisk are required.' Below this note, there are three buttons: '<< Back', 'Next >>', and 'Cancel'.

For update keys or configuration keys a device folder makes no sense, because these ZENworks Configuration Management keys should never be used for the initial registration of a device. However, because this step is required, devices that are accidentally registered with an update or configuration key are placed in the `/Devices/Servers/NOKEY` folder.

If a device object is found in the `NOKEY` folder, administrators know that a wrong registration key was used during initial registration of the device.

Figure 11-25 Key Creation - Device Folder

Create New Registration Key **PROD_EDIR**

Step 2: Device Folder

Supply the folder the machine should be placed in when registered.

Folder:

The information in the next three fields is not required for update and configuration keys. The fields can be left empty:

Figure 11-26 Key Creation - Device Fields

Create New Registration Key **PROD_EDIR**

Step 3: Device Fields

Supply the values new machines should receive when registered.

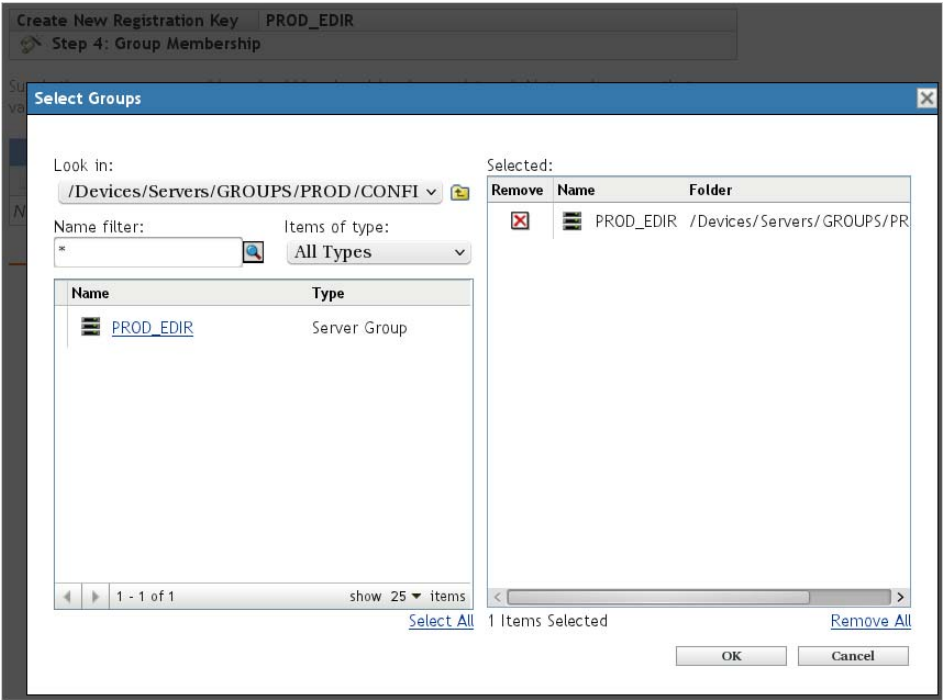
Department:

Site:

Location:

The next dialog box requires you to choose one or more groups for the device to become a member of if this key is used:

Figure 11-27 Key Creation - Group Membership



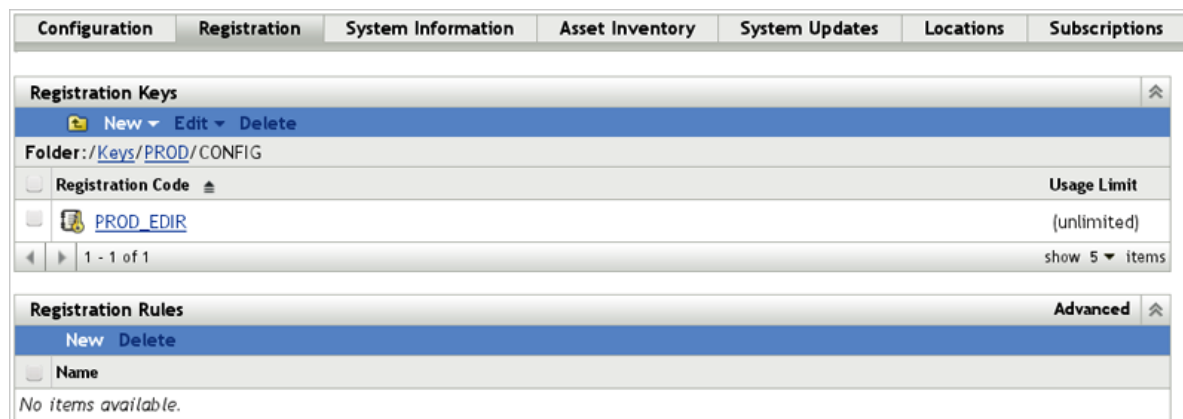
Finally, all selections are summarized:

Figure 11-28 Key Creation - Summary



The newly created key can be validated, edited, or deleted in the *Configuration > Registration > Registration Keys > PROD > CONFIG* folder:

Figure 11-29 New Key - Ready to Use



11.4 Bundles

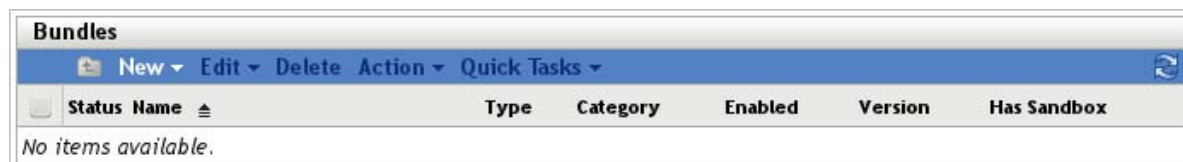
The Bundles menu is at least as important as the device menu. Objects that represent software items such as RPMs, patches, archives, and configuration files are located here.

- ♦ Section 11.4.1, “Folders,” on page 109
- ♦ Section 11.4.2, “Bundles and Bundle Groups,” on page 113
- ♦ Section 11.4.3, “Creating a File Bundle,” on page 118
- ♦ Section 11.4.4, “Bundle Groups and Bundles Developed by Novell Consulting,” on page 125

11.4.1 Folders

The bundles menu is empty for a newly installed ZENworks Configuration Management server.

Figure 11-30 Bundle Folder after Installation



In general, two different kinds of software appear under this menu:

- ♦ For every operating system and service pack level provided by vendors such as Novell or Red Hat, folders and bundle objects are automatically created here by the mirror process

These objects are used by Novell Consulting to build an update methodology based on reliable frozen patch bundles, which are described later in [Section 12.1, “Frozen Patch Level,” on page 155](#).

- ♦ Any kind of in-house software or in-house task object is located below this menu. These objects can be used to configure the target devices or to deploy software that was obtained by other means than mirroring distribution channels.
- ♦ [“Folder Hierarchy” on page 110](#)
- ♦ [“Naming Standards for Bundle Folders with Mirror Content” on page 110](#)
- ♦ [“The CUSTOM Folder” on page 112](#)

Folder Hierarchy

Even if some objects are created automatically, the folder structure underneath the *Bundles* menu must be predefined.

Because this part of the ZENworks Configuration Management system will hold a large number of entries, the folder structure must be well designed to allow for easy administration:

- ♦ Deep folder hierarchies should be avoided. Novell Consulting recommends that you design a hierarchy with a maximum of five levels.
- ♦ The first level of the folder hierarchy starts with `Linux` to distinguish between software bundles for products of other vendors such as Apple or Microsoft and software items for Linux products.

Figure 11-31 Bundle Folder Level 1

Status	Name	Type	Category	Enabled	Version	Has Sandbox
	Linux (Details)	Folder				
	MAC (Details)	Folder				
	Windows (Details)	Folder				

- ♦ The second level of the hierarchy is made up of the `CUSTOM` folder for customer-specific bundles and folders based on operating system versions or add-on product versions.
- ♦ With the exception of `CUSTOM`, each of the previously mentioned Level 2 folders is subdivided into folders for the corresponding support packs at the third level.
- ♦ Folders below Level 3 are automatically created by the mirror process.

Naming Standards for Bundle Folders with Mirror Content

In addition to the general naming rules formulated in [Section 11.2.1, “General Rules for Folder Names,” on page 90](#), the following additional definitions are required:

- ♦ Names of folders on Level 2 are built from the product tag `%PRODUCT%` and the version tag `%VERSION%`. For example, `SLES11`.
- ♦ Folders on Level 3 are named either after the corresponding service pack or the term “GA” if no service pack exists. For example, `SP1 | SP2 | GA | ...`

The following figure illustrates the folder structure for bundles at folder Level 2 with subfolders GA and SP1 at Level 3:

Figure 11-32 Bundle Folder Levels 2 and 3

[Bundles](#) > [Linux](#) > **SLES11**

Bundles						
New ▾ Edit ▾ Delete Action ▾ Quick Tasks ▾						
<input type="checkbox"/>	Status	Name	Type	Category	Enabled	Has Sandbox
<input type="checkbox"/>		GA (Details)	Folder			
<input type="checkbox"/>		SP1 (Details)	Folder			

1 - 2 of 2 show 25 ▾ items

Folders in the following figure are created automatically by the mirror process. Details about the configuration of mirror channels are explained [Section 11.5.4, “Creating a Subscription,” on page 146](#).

Figure 11-33 Bundle Folders Created by the Mirror Process - Level 4

[Bundles](#) > [Linux](#) > [SLES11](#) > **SP1**

Bundles						
New ▾ Edit ▾ Delete Action ▾ Quick Tasks ▾						
<input type="checkbox"/>	Status	Name	Type	Category	Enabled	Has Sandbox
<input type="checkbox"/>		SLE11-HAE-SP1-Pool (Details)	Folder			
<input type="checkbox"/>		SLE11-HAE-SP1-Updates (Details)	Folder			
<input type="checkbox"/>		SLES11-SP1-Pool (Details)	Folder			
<input type="checkbox"/>		SLES11-SP1-Updates (Details)	Folder			

1 - 4 of 4 show 25 ▾ items

In the following figure, you can see the bundles at the lowest level originally mirrored from Novell:

Figure 11-34 Bundles at Folder Level 5

[Bundles](#) > [LINUX](#) > [SLES11](#) > [SP1](#) > **SLES11-SP1-Updates**

Bundles						
New ▾ Edit ▾ Delete Action ▾ Quick Tasks ▾						
<input type="checkbox"/>	Status	Name	Type	Category	Enabled	Has Sandbox
<input type="checkbox"/>		SLES11-SP1-Updates-bundle	Linux Bundle		Yes	No

1 - 1 of 1 show 25 ▾ items

The CUSTOM Folder

The `CUSTOM` folder serves as a container for all in-house software and special patches retrieved from Novell (PTF, FT, engineering builds). It contains several subcontainers for additional subdivision of bundles based on their operating system, patch level, and other criteria.

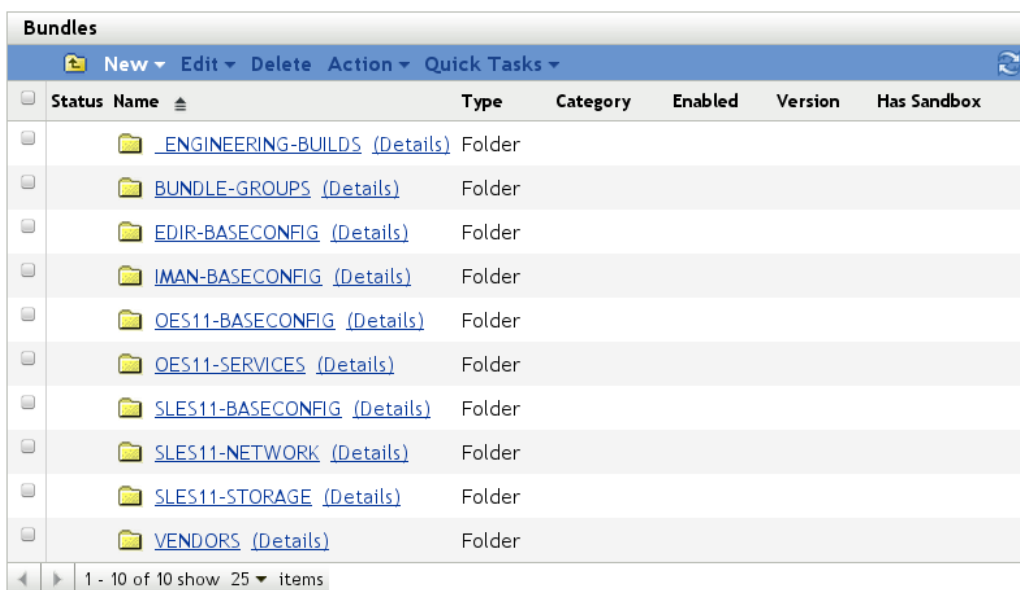
In addition to the general naming rules defined in [Section 11.2.1, “General Rules for Folder Names,” on page 90](#), the following rules apply to folders in the `CUSTOM` branch of the `Bundles` folder structure:

- ♦ The top level folder is named `CUSTOM`
- ♦ Folders below `CUSTOM` are named to symbolize the task the folder was created for
- ♦ Folders prefixed with `SLES11` contain software for SLES devices that is also applicable to OES 11 system
- ♦ Folders prefixed with `OES11` contain only software for OES 11 devices

The following figure shows the structure of the `CUSTOM` folder created during a recent project:

Figure 11-35 Bundle Folder `CUSTOM`

[Bundles](#) > [Linux](#) > `CUSTOM`



The screenshot shows a web interface for managing bundles. The breadcrumb path is Bundles > Linux > CUSTOM. Below the path is a table listing the contents of the CUSTOM folder. The table has columns for Status, Name, Type, Category, Enabled, Version, and Has Sandbox. There are 10 rows, each representing a folder: _ENGINEERING-BUILDS, BUNDLE-GROUPS, EDIR-BASECONFIG, IMAN-BASECONFIG, OES11-BASECONFIG, OES11-SERVICES, SLES11-BASECONFIG, SLES11-NETWORK, SLES11-STORAGE, and VENDORS. Each row has a folder icon and a link to its details page. At the bottom, there is a pagination bar showing '1 - 10 of 10 show 25 items'.

Status	Name	Type	Category	Enabled	Version	Has Sandbox
	_ENGINEERING-BUILDS (Details)	Folder				
	BUNDLE-GROUPS (Details)	Folder				
	EDIR-BASECONFIG (Details)	Folder				
	IMAN-BASECONFIG (Details)	Folder				
	OES11-BASECONFIG (Details)	Folder				
	OES11-SERVICES (Details)	Folder				
	SLES11-BASECONFIG (Details)	Folder				
	SLES11-NETWORK (Details)	Folder				
	SLES11-STORAGE (Details)	Folder				
	VENDORS (Details)	Folder				

The purpose of the individual folders is as follows:

- ♦ `_ENGINEERING-BUILDS` contains field test files (FTF) that are not available from update channels. These bundles are assigned to individual device objects or to device groups as needed. They must not be assigned automatically because they might be obsolete with the next patch level.
- ♦ `BUNDLE-GROUPS` contains all bundle groups that are used for easier bundle assignment.
- ♦ `EDIR-BASECONFIG` stores any bundles that configure eDirectory.
- ♦ `IMAN-BASECONFIG` contains bundles that affect the configuration of iManager.
- ♦ `OES11-BASECONFIG` contains bundles that are suitable for all OES 11 devices.
- ♦ `OES11-SERVICES` contains bundles with configuration settings for OES 11 services such as NOV-FTP, LUM, and NSS.

- ♦ `SLES11-BASECONFIG` contains bundles with configuration settings for SLES 10 or SLES 11 components, regardless of whether OES 11 is installed as an add-on product.
- ♦ `SLES11-NETWORK` contains bundles that execute network-related configurations for SLES 11.
- ♦ `SLES11-STORAGE` contains bundles located that configure properties of HBA hardware, volume managers, or multi-pathing for SLES 11.
- ♦ `VENDORS` contains vendor-specific software such as hardware tools, virus scanners, backup software, and others from vendors such as IBM, HP, or Symantec. It has been introduced as a separate subfolder because possible vendor folders below the `CUSTOM` folder would destroy the ordering within the UI, because it is sorted alphabetically.

11.4.2 Bundles and Bundle Groups

In addition to subfolders, the most important objects in this folder are the bundles themselves, representing packages, configuration files, archives, and patches.

Several different types of bundle objects exist in the name space of ZENworks Configuration Management. They can be classified into the following categories:

- ♦ Bundles created by a mirror process from external sources
 - ♦ Update bundles
 - ♦ Pool/Core bundles/dependency bundles
 - ♦ Online bundles
 - ♦ Bundles from external repositories
- ♦ In-house bundles created via ZENworks Control Center or the `zman` command line utility
 - ♦ File bundles
 - ♦ RPM bundles
 - ♦ Empty bundles

An additional object type within the bundles menu is the bundle group.

- ♦ [“Update Bundles” on page 113](#)
- ♦ [“Pool Bundles” on page 114](#)
- ♦ [“Core Bundle” on page 114](#)
- ♦ [“Online Bundles” on page 114](#)
- ♦ [“File Bundles” on page 114](#)
- ♦ [“RPM Bundles” on page 115](#)
- ♦ [“Empty Bundles” on page 115](#)
- ♦ [“Bundle Groups” on page 115](#)

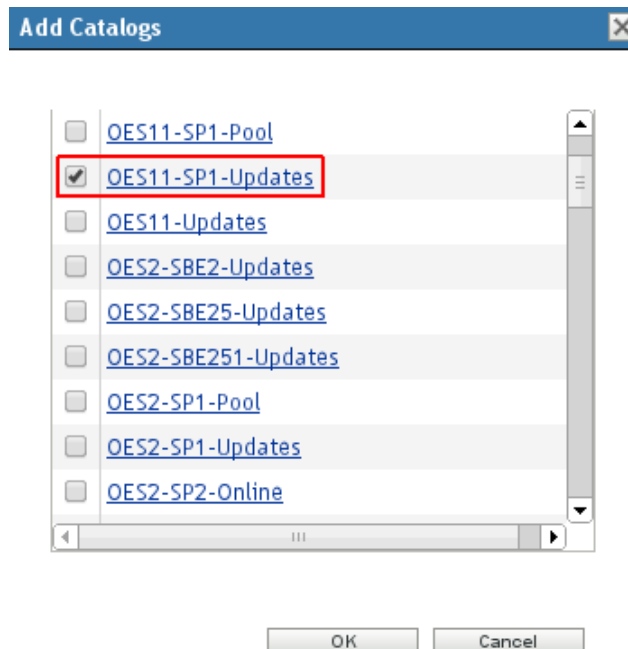
Update Bundles

Vendors like Novell or Red Hat publish their patches via special update repositories. The contents of these repositories can be accessed by the device needing the updates directly or by a mirror server such as a ZENworks Configuration Management server that can download the updates so that devices on the corporate network have access to these updates locally.

The notion of an update bundle is used for a set of patches or packages that is targeted at a given OS level. Update bundles usually contain several RPMs or delta RPMs.

Updates bundles are also provided by the Novell update server `nu.novell.com`.

Figure 11-36 Bundle Types Provided by Novell



Pool Bundles

A pool bundle contains the same content as the installation medium of the corresponding product and service pack. It is primarily used for dependency resolution.

Core Bundle

This bundle type was introduced with SLES 11 SP2 as part of a recent change to the SUSE Linux Enterprise 11 Maintenance Model (see [Section 11.1, “Overview,” on page 89](#)). It is simply a subset of the installation media that contains approximately 30 per cent of the original Pool channel, considered as the core of the new Service Pack. Core bundles are also used for dependency resolution. Packages that are not provided by a Core bundle are still provided by the earlier Pool bundle. This type of bundle is only used by SLES 11 but not by earlier SLES versions or by OES 11.

Online Bundles

An online bundle is a subtype of a pool bundle. It is used only when upgrading an operating system or an add-on product to the next release, such as upgrading from OES 2 SP2 to OES 2 SP3.

File Bundles

File bundles are typically used to distribute configuration files or archives from other vendors (McAfee virus scanner, HP Proliant Utilities, IBM Tivoli backup software, and so forth) if they are not provided as RPM format.

RPM Bundles

RPM bundles contain one or more RPM packages. The bundle can be made up of packages from the ZENworks Configuration Management repository, of external packages uploaded via the upload dialog box, or of a mixture from both sources.

Empty Bundles

At first glance, an empty bundle seems to make no sense. However, it is not really empty, but it has been created as a placeholder to be filled later with other bundle types. An empty bundle can play an important role when you need to group different bundles in a specific order. An empty bundle can serve as a container for other bundles. This lets you put different bundles in order for easier assignment, controlling the installation order, and applying properties such as filters to the whole set.

In addition, an empty bundle is often used for scripted deployments, where it serves as the carrier for a script that is executed at the target devices without distributing the script content to the devices.

The following figure illustrates different bundle types discussed earlier. OES11ALL_NOV-AUDIT-OFF and OES11GA_NOV-DEMO-EMPTYBDL are examples of empty bundles. The type of the remaining two bundles is displayed in the *Category* field.

Figure 11-37 Bundle Types

[Bundles](#) > [Linux](#) > [CUSTOM](#) > OES11-BASECONFIG

Bundles						
New Edit Delete Action Quick Tasks						
Status	Name	Type	Category	Enabled	Version	Has Sandbox
	OES11ALL_NOV-AUDIT-OFF	Linux Bundle		Yes	1	No
	OES11ALL_NOV-BASHRC	Linux Bundle	Install File(s)	Yes	0	Yes
	OES11GA_NOV-DEMO-EMPTYBDL	Linux Bundle		Yes	0	No
	OES11GA_NOV-DEMO-RPM	Linux Bundle	RPM Application	Yes	0	No
1 - 4 of 4						show 25 items

Bundle Groups

Bundle groups combine multiple bundles into one group object similar to empty bundles. By assigning bundle groups rather than individual bundles to devices, the number of required assignments can be reduced considerably. A further reduction in assignments can be achieved when bundle groups are

assigned to device groups. Examples for bundle groups used by Novell Consulting are explained in the following figure. Their importance is explained in [Section 11.4.4, “Bundle Groups and Bundles Developed by Novell Consulting,”](#) on page 125.

Figure 11-38 Bundle Groups Example

[Bundles](#) > [Linux](#) > [CUSTOM](#) > **BUNDLE-GROUPS**

Bundles							
New ▾ Edit ▾ Delete Action ▾ Quick Tasks ▾							
	Status	Name	Type	Category	Enabled	Version	Has Sandbox
		OES11SP1-DEV-LOGIN-GRP	Bundle Group				
		OES11SP1-PROD-DUSCL01-GRP	Bundle Group				
		OES11SP1-PROD-LOGIN-GRP	Bundle Group				
		OES11SP1-TEST-LOGIN-GRP	Bundle Group				
1 - 4 of 4 show 25 ▾ items							

Naming Standards for Bundles and Bundle Groups

- ♦ [“Naming Standards for Bundles”](#) on page 116
- ♦ [“Naming Standards for Bundle Groups”](#) on page 117

Naming Standards for Bundles

Novell Consulting has developed and applied the following naming standard for bundle objects located below the CUSTOM folder:

`%PRODUCT% [%VERSION% [SP%SP-LEVEL%]] _%INITIATOR%-%PURPOSE% [-%TARGET%]`

Name	Description
PRODUCT	<p>SLES OES EDIR IMAN NCS ...</p> <p>This part of a name identifies the product for which a certain bundle is relevant. Objects with names that start with SLES must be assigned to pure SLES systems (such as SLES 10) as well as to the corresponding OES system (such as OES 2).</p> <p>This part of the name is also used to identify if an object is only relevant for specific systems such as an eDirectory server (SLES or OES), a server with iManager installed (SLES or OES), or an NCS cluster.</p>
VERSION	<p>2 9 10 11 [only] ALL</p> <p>This part of the naming standard identifies for which version (SLES or OES) an object is applicable. If an object is only applicable for a certain support pack level, this is denoted by <i>only</i>. If it is not specific for a particular version, <i>ALL</i> is used instead.</p>
SP-LEVEL	<p>1 2 3 [only] ALL</p> <p>This becomes part of an object name only if an object is specific for a certain support pack level and later. If an object is only applicable for a specific support pack level, it is indicated by <i>only</i>.</p> <p>If a bundle is not specific to a support pack, the SP level is replaced with <i>ALL</i>.</p>

Name	Description
INITIATOR	NOV CUST ENG Bundles that were initiated by Novell and that are not specific to a certain customer are identified by NOV. CUST indicates that an object is specific for the customer environment. ENG indicates an engineering build. Replace CUST with an identifier for the customer (maximum 4 characters).
PURPOSE	NAM FTP ... This part of the name very briefly indicates the purpose of the object or the component being configured.
TARGET	DUSCL01 PRV ... This is an optional part of the naming standard that is used only if an object is specific to certain systems such as the nodes of an NCS cluster or the systems in a specific location.

Naming Standards for Bundle Groups

Similar rules apply for bundle groups:

%PRODUCT%[%VERSION%[SP%SP-LEVEL%]]-%ENVIRONMENT%[-%Target%]-GRP

Name	Description
PRODUCT	SLES OES Equivalent to bundles
VERSION	2 9 10 11 [only] ALL Equivalent to bundles
SP-LEVEL	1 2 3 [only] ALL Equivalent to bundles
ENVIRONMENT	DEV - Development system PROD - Production system TEST - System from a technical proof of concept
TARGET	Equivalent to bundles

These naming standards are case-sensitive. The separators “_” and “-” also are a relevant part of the naming standard.

Some examples of how to use these naming standards to compose names for bundles or bundle groups are listed in the following table:

Name	Description
OE11GA_CUST-NAM-xyzcl01	The bundle configures NAM on the OES11GA nodes of cluster xyzcl01.
SLESALL_NOV-SUPPORT-CONFIG	The bundle contains the support-config packages from Novell for all SLES versions and service packs.
OES11GA_ENG_IFOLDER-SETTINGS	The bundle provides an FTF (field test file) obtained from engineering with specific settings for Novell iFolder.
IMAN_NOV-J2EE	The bundle contains configuration settings for iManager /etc/sysconfig/j2ee.
OES11GA-PROD-DUS01-GRP	A bundle group that deploys bundles to configure settings that are specific for cluster dusl01).

11.4.3 Creating a File Bundle

The main purpose of file bundles is to distribute simple files or archives. The creation of a file bundle starts in the appropriate subfolder below Bundles > Linux > CUSTOM. The following example shows the creation of a file bundle that configures a bond interface on SLES 11. The bundle is described in detail in [“SLES11ALL_CUST-BONDING” on page 138](#).

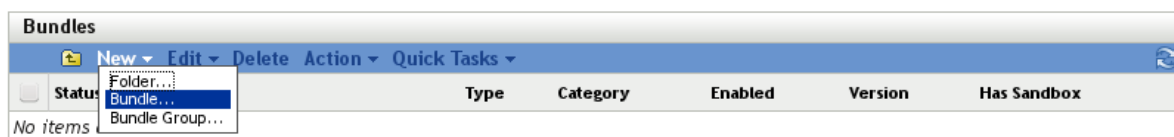
Following the naming standard defined in [“Naming Standards for Bundles and Bundle Groups” on page 116](#), the bundle is called SLES11ALL_CUST-BONDING for the following reasons:

- ♦ SLES11ALL indicates that the bundle is intended for SLES 11 independent of the service pack version
- ♦ _CUST indicates that the bundle is using some customer-specific information, such as the target IP address for link monitoring.
- ♦ -BONDING indicates that the purpose of the bundle is to configure a bond device.

The bundle has been categorized as a network-related bundle and is therefore created below the Bundles > Linux > CUSTOM > SLES11-NETWORK folder as shown in the following figure:

Figure 11-39 Bundle Creation

[Bundles](#) > [LINUX](#) > [CUSTOM](#) > [SLES11-NETWORK](#)



The Linux Bundle entry is selected according to the default:

Figure 11-40 Bundle Creation - Select Bundle Type

[Bundles](#) > [Linux](#) > [CUSTOM](#) > [SLES11-NETWORK](#) > [Create New Bundle](#)

Create New Bundle
Step 1: Select Bundle type

Select the type of Bundle you wish to create from the list of options.

New Bundle Type:

- Linux Bundle
- Linux Dependency Bundle
- Macintosh Bundle
- Preboot Bundle
- Windows Bundle

Description:

Linux Bundle - This allows you to configure and manage applications on Linux devices.

<< Back Next >> Cancel

Because the bundle distributes a script and a binary that are not part of any RPM, you need to select the *Install File(s)* bundle category:

Figure 11-41 Bundle Creation - Select Bundle Category

[Bundles](#) > [Linux](#) > [CUSTOM](#) > [SLES11-NETWORK](#) > [Create New Linux Bundle](#)

Create New Linux Bundle
Step 2: Select Bundle category

Select the category of Bundle you wish to create from the list of options.

New Bundle Category:

- (Empty Bundle)
- Create/Delete Directory
- Install Directory
- Install File(s)
- RPM Application

Description:

Install File(s) - Uploads selected files to the ZENworks content system and then installs them to the destination path on the managed device. The content (by default) will be replicated to all primary servers.

<< Back Next >> Cancel

The bundle name is assigned in the next step. If the *Icon* box is left empty, a standard icon is displayed near the bundle name:

Figure 11-42 Bundle Creation - Define Details

Bundles > Linux > CUSTOM > SLES11-NETWORK > Create New Linux Bundle

Create New Linux Bundle

Step 3: Define Details

Enter the details for the bundle.

Bundle Name: *

SLES11ALL_CUST_BONDING

Folder: *

/Bundles/Linux/CUSTOM/SLES11-NETWORK

Icon:

Description:

This bundle creates a bond device bond0 out of interfaces eth0 and eth1

Fields marked with an asterisk are required.

<< Back Next >> Cancel

Next opens the file upload dialog box, where one or more files can be uploaded into ZENworks Configuration Management:

Figure 11-43 Bundle Creation - Select Files

Bundles > Linux > CUSTOM > SLES11-NETWORK > Create New Linux Bundle

Create New Linux Bundle SLES11ALL_CUST_BONDING

Step 4: Select Files

Select the file(s) to be installed.

File: *

File

File path: /tmp/projects/bonding/ayast_ Durchsuchen...

File Upload extension not installed...file uploads are limited to 1 GB.

OK Cancel

[Install the Novell File Upload extension to upload multiple files](#)

A security feature in Firefox 4 and above might prevent you from installing Novell File Upload extension. For information on how to install Novell File Upload extension on such browsers, [click here](#).

Novell File Upload Extension: The extension displayed in the Bundle Creation Step 4 allows you to upload multiple files and to upload files larger than 1 GB. However, you will have a problem when installing the extension on Mozilla Firefox if the SSL certificate of the hosting server is signed as “Certificate Authority unknown to the browser.” To resolve this, use the [click here](#) button and follow the instructions. In addition, the extension does not work on newer versions of Mozilla Firefox.

The next figure shows the configuration of the destination directory, the file permissions, and the file ownership:

Figure 11-44 Bundle Creation - Destination Directory, File Permissions, and File Ownership

[Bundles](#) > [Linux](#) > [CUSTOM](#) > [SLES11-NETWORK](#) > [Create New Linux Bundle](#)

Create New Linux Bundle SLES11ALL_CUST_BONDING

Step 4: Select Files

Select the file(s) to be installed.

File: *

File	Size	Packaging
ayast_setup.ycp	3 KB	Auto
config_bond.sh	13 KB	Auto

Add

Clear

The content will be uploaded to the server and then installed from the server when the bundle is launched on the managed device.

Destination Directory: *

/usr/local/zac

The path must be resolved by the device on which the bundle is run.

Permissions

☒ mode: (Enter three digit Octal numbers e.g. 675)

☐ Owner: ☒ Read ☒ Write ☐ Execute

☐ Group: ☒ Read ☐ Write ☐ Execute

☐ Others: ☒ Read ☐ Write ☐ Execute

Ownership

☒ root

☐ user *

Group:

☐ Unpack

☐ Delete After Unpack

Copy Option:

Copy Always

Fields marked with an asterisk are required.

<< Back

Next >>

Cancel

The last step in the creation process sums up all the selections made to this point:

Figure 11-45 Bundle Creation - Summary

[Bundles](#) > [Linux](#) > [CUSTOM](#) > [SLES11-NETWORK](#) > Create New Linux Bundle

Create New Linux Bundle

SLES11ALL_CUST_BONDING

Step 5: Summary

Review the information and click "Finish" to create the new bundle.

Name: SLES11ALL_CUST_BONDING
Icon:
Folder: SLES11-NETWORK
Description: This bundle creates a bond device bond0 out of interfaces eth0 and eth1

☐ Create as Sandbox
☐ Define Additional Properties

<< Back

Finish

Cancel

The bundle has been created and is almost ready for use:

Figure 11-46 Bundle Creation Success

Success: The Bundle has been created successfully.

[Bundles](#) > [Linux](#) > [CUSTOM](#) > [SLES11-NETWORK](#)

Bundles							
New Edit Delete Action Quick Tasks							
Status	Name	Type	Category	Enabled	Version	Has Sandbox	
	SLES11ALL_CUST_BONDING	Linux Bundle	Install File(s)	Yes	0	No	

1 - 1 of 1 show 25 items

Before the new bundle can be used, you will need to make a few modifications. Because an unintended redeployment of this bundle would almost certainly result in a service interruption, you need to ensure that it will only be deployed to devices that do not have a bond device already configured.

This can be done by adding a requirement to the bundle specifying that the configuration file for the bond0 device must not exist in `/etc/sysconfig/network`. If you ever want to reconfigure your bond0 device, ensure that you delete its configuration file before you reinstall the bundle.

Figure 11-47 Bundle Modification - System Requirements

Bundles > Linux > CUSTOM > SLES11-NETWORK > SLES11ALL_CUST_BONDING

SLES11ALL_CUST_BONDING

Displayed Version: 0 (Published)

Summary Relationships **Requirements** Actions Packages Settings

System Requirements

Specify the system requirements for this bundle and indicate if the application icon should be displayed to the user.

☐ Show application icon if system requirements fail

Add Filter Add Filter Set Insert Filter ▾ Delete

Combine Filters using: and ▾

The following conditions should be effective for this object to be enforced:

☐ File Exists ▾ etwork/ifcfg-bond0 No ▾

Apply Reset

By default, ZENworks Configuration Management does not have the uninstall task enabled. Novell Consulting recommends that you enable this feature under the *Options* link of the Uninstall action.

Figure 11-48 Bundle Modification - Enable Uninstall

Bundles > Linux > CUSTOM > SLES11-NETWORK > SLES11ALL_CUST_BONDING@Sandbox

SLES11ALL_CUST_BONDING@Sandbox

Displayed Version: Sandbox Publish Revert

Summary Relationships Requirements **Actions** Packages Settings

Distribute Install Launch Verify **Uninstall**

Add ▾ Remove Move Up Move Down Enable Disable Options

Name	Type	State	Continue on Failure
Undo Install Actions	Undo Install Actions	Enabled	<input type="checkbox"/>

1 - 1 of 1

Uninstall is disabled. Click 'Options' to enable

Apply Reset

Uninstall Options

☒ Enable Uninstall

☐ Allow user to perform uninstall

Blocked Assignment Options

☐ Uninstall application

OK Cancel

Finally, you need to allow the `config_bond.sh` script contained in this bundle to be executed by the root user; otherwise, the bundle will never install.

To do this, you need to double-click the script in the Install action and go to the *Advanced* tab, then select *Run as Root*. This modification is required with every ZENworks Configuration Management bundle that contains a custom script.

We also recommend that you change the action name to something specific for your bundle, such as `run_config-bond` rather than the default `Run Script` action name

Figure 11-49 Bundle Modification - Run As root

The screenshot shows the 'Add Action - Run Script' dialog box with the 'Advanced' tab selected. The 'Action Name' field contains 'run_config-bond'. The 'Executable security level' section has 'Run as root' selected. The 'OK' and 'Cancel' buttons are at the bottom right.

Fields marked with an asterisk are required.

Changes always create a sandbox version of a bundle. This version must be published to become available for all devices to which it is assigned.

Figure 11-50 Bundle Modification Completed

The screenshot shows the bundle modification interface for 'SLES11ALL_CUST_BONDING@Sandbox'. The 'Displayed Version' dropdown is set to 'Sandbox'. Below the tabs, a table lists the actions: 'config_bond' (Install File(s)) and 'run_config-bond' (Run Script), both with a state of 'Enabled'.

Distribute		Install		Launch		Verify		Uninstall	
Add	Remove	Move Up	Move Down	Enable	Disable	Options			
Name						Type	State		Continue on Failure
config_bond						Install File(s)	Enabled		<input type="checkbox"/>
run_config-bond						Run Script	Enabled		<input type="checkbox"/>

If the new bundle is assigned to a device in its current state it does not work because the last three modifications only exist in the sandbox of the bundle. To make these modifications available to devices, a new version of the bundle needs to be published.

Figure 11-51 Publish a New Bundle Version

[Bundles](#) > [Linux](#) > [CUSTOM](#) > [SLES11-NETWORK](#) > [SLES11ALL_CUST_BONDING@Sandbox](#) > **Publish**

Publish

Step 1: Publish Option

☒ Publish as new version

☐ Publish as new bundle

Name:

Folder:

/Bundles/Linux/CUSTOM/SLES11-NET

☐ Create as Sandbox

Select the groups that the new bundle should be a member of.

Select Groups

<input type="checkbox"/>	Name	In Folder
No items available.		

Fields marked with an asterisk are required.

<< Back

Next >>

Finish

Cancel

When the bundle is published, the version number is incremented by one and the sandbox flag is removed.

Figure 11-52 New Bundle Version Published

[Bundles](#) > [Linux](#) > [CUSTOM](#) > **SLES11-NETWORK**

Bundles							
New Edit Delete Action Quick Tasks							
<input type="checkbox"/>	Status	Name	Type	Category	Enabled	Version	Has Sandbox
<input type="checkbox"/>		SLES11ALL_CUST_BONDING	Linux Bundle	Install File(s)	Yes	1	No
1 - 1 of 1 show 25 items							

11.4.4 Bundle Groups and Bundles Developed by Novell Consulting

Standard settings of the operating system and for some Novell services do not necessarily meet the special requirements of distributed, clustered SAN-based environments that dominate IT infrastructures today. Although tools are necessary to manage the configuration aspects that are typical for this kind of infrastructure, Novell Consulting has encountered many environments where

no configuration management tools were available for SLES servers or OES servers. Therefore, Novell Consulting has instrumented ZENworks Configuration Management to deploy the various configuration items that are necessary to manage settings and services of OES servers.

In recent projects, similar configuration items have been identified for different customers. ZENworks Configuration Management bundles have been build by Novell Consulting to perform configuration tasks common to standard OES environments. These bundles and associated bundle groups are described in this section, along with identifying their appropriate parent folders.

- ♦ [“The BUNDLE-GROUPS Folder” on page 126](#)
- ♦ [“Bundles in the _ENGINEERING-BUILDS Folder” on page 127](#)
- ♦ [“Bundles in the EDIR-BASECONFIG Folder” on page 127](#)
- ♦ [“Bundles in the IMAN-BASECONFIG Folder” on page 132](#)
- ♦ [“Bundles in the SLES11-BASECONFIG Folder” on page 133](#)
- ♦ [“Bundles in the OES11-BASECONFIG Folder” on page 134](#)
- ♦ [“Bundles in the OES11-SERVICES Folder” on page 136](#)
- ♦ [“Bundles in the SLES11_NETWORK Folder” on page 138](#)
- ♦ [“Bundles in the SLES11-STORAGE Folder” on page 140](#)

The BUNDLE-GROUPS Folder

The purpose of this folder is to collect all customer-specific bundle groups in one place. These bundle groups in turn are assigned to device groups (see [Section 11.3.2, “Server Group Objects in the Device Menu,” on page 96](#)). Typically there is a 1:1 relationship between a bundle group and a device group whereas the same bundle can be a member of multiple bundle groups. This approach has been chosen to reduce the number of mouse clicks required to assign a bundle to its target devices

Novell Consulting strongly recommends that you distinguish between bundle groups for production environments and bundle groups for test/development environments. Therefore, every environment needs its own bundle groups.

Most customers who are utilizing OES services are also using Novell Cluster Services. Novell Consulting recommends that you create separate bundle groups for every cluster.

For example, a cluster named `du scl01` uses the `OES11ALL-PROD-du scl01-GRP` bundle group, which contains the following bundles:

- ♦ `OES11ALL_NOV-LVM-CONF`
- ♦ `SLES11ALL_CUST-BINDINGS-du scl01`

These bundles configure the Linux Volume Manager and name the multi-path devices.

NOTE: In the following sections, only examples for production environments are provided. The examples can be easily adapted for other environments.

Bundles in the _ENGINEERING-BUILDS Folder

Field test files provided by Novell Engineering for a certain customer are not part of official update channels and should be placed in this folder. These bundles are assigned to individual device objects or to device groups as needed.

According to the established naming standards in [“Naming Standards for Bundles” on page 116](#), a bundle located here must contain the string `ENG` in its name to identify its origin. For example, `OES11SP1only_ENG-CIFSD-20120215`, where the date string in the example above is part of the `%PURPOSE%` tag. Its intention is to inform the administrator about the exact date when the particular FTF was delivered and to allow a distinction between different engineering builds of the same module.

Bundles in the EDIR-BASECONFIG Folder

- ♦ [“EDIR_NOV-ARC” on page 127](#)
- ♦ [“EDIR_NOV-DSRMENU” on page 132](#)

EDIR_NOV-ARC

Advanced Referral Costing (ARC) provides an improved server-to-server communication. ARC helps eDirectory servers avoid servers that are responding at a transport level, but are slow or unresponsive to eDirectory requests. This feature must be enabled by using DSTrace or NDSTrace. It is not enabled by default.

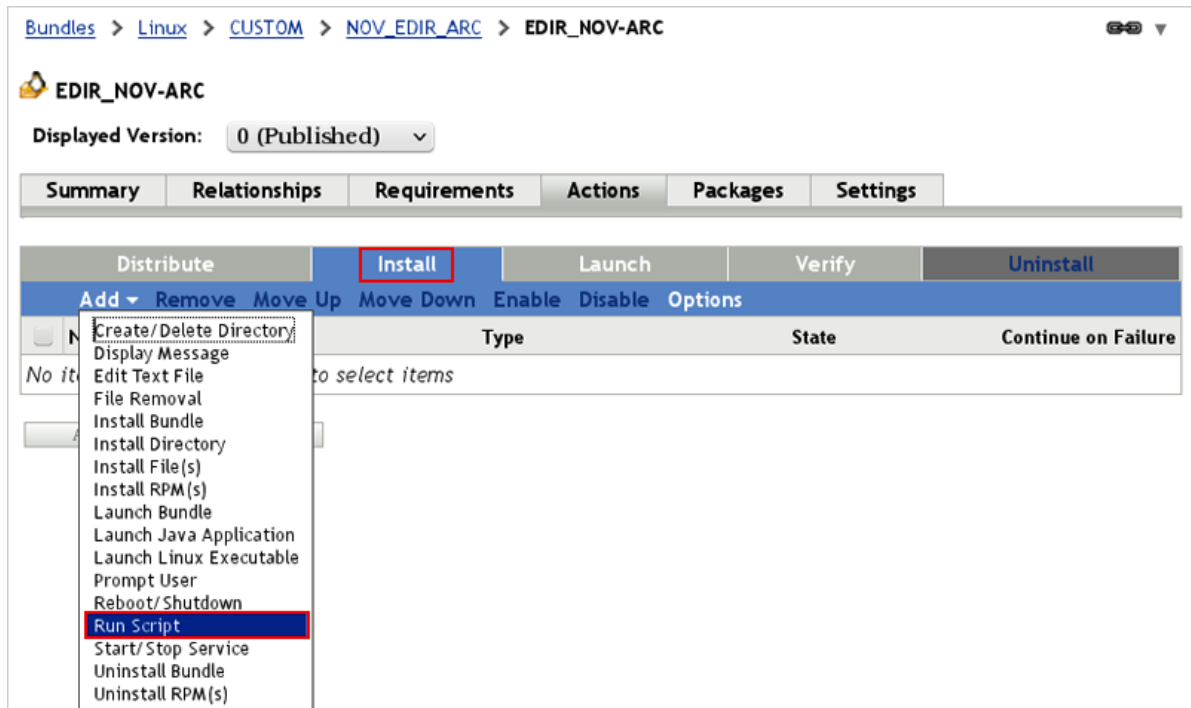
An empty `EDIR_NOV-ARC` file bundle without content but with the following post-installation script needs to be deployed to configure ARC at the target devices:

```
#!/bin/sh
my_cmd=/opt/novell/eDirectory/bin/ndstrace
$my_cmd -l >>/dev/null &
$my_cmd -c 'set ndstrace = !ARC1'
$my_cmd -u
```

The creation process for the `EDIR_NOV-ARC` file bundle is shown here because it is a task that must be configured many times. The first common steps when creating a new bundle are not displayed because they are the same for any bundle type and they have been shown already in [Figure 11-41 on page 119](#) through [Figure 11-50 on page 124](#).

After the bundle has been created, you need to define the post-installation script in the *Additional Properties* menu by clicking *Actions > Install* and activating the *Add* link. You must select *Run Script* from the drop-down list.

Figure 11-53 EDIR_NOV-ARC - Add Install Action



The following menu asks whether a script located on the managed device should be executed, a script from the workstation where the browser has been started should be uploaded, or an own script should be defined. To configure ARC, you need to define an own script (see [Figure 11-54 on page 129](#) and [Figure 11-59 on page 131](#)).

You can select *Edit* and type the script, or you can or paste the script (Figure 11-55 on page 129). This screen also allows you to change the *Action Name* to a descriptive name such as `configure_ARC` in the example:

Figure 11-54 *EDIR_NOV-ARC - Define Script*

Add Action - Run Script

Action Name: *

General **Advanced** **Requirements**

Script to Run:

Script File Name: *
(e.g. /root/scripts/xyz.pl)

Script Parameters:

Path to Script Engine:

Script Engine Parameters:

Wait before proceeding to next action

☒ No wait

☐ When action is complete

☐ Wait for
seconds

☐ Terminate action if wait period is exceeded

Fields marked with an asterisk are required.

Figure 11-55 *EDIR_NOV-ARC - Define Script Content*

Edit - Script Content

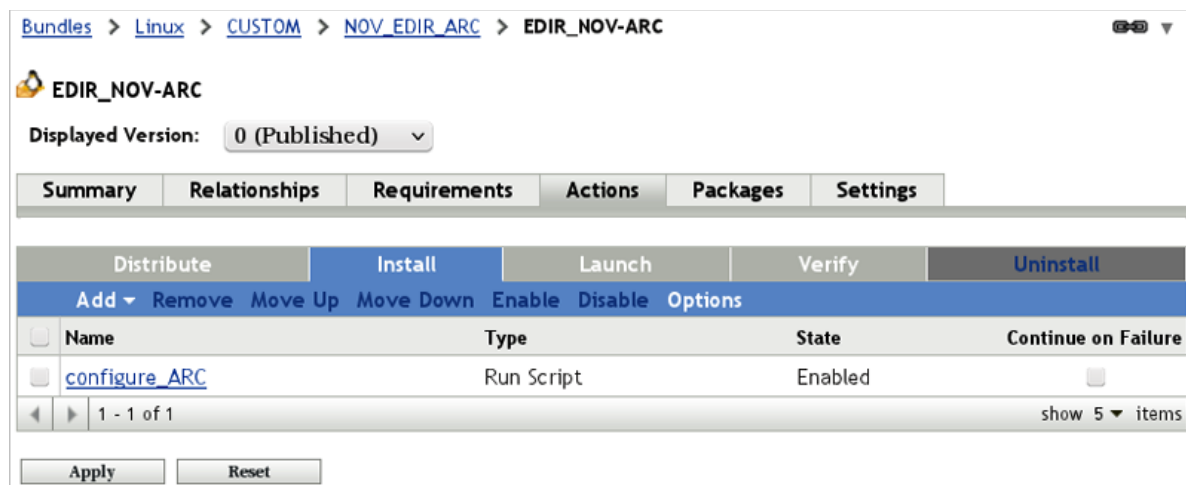
Script content: *

```
#!/bin/sh

my_cmd=/opt/novell/eDirectory/bin/ndstrace
$my_cmd -l >>/dev/null &
$my_cmd -c 'set ndstrace =!ARC'
$my_cmd -u
```

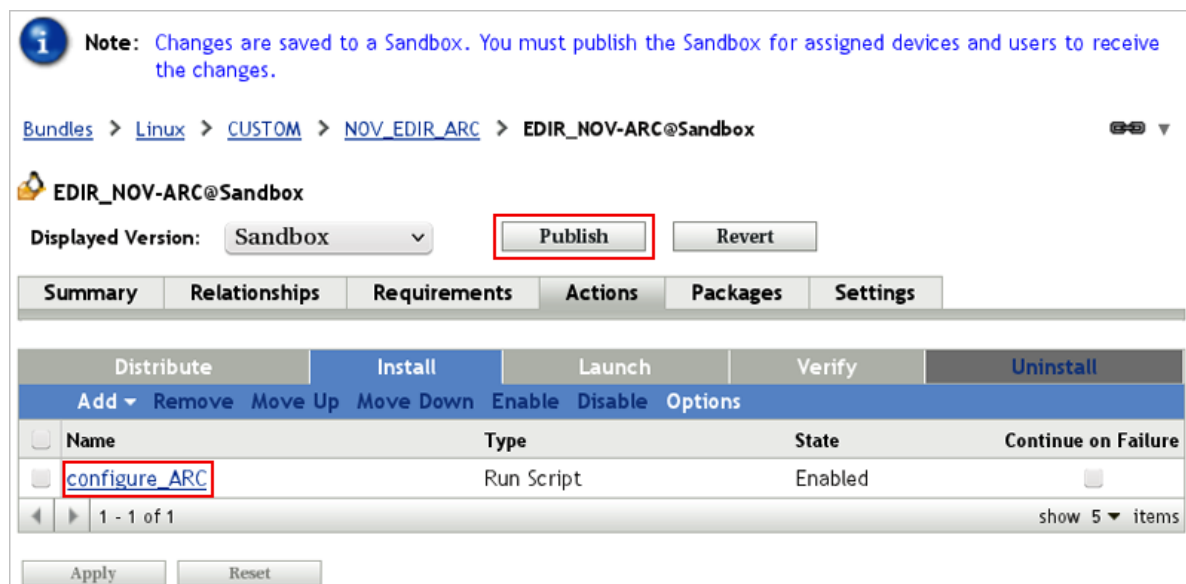
Click *OK* to display the following figure. The bundle is ready for use; it becomes available after you click *Apply*.

Figure 11-56 *EDIR_NOV-ARC - Action Created*



Changes always create a sandbox version for a bundle. This version must be published to become available for all device groups:

Figure 11-57 *EDIR_NOV-ARC - Sandbox*



Finally, you need to click *Finish*.

Figure 11-58 *EDIR_NOV-ARC - Publish as New Version*

Bundles > Linux > CUSTOM > NOV_EDIR_ARC > EDIR_NOV-ARC@Sandbox > Publish

Publish

Step 1: Publish Option

☒ Publish as new version

☐ Publish as new bundle

Name:

Folder:

☐ Create as Sandbox

Select the groups that the new bundle should be a member of.

Select Groups

<input type="checkbox"/>	Name	In Folder
No items available.		

Fields marked with an asterisk are required.

<< Back Next >> **Finish** Cancel

The bundle is now published and ready for assignment:

Figure 11-59 *EDIR_NOV-ARC - Bundle Published*

Bundles > Linux > CUSTOM > NOV_EDIR_ARC > EDIR_NOV-ARC

EDIR_NOV-ARC

Displayed Version: 1 (Published) ▼

Summary Relationships Requirements Actions Packages Settings

Distribute Install Launch Verify Uninstall

Add ▼ Remove Move Up Move Down Enable Disable

<input type="checkbox"/>	Name	Type	State	Continue on Failure
<input type="checkbox"/>	Distribute Files	Distribute Files	Enabled	<input type="checkbox"/>

1 - 1 of 1 show 5 ▼ items

Apply Reset

The bundle must be a member of all bundle groups that are assigned to OES or SLES device groups representing systems where eDirectory is installed.

Summary

- ♦ **Components:** eDirectory, ARC

- ♦ **Tools:** ndstrace
- ♦ **Environments:** All eDirectory instances
- ♦ **Configuration Files/Paths:** N/A
- ♦ **Documentation:** “Advanced Referral Costing” (<https://www.netiq.com/documentation/edir88/edir88/?page=/documentation/edir88/edir88/data/b9u7705.html>) in the *Novell eDirectory 8.8 Administration Guide*.
- ♦ **Technical Information Documents:** N/A

EDIR_NOV-DSRMENU

NetWare administrators managing eDirectory have been very familiar with the `DSREPAIR.NLM` utility for many years. On OES Linux, the `ndsrepair` command line tool is used for the same eDirectory maintenance tasks, but its appearance and usage is quite different from `DSREPAIR.NLM`.

This file bundle contains a wrapper script from Novell called `dsmenu.sh`, which provides a similar layout for `ndsrepair` as for `DSREPAIR.NLM` on NetWare.

The `dsmenu.sh` file is distributed to the local `/bin` directory of the target device via the `EDIR_NOV-DSRMENU` file bundle.

The bundle must be a member of all bundle groups that are assigned to OES or SLES device groups representing systems where eDirectory is installed.

Summary

- ♦ **Components:** eDirectory, `dsmenu.sh`
- ♦ **Tools:** `ndsrepair`, `dsmenu.sh`
- ♦ **Environments:** All eDirectory instances
- ♦ **Configuration Files/Paths:** `/bin/dsmenu.sh`
- ♦ **Documentation:** `DSRMENU` (<http://www.novell.com/communities/node/2282/ndsrepair-unix-menu-wrapper>)
- ♦ **Technical Information Documents:** N/A

Bundles in the IMAN-BASECONFIG Folder

This folder contains bundles that change the configuration of Novell iManager.

- ♦ “`IMAN_NOV-J2EE`” on page 132

IMAN_NOV-J2EE

OES servers hosting services like iManager or NetStorage are currently using Tomcat 6 as an application server. The Java memory settings of Tomcat 6 are set to conservative values by default. Because sufficient memory is typically available on modern server hardware, a possible performance bottleneck can be avoided by increasing the Java memory settings for Tomcat 6.

An empty `IMAN_NOV-J2EE` file bundle must be created that contains no files, but modifies the `/etc/sysconfig/j2ee` configuration file to adopt the memory settings via a script.

This bundle must be a member of all bundle groups assigned to devices running iManager.

Summary

- ♦ **Components:** iManager

- ♦ **Tools:** iManager instances
- ♦ **Environments:** All eDirectory instances
- ♦ **Configuration Files/Paths:** /etc/sysconfig/j2ee
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** TID 7009773 (<http://www.novell.com/support/kb/doc.php?id=7009773>)

Bundles in the SLES11-BASECONFIG Folder

This folder contains bundles created for devices running SLES 11 and its service packs. It also includes devices that have OES 11 configured as an add-on product.

- ♦ “SLES11ALL_NOV-FORCEDUMP-CONFIG” on page 133
- ♦ “SLES11ALL_NOV-APPCORE-CONFIG” on page 133

SLES11ALL_NOV-FORCEDUMP-CONFIG

By default, a SLES 11 system is not configured to capture a memory dump (vmcore) after a system crash.

With kdump, several components can be configured, such as where to store a kernel memory dump, the name of the dump file, and how many dump files should be preserved.

The `kexec-tools` and `kdump` packages must be installed on target devices. You need sufficient space below `/var` to save up to five memory dumps.

A file bundle containing `kdump` configuration parameters has been created. The file is delivered to `/etc/sysconfig/kdump` on target devices. In addition, the `/etc/sysconfig/bootloader` file is modified by a post-installation script at the target devices to preset the `DEFAULT_APPEND (/etc/sysconfig/bootloader)` variable for configuring the crashkernel as described in TID 3374462 (<http://www.novell.com/support/kb/doc.php?id=3374462>). The post-installation script ensures that `boot.kdump` is enabled (`/sbin/chkconfig`) and the crash kernel parameters are inserted into the boot configuration file for grub (`/boot/grub/menu.lst`).

The bundle is a member of all bundle groups assigned to SLES or OES devices.

Summary

- ♦ **Components:** `kdump`, `grub`, `kexec`
- ♦ **Tools:** `sysctl`
- ♦ **Environments:** Devices where kernel core files need to be taken from for debugging reasons
- ♦ **Configuration Files/Paths:** `/etc/sysconfig/kdump`, `/etc/sysconfig/ulimit`
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** TID 3374462 (<http://www.novell.com/support/kb/doc.php?id=3374462>)

SLES11ALL_NOV-APPCORE-CONFIG

This bundle configures systems for application dumps.

The `ulimit` package must be installed at the target device and sufficient space below `/var` is required.

An empty bundle that carries a post-installation script has been created. The post-installation script performs the following steps:

- ♦ Creates the `/var/cores` directory that contains the dump files
- ♦ Persistent configuration of the dump directory by inserting

```
kernel.core_pattern=/var/cores.%e.%p
```


into

```
/etc/sysctl.conf
```
- ♦ Enables core dumps for `setuid` and `setgid` processes in `/etc/sysctl.conf` -

```
kernel.suid_dumpable=2
```
- ♦ Issuing the `sysctl -p /etc/sysctl.conf` command activates these new settings without rebooting.
- ♦ Disables the limit for the maximum size of a core dump file by setting

```
HARDCORELIMIT="unlimited" and SOFTCORELIMIT="0"
```

 in `/etc/sysconfig/ulimit`

The bundle is a member of all bundle groups assigned to SLES or OES devices.

Summary

- ♦ **Components:** `sysctl`, `ulimit`
- ♦ **Tools:** `sysctl`
- ♦ **Environments:** All devices
- ♦ **Configuration Files/Paths:** `/etc/sysconfig/ulimit`, `/etc/sysctl.conf`
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** [TID 3054866 \(http://www.novell.com/support/kb/doc.php?id=3374462\)](http://www.novell.com/support/kb/doc.php?id=3374462)

Bundles in the OES11-BASECONFIG Folder

This folder contains bundles that are applicable to OES 11 devices.

- ♦ [“OES11ALL_NOV-AUDIT-OFF” on page 134](#)
- ♦ [“OESALL_NOV-BASHRC” on page 135](#)
- ♦ [“OES11ALL_NOV-NCP-SETTINGS” on page 135](#)

OES11ALL_NOV-AUDIT-OFF

This bundle stops and de-activates the audit daemon that is automatically activated when `novell-afp` is installed. It must be assigned to all systems running `novell-afp`.

A post-installation script that is configured in an empty file bundle stops the audit daemon and ensures that it is not started after reboot. It is implemented by calling `chkconfig` and stopping the daemon via its init script.

The bundle needs to be a member of all bundle groups assigned to devices where `novell-afp` is installed.

Summary

- ♦ **Components:** `novell-afp`, `auditd`
- ♦ **Tools:** `chkconfig`

- ♦ **Environments:** NSS file server with AFP enabled
- ♦ **Configuration Files/Paths:** `/etc/init.d/auditd`
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** TID 7006838 (<http://www.novell.com/support/kb/doc.php?id=7006838>) (in particular, take note of Problem 4).

OESALL_NOV-BASHRC

For easier administration of cluster nodes, some alias commands were defined by Novell Consulting. This file bundle distributes the appropriate configuration file. Currently, the file contains mostly cluster-related aliases, but any other useful aliases can be implemented through this file.

The OESALL_CUST_BASHRC file bundle distributes the `bash.bashrc.local` file to `/etc` at the target device. Within this file, aliases for console commands are defined. It is automatically read when starting a BASH shell (for example, at login). This file bundle is strongly recommended to be a member of all bundle groups assigned to NCS cluster nodes. It can be made a member of all bundle groups assigned to OES devices.

Summary

- ♦ **Components:** Novell Cluster Services, `bashrc`
- ♦ **Tools:** `shell`
- ♦ **Environments:** Novell Cluster Services
- ♦ **Configuration Files/Paths:** `/etc/bash.bashrc.local`
- ♦ **Documentation:** [Novell Cool Solutions \(http://www.novell.com/cool solutions/tools/17142.html\)](http://www.novell.com/cool solutions/tools/17142.html)
- ♦ **Technical Information Documents:** N/A

OES11ALL_NOV-NCP-SETTINGS

Default NCP server settings are configured for a broad range of environments. However, they need to be optimized for customer environments by using this bundle.

An empty OES11ALL_NOV-NCP-SETTINGS file bundle must be created that configures NCP settings such as cache and watchdog parameters via a post-installation script by executing `ncpcon`.

For example:

- ♦ `MAXIMUM_CACHED_FILES_PER_VOLUME=NNNNNN`
- ♦ `MAXIMUM_CACHED_SUBDIRECTORIES_PER_VOLUME=NNNNNN`

Typically, you only need to modify these settings on OES 11 systems that provide very large file systems to a large number of users over NCP. Make this bundle a member of all bundle groups assigned to such devices.

Summary

- ♦ **Components:** NCP
- ♦ **Tools:** `ncpcon`
- ♦ **Environments:** Novell Cluster Services
- ♦ **Configuration Files/Paths:** `/etc/opt/novell/ncpserv.conf`
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** TID 7004848 (<http://www.novell.com/support/kb/doc.php?id=7004848>), TID 7004888 (<http://www.novell.com/support/kb/doc.php?id=7004888>).

Bundles in the OES11-SERVICES Folder

This folder contains bundles that provide configurations for OES 11 services such as Novell-FTP, LUM, and NSS.

- ♦ [“OES11ALL_NOV-FTP-SYSLOG-xxx” on page 136](#)
- ♦ [“OESALL_NOV-PURE-FTPD-CONF” on page 136](#)
- ♦ [“OES11ALL-CUST-NAM-xxx” on page 137](#)
- ♦ [“OESALL_NOV-NSS-SET-XATTR” on page 137](#)

OES11ALL_NOV-FTP-SYSLOG-xxx

The Novell-FTP service utilizes pure-ftpd to provide its functionality. The standard pure-ftpd writes its log files to the local file system. However, in clustered environments, it is desirable to have the log files on the cluster resource. A configuration change for `syslog-ng` is necessary in those environments to change the default logging behavior from local devices to the cluster resource.

Because the configuration of `syslog-ng` is related to the resource name, it is different on every cluster, so every cluster that needs this redirection of FTP logs needs its own bundle.

A configuration setting in `syslog-ng.conf` must be distributed via a file bundle to `/etc/syslog-ng` and the `SuSEconfig` command must be executed within a post-installation script of the file bundle.

NOTE: The `SyslogFacility` directive in the `ftp.conf` file for your cluster-enabled FTP instance must be set to the syslog facility configured in `syslog-ng.conf`. For example, `local3`.

In addition to the prefix defined in the bundle naming standards, an identifier of the appropriate cluster should be chosen as the suffix. For example, `OES11ALL_NOV-FTP-SYSLOG-duscl01` is a bundle name consistent with the naming standard for a bundle that configures FTP logging for the `duscl01` cluster.

The bundle should be a member of a bundle group with the purpose of cluster assignment, such as `OES11ALL-PROD-duscl01-GRP`.

Summary

- ♦ **Components:** `novell-ftp`, `pure-ftpd`, `syslog-ng`
- ♦ **Tools:** `SuSEconfig`
- ♦ **Environments:** Novell Cluster Services with `novell-ftp` enabled
- ♦ **Configuration Files/Paths:** `/etc/syslog-ng`
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** N/A

OESALL_NOV-PURE-FTPD-CONF

This bundle disables the start of the local instance of `pure-ftpd` on cluster nodes using `novell-ftp`. To avoid interference of local FTP instances with the configuration of FTP cluster resources, the local `pure-ftpd` instance is also configured to listen on the host IP address only. FTP for cluster resources is managed from the NCS load/unload scripts.

A script carried by an empty file bundle ensures that the `bind` directive in the `/etc/pure-ftpd/pure-ftpd.conf` global configuration file is changed to the host IP address only. In addition, the script deactivates the start of `pure-ftpd` during the boot process by executing `checkconfig`.

This bundle is an example for a bundle that is in multiple bundle groups because the configuration changes fit any cluster environment. The bundle must be a member of all bundle groups for clusters where novell-ftp services are running.

Summary

- ♦ **Components:** novell-ftp, pure-ftpd
- ♦ **Tools:** chkconfig
- ♦ **Environments:** Novell Cluster Services with novell-ftp enabled
- ♦ **Configuration Files/Paths:** /etc/pure-ftpd/pure-ftpd.conf
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** TID 7005792 (<http://www.novell.com/support/kb/doc.php?id=7005792>)

OES11ALL-CUST-NAM-xxx

This bundle deploys settings for the namcd daemon to configure LUM for a specific device group.

Depending on the network infrastructure, it might be necessary to have different LDAP servers configured per system. Systems in different locations might need a separate bundle to properly configure LUM.

Important properties of namcd are changed by a post-installation script that is part of an empty file bundle and that restarts the daemon after the configuration change.

The usual naming rules are also applied here. The bundle also contains the customer name `CUST` because it is a customer-related change in the configuration file. In addition, identifiers for the location should be appended, such as `OES11ALL-CUST-NAM-<Location1>`.

The bundle is a member of a bundle group collecting all bundles that need to be assigned to all devices in a specific location, such as `OES11-PROD-DUS-GRP`.

Summary

- ♦ **Components:** novell-lum
- ♦ **Tools:** namconfig
- ♦ **Configuration Files/Paths:** /etc/nam.conf
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** N/A

OESALL_NOV-NSS-SET-XATTR

This bundle sets attributes in `nssstart.cfg` that determine how creation time, modification time, and extended attributes are handled. It is necessary in environments where the backup software has no interface to `novell-sms` and therefore does not recognize extended attributes of the NSS implementation on Linux.

A post-installation script of the empty file bundle configures the correct entries in `/etc/opt/novell/nss/nssstart.cfg`. A reboot is necessary to activate these settings.

The bundle needs to be a member of all bundle groups assigned to OES devices that have their NSS file systems backed up with non-SMS-aware backup solutions.

Summary

- ♦ **Components:** NSS, backup software

- ♦ **Tools:** N/A
- ♦ **Environments:** Novell File Services, Backup
- ♦ **Configuration Files/Paths:** `/etc/opt/novell/nss/nssstart.cfg`
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** N/A

Bundles in the SLES11_NETWORK Folder

Bundles located in this folder are used to configure network-related properties such as bond devices and routing behavior.

- ♦ [“SLES11ALL_CUST-BONDING” on page 138](#)
- ♦ [“SLES11ALL_NOV-VNIC” on page 139](#)
- ♦ [“SLES11ALL_CUST-QUAGGA” on page 140](#)

SLES11ALL_CUST-BONDING

This bundle joins multiple network interfaces to a bond device by using an XML configuration file and a part of the AutoYaST framework `ay_setup.ycp`. The reliability of the bond members is monitored by the ARP ping technique. The bond members can be passed as parameters to the bundle.

The bundle checks for the existence of the `/etc/sysconfig/network/ifcfg-bond0` configuration file, and executes only if this file does not exist. This protects against service interruption through unwanted reconfiguration of the network. If you need to reconfigure your bond device, ensure that you delete this configuration file first.

If the configuration file does not exist, the `config_bond.sh` script and the `ay_setup.ycp` tool are distributed to `/usr/local/zac` and executed afterwards.

Existing routes in addition to the default route are not considered when the network interfaces are reconfigured. Existing secondary IP addresses are lost.

The bundle name contains a customer part because it differs between particular environments regarding IP target address and bond members.

The bundle must be assigned to all cluster nodes and machines where a bond interface is configured. This means that membership in all bundle groups intended to configure cluster and high availability environments is required.

Summary

- ♦ **Components:** Pacemaker, Network interfaces, routes
- ♦ **Tools:** `ay_setup.ycp`
- ♦ **Environments:** Novell Cluster Services, SLES HAE
- ♦ **Configuration Files/Paths:** `/etc/sysconfig/network/*`
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** N/A

SLES11ALL_NOV-VNIC

In some situations it is beneficial if the IP address of a service is independent of the network attributes of the host devices (such as address ranges and net masks) so you can have maximum flexibility when those parameters change (for example, the IP address of the host device or the network mask).

This is required in a Business Continuity Cluster but can also be used in an ordinary cluster to assign IP addresses to services that are independent of the IP addresses of the cluster nodes (service mobility).

One of the means implemented by Novell Consulting to achieve this goal is to configure service IP addresses with 32-bit net masks (CIDR notation /32) bound to a virtual network interface (VNIC). If a service with this configuration migrates to a different location, only the route to the particular address changes. In addition, the service can be moved to any network segment if the route to the device hosting the service can be propagated.

Route propagation requires a dynamic routing protocol such as OSPF. Host servers and cluster nodes using this approach need to be configured as routers within a particular OSPF area.

The intention of this bundle is to implement the virtual network interface on the target machines. The virtual device name is `VNIC`.

The bundle contains several files to distribute to the target devices:

- ♦ `hwcfg-static-0` contains information about the dummy module and its options. It is distributed to `/etc/sysconfig/hardware`.
- ♦ `boot.local` loads the dummy driver and ensures that the virtual device is named `VNIC`. It is distributed to `/etc/init.d`.
- ♦ `ifup-VNIC` ensures that IP addresses bound to `VNIC` will persist a network restart. It is distributed to `/etc/sysconfig/network/scripts`.
- ♦ `ifcfg-VNIC` configures the required network attributes (startmode, MTU, ...) for the virtual device. It is distributed to `/etc/sysconfig/network`.

A post-installation script configures a link from `/etc/sysconfig/network/scripts/ifup-VNIC` to `/etc/sysconfig/network/ifdown-VNIC`.

A reboot is necessary to activate the virtual device.

The bundle must be assigned to all bundle groups configuring hosts of services that use virtual IP.

Summary

- ♦ **Components:** Quagga, zebra, ospfd, VNIC
- ♦ **Tools:** modprobe
- ♦ **Environments:** Novell Cluster Services, SLES HAE
- ♦ **Configuration Files/Paths:** `/etc/sysconfig/network/*`, `/etc/init.d/boot.local`
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** N/A

SLES11ALL_CUST-QUAGGA

The bundle configures OSPF routing via the zebra daemon. OSPF routing can be used for the automatic propagation of cluster resource IP addresses after a service using virtualIP has moved to a different node.

Template files for `/etc/quagga/ospfd.conf` and `/etc/quagga/zebra.conf` are distributed to the target devices. A post-installation script replaces server names and network addresses within the templates and starts and enables (`chkconfig`) the relevant `zebra` and `ospfd` daemons via their respective init scripts.

The bundle name contains a customer part because environments differ with respect to the OSPF ID, OSPF passwords, and other OSPF information.

The bundle must be a member of all bundle groups assigned to hosts that provide services using virtual IP.

Summary

- ♦ **Components:** Quagga, zebra, ospfd, VNIC
- ♦ **Tools:** `chkconfig`
- ♦ **Environments:** Novell Cluster Services, SLES HAE
- ♦ **Configuration Files/Paths:** `/etc/sysconfig/network/*`
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** N/A

Bundles in the SLES11-STORAGE Folder

Bundles located in this folder have the purpose to configure properties of HBA hardware, volume management, or multi-pathing.

- ♦ [“SLES11ALL_NOV-LVM-CONF” on page 140](#)
- ♦ [“SLES11ALL_CUST-MULTIPATH-CONF” on page 141](#)
- ♦ [“SLES11ALL_CUST-BINDINGS-xxx” on page 141](#)

SLES11ALL_NOV-LVM-CONF

LVM (Logical Volume Management), which is the standard logical volume management system on Linux, is used to manage local and shared devices and also configures `clvm`, which manages cluster-enabled POSIX file systems.

The `lvm.conf` file is distributed to target devices in the `/etc/lvm` path .

This bundle is required by all devices that use LVM to manage storage systems and must be a member of all bundle groups assigned to such devices.

Summary

- ♦ **Components:** `lvm`, `clvm`, Novell Cluster Services, Pacemaker (pure SLES)
- ♦ **Tools:** N/A
- ♦ **Environments:** Novell Cluster Services, Pacemaker
- ♦ **Configuration Files/Paths:** `/etc/lvm.conf`
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** N/A

SLES11ALL_CUST-MULTIPATH-CONF

A multipath configuration file has settings for different storage systems. They may come from the same vendor or different vendors. Settings like path access, location of the bindings file, policies, behavior on errors, and blacklisted devices must be configured for an optimal storage access. All of these settings are configured by `/etc/multipath.conf` on SLES 11. Novell Consulting typically uses a single `multipath.conf` file that holds the settings for all storage systems in an environment.

The `multipath.conf` file is distributed by this bundle and the multipath daemon is reconfigured by issuing the `multipathd -k'reconfigure'` command.

The bundle must be assigned to all OES 11 and SLES 11 machines with SAN attached storage, so it needs to go to all bundle groups that configure these device groups, such as `OES11ALL-PROD-duscl01-GRP`.

Summary

- ♦ **Components:** HAE (Pacemaker), Novell Cluster Services, multipath
- ♦ **Tools:** N/A
- ♦ **Environments:** All systems with SAN attached storage devices
- ♦ **Configuration Files/Paths:** `/etc/multipath.conf`
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** N/A

SLES11ALL_CUST-BINDINGS-xxx

A bindings file is specific for the nodes of a certain cluster or for a stand-alone server with SAN attached storage devices. It contains a table that assigns alias names to world-wide IDs presented by the MPIO module that retrieves them from the target storage system.

`Bindings` is a file that is distributed through a file bundle to the `/etc/multipath` directory.

By default the multi-path daemon places this file in `/var/lib/multipath`. However, if `/var` is in a separate partition and the partition has not finished mounting when the multipath daemon loads, the file will not be available for the daemon.

To avoid this kind of problem the `bindings` file must be stored in the root partition of the file system. Novell Consulting has chosen the `/etc/multipath` directory for this purpose.

The `xxx` in the heading must be replaced by an identifier that describes the environment of the target devices (that is, a cluster identifier). A name that is with the naming standard is `OES11ALL_<CUST>-BINDINGS-duscl01`.

This bundle must be a member of the bundle group assigned to the device group representing all nodes of the corresponding cluster, such as `OES11ALL-PROD-duscl01-GRP` in this example.

Summary

- ♦ **Components:** Pacemaker, Novell Cluster Services, multipath
- ♦ **Tools:** N/A
- ♦ **Environments:** Novell Cluster Services, SLES HAE
- ♦ **Configuration Files/Paths:** `/etc/multipath/bindings`
- ♦ **Documentation:** N/A
- ♦ **Technical Information Documents:** N/A

11.5 Subscriptions

Subscriptions in ZENworks Configuration Management configure how packages and patches are mirrored from external repositories and how these packages and patches are presented within ZENworks Configuration Management. A subscription can define the following properties:

- ♦ The credentials to access the repositories
- ♦ The repositories or package selections that are mirrored
- ♦ The name of the resulting bundles
- ♦ The type of the resulting bundles
- ♦ The ZCM servers that provide the bundles to the intranet
- ♦ Highly configurable scheduling
- ♦ The location in the bundle menu where resulting bundles are placed

These properties are explained in detail in the next sections:

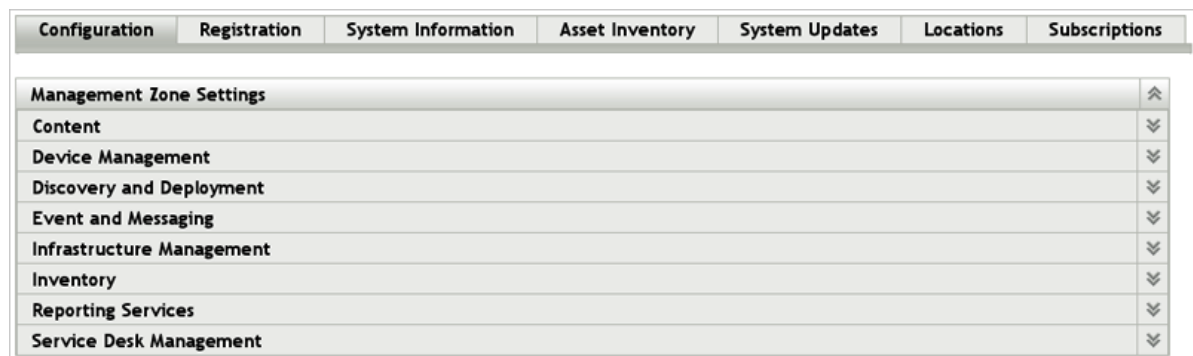
- ♦ [Section 11.5.1, “Credential Vault,” on page 142](#)
- ♦ [Section 11.5.2, “Proxy Setup,” on page 144](#)
- ♦ [Section 11.5.3, “Creating a Subscription Folder Structure,” on page 144](#)
- ♦ [Section 11.5.4, “Creating a Subscription,” on page 146](#)
- ♦ [Section 11.5.5, “Pool Channels, Core Channels, and Online Channels,” on page 153](#)

11.5.1 Credential Vault

Novell update repositories or repositories of other vendors like Red Hat are not freely accessible. A credential normally tied to a maintenance contract is needed to gain access to such repositories. In ZENworks Configuration Management, credentials are defined as objects within the credential vault.

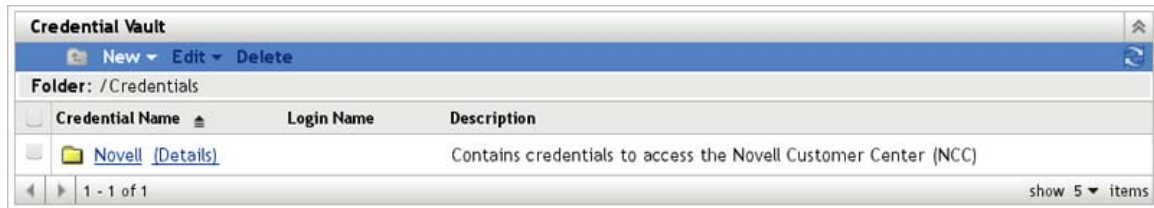
The credential vault can be accessed from ZENworks Control Center by selecting the *Configuration* tab and navigating to the bottom of the configuration frame:

Figure 11-60 Configuring the Credential Vault (1)



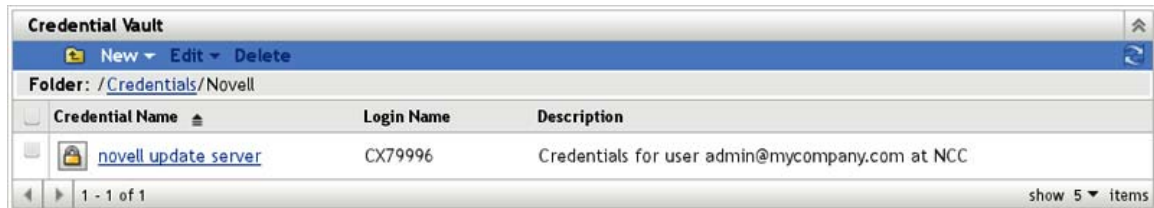
The vault can be organized into folders as already shown for bundles or devices. Novell Consulting recommends that you create a folder structure where the parent folders correspond to the appropriate vendors whose repositories are managed by ZENworks Configuration Management.

Figure 11-61 Configuring the Credential Vault (2)



The following figure shows a credential object below the `Novell` folder:

Figure 11-62 Configuring the Credential Vault (3)



The credential has the following properties:

- ♦ **Credential name:** A descriptive name that can be chosen by the administrator.
- ♦ **Login name:** Novell Customer Center (NCC) mirror credential login name.
- ♦ **Password:** NCC mirror credential password.
- ♦ **Description:** A meaningful description to tell others what this credential is for.

Figure 11-63 Configuring the Credential Vault (4)

The screenshot shows the 'Edit Credential' dialog box. It contains several input fields: 'Credential Name' (with 'novell update server' entered), 'Login Name' (with 'CX79996' entered), 'Password' (masked with dots), 'Reenter Password' (masked with dots), and 'Description' (with 'Credentials for user admin@mycompany.com at NCC' entered). Fields marked with an asterisk are required. At the bottom, there are 'OK' and 'Cancel' buttons.

Using a credential object is explained in [Section 11.5.4, “Creating a Subscription,”](#) on page 146.

11.5.2 Proxy Setup

For downloading patches from `nu.novell.com` or from external resources of other vendors, a connection outside the company networks is necessary. Usually this is achieved by using an HTTP proxy server.

Although ZENworks Configuration Management provides many menus to configure proxy settings, it does not have a menu to configure a proxy for downloading patches from outside the ZENworks Configuration Management zone. Currently, this proxy can be configured only via the command line in the `/etc/opt/novell/zenworks/lpm-server.properties` file.

An example `lpm-server.properties` file is displayed below:

```
Debug=false
TTL=24
subscription-proxyaddress=proxy.abc.com
subscription-proxyport=8080
subscription-proxyuser=ABCDE
subscription-proxypassword=Linux
subscription-useNTLM=false

useNCCViaProxy=true

# If you want to use same proxy for both NCC and Subscriptions use the properties
# mentioned in the lpm-server.properties file.
# Else If you want to use different proxy settings for NCC use the following
# properties by adding them into the properties file.

#ncc-proxyaddress=
#ncc-proxyport=
#ncc-proxyuser=
#ncc-proxypassword=
```

11.5.3 Creating a Subscription Folder Structure



Before you create a subscription, you need to define a suitable folder structure for the subscriptions to be managed.

The following rules apply for the naming scheme for subscription folders defined by Novell Consulting:

- ♦ The first folder level contains the vendor names of the package provider, such as Novell
- ♦ Folder names for products and releases are in uppercase
- ♦ A second folder level consists of the remote channel types of the appropriate vendors, such as ONLINE, POOLS, and UPDATE for Novell channels

The subscription tasks can be accessed by navigating to the configuration frame and choosing the *Subscription* tab.

Figure 11-64 Subscription Tab in ZENworks Configuration Management

Configuration	Registration	System Information	Asset Inventory	System Updates	Locations	Subscriptions
Subscriptions						
New Edit Delete Action Quick Tasks						
Folder: /Subscriptions						
<input type="checkbox"/>	Name	Type	Enabled	Status	Last Replication	
<input type="checkbox"/>	 Novell (Details)	Folder				
<input type="checkbox"/>	 Suse (Details)	Folder				
1 - 2 of 2						

The channel folder level and its content are illustrated in the following figures.

Figure 11-65 Subscription Folder Structure Layer 2















Configuration	Registration	System Information	Asset Inventory	System Updates	Locations	Subscriptions
Subscriptions						
New Edit Delete Action Quick Tasks						
Folder: /Subscriptions/Novell						
<input type="checkbox"/>	Name	Type	Enabled	Status	Last Replication	
<input type="checkbox"/>	 OES11 (Details)	Folder				
<input type="checkbox"/>	 OES2 (Details)	Folder				
1 - 2 of 2						

Figure 11-66 Subscription Folder Structure Layer 3

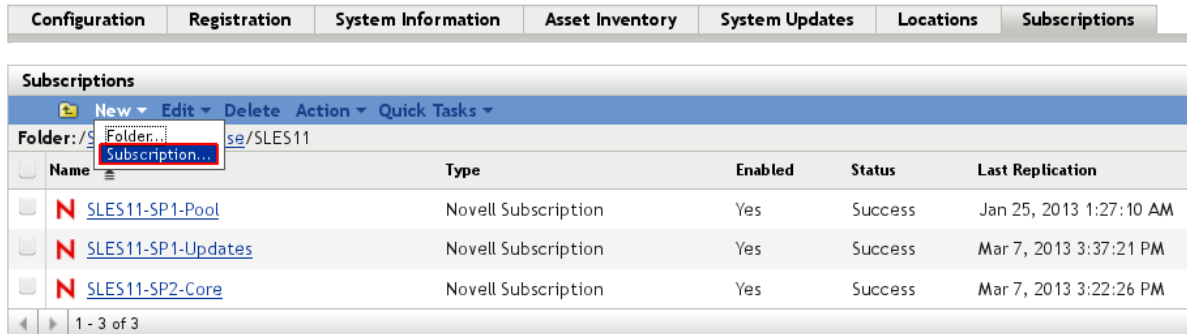
Configuration	Registration	System Information	Asset Inventory	System Updates	Locations	Subscriptions
Subscriptions						
 New ▾  Edit ▾  Delete  Action ▾  Quick Tasks ▾						
Folder: /Subscriptions/Novell/OES11						
<input type="checkbox"/>	Name 	Type	Enabled	Status	Last Replication	
<input type="checkbox"/>	 OES11-Pool	Novell Subscription	Yes	New		
<input type="checkbox"/>	 OES11-SP1-Pool	Novell Subscription	Yes	Success	Feb 19, 2013 12:26:37 PM	
<input type="checkbox"/>	 OES11-SP1-Updates	Novell Subscription	Yes	Success	Feb 19, 2013 11:57:58 AM	
<input type="checkbox"/>	 OES11-Updates	Novell Subscription	Yes	New		
  1 - 4 of 4						

11.5.4 Creating a Subscription

This section uses the `UPDATES` folder to demonstrate how to create a new subscription.

You start the creation of a new subscription by clicking *New* in the *Subscriptions* menu, then clicking *Subscription* in the drop-down list.

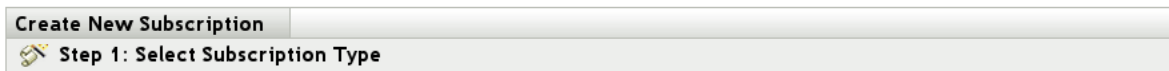
Figure 11-67 Creating a Subscription



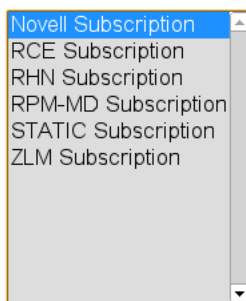
Click *Novell Subscription* in the list of available subscriptions.

Figure 11-68 Creating a Subscription - Select Subscription Type

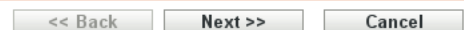
[Subscriptions](#) > **Create New Subscription**



Select the type of Subscription Repository that you would like to download from.



Novell Subscription - Allows you to download the catalogs available in the NU repository from Novell Customer Center and replicate them to the local server in your **Management Zone**.




Click *Next* to continue.

Figure 11-69 Creating a Subscription - Define Subscription Details

[Subscriptions](#) > Create New Subscription


Create New Subscription

 Step 2: Define Details


Enter the details for the Subscription

Subscription Name: *

Folder: *

Download To Folder: *

Description:

Mirror of the SLES11-SP2-Update channel
every night at 23:30

* Fields marked with an asterisk are required.

<< Back

Next >>

Cancel

Fill in the fields as follows:

Subscription Name: Novell Consulting recommends that you choose a descriptive name, preferably the name of the remote channel.

Folder: Leave this as is, because it points to the current subscription folder.

Download To Folder: This field specifies the folder where the resulting bundle will be placed. You can leave it as is, and change it in a later step.

Description: Provide a description to inform other administrators about the initial purpose of the object.


Click *Next* to display the menu shown in [Figure 11-70 on page 148](#). You now need the credential object you created in [Section 11.5.1, "Credential Vault," on page 142](#).

Click the *Search* icon (magnifying glass) to select the credentials.

Figure 11-70 *Creating a Subscription - Select Remote Server Credentials*


[Subscriptions](#) > **Create New Subscription**

Create New Subscription **SLES11-SP2-Updates**

 **Step 3: Define Remote Server Details**

Enter the details of the Remote Server to download from.

Remote Server URL: *

Remote Server Credentials: * 

Fields marked with an asterisk are required.


Click *OK* again to display a summary, then click *Next* to proceed with the next step of the subscription creation process.

If the credentials are valid, all Novell channels accessible by the account are shown and the appropriate catalog can be selected.

Figure 11-71 *Creating a Subscription - Select Catalog.*

[Subscriptions](#) > **Create New Subscription**

Create New Subscription **SLES11-SP2-Updates**

 **Step 4: Select Catalogs and Categories for download**

Based on the credentials you entered, you are entitled to download the following. Select the catalogs you want to replicate to the local server.

<input type="checkbox"/>	SLES11-SP1-VMware-Updates	All
<input type="checkbox"/>	SLES11-SP2-Core	All
<input type="checkbox"/>	SLES11-SP2-Extension-Store	All
<input type="checkbox"/>	SLES11-SP2-SUSE-Manager-Tools	All
<input checked="" type="checkbox"/>	SLES11-SP2-Updates	All
<input type="checkbox"/>	SLES11-SP2-VMware-Core	All
<input type="checkbox"/>	SLES11-SP2-VMware-Updates	All
<input type="checkbox"/>	SLES11-SP3-Extension-Store	All
<input type="checkbox"/>	SLES11-SP3-Pool	All
<input type="checkbox"/>	SLES11-SP3-Updates	All
<input type="checkbox"/>	SLES11-SP3-VMware-Pool	All

Fields marked with an asterisk are required.

You must also determine the target architectures for the bundle to be mirrored, as shown in the next figure. If you do not perform this step, updates for both targets (`sles-11-i586` and `sles-11-x86_64`) are downloaded by default and included in the bundle, which doubles the disk space required on the ZCM server.

Figure 11-72 Creating a Subscription - Select Target

Subscriptions > Create New Subscription

Create New Subscription SLES11-SP2-Updates

Step 4: Select Catalogs and Categories for download

Based on the credentials provided, the system will attempt to replicate to the local catalog.

Edit Catalog

SLES11-SP2-Updates

Targets

<input type="checkbox"/>	Target Name	All	Optional	Recommended	Security
<input type="checkbox"/>	sles-11-i586	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	sles-11-x86_64	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Local Catalog Name: SLES11-SP2-Updates

OK Cancel

Fields marked with an asterisk are required.

<< Back Next >> Cancel

Click *OK*, then click *Next* to go to the page where you can configure the schedule for the download as well as the subscription server. If the ZENworks Configuration Management environment includes several servers, you need to select a particular subscription server.

Many options are available to configure how you schedule the mirror. The following figure shows a recurring daily mirror starting at 11 p.m. For further information, see “Schedule Types” (http://www.novell.com/documentation/zenworks11/zen11_sys_servers/data/schedule_types.html) in the *ZENworks 11 SP2 Primary Server and Satellite Reference*.

Figure 11-73 Creating a Subscription - Schedule Download

[Subscriptions](#) > **Create New Subscription**

Create New Subscription SLES11-SP2-Updates
Step 5: Schedule Download

Schedule the timeslot for download of data.

Schedule Type:
Recurring ▾

Days of the week

★

Sun	Mon	Tue	Wed	Thu	Fri	Sat
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Start Time: 23 : 30 ▾

[More Options](#)

☐ **Monthly**

☐ Day of the month: 1
☐ Last day of the month
☐ First ▾ Sunday ▾

Start Time: 1 : 00 ▾

[More Options](#)

☐ **Fixed Interval**

0 Months 0 Weeks 0 Days 0 Hours 0 Minutes

Start Date: 3/21/201: Start Time: 1 : 00 ▾


[More Options](#)

Subscription Server:

The final step displays the summary of the configuration of the current subscription.

Figure 11-74 Creating a Subscription - Summary

[Subscriptions](#) > **Create New Subscription**

Create New Subscription	SLES11-SP2-Updates
 Step 6: Summary	

Review the information and click Finish to create the new Subscription.

Subscription Type: NU
Folder: /Subscriptions/Suse/SLES11
Subscription Name: SLES11-SP2-Updates
Subscription Server: /Devices/Servers/zcm11-node1
Description:

Mirror of the SLES11-SP2-Update channel every night at 23:30

RemoteServer Data:

Remote Server URL: https://nu.novell.com/repo
Remote Server Credentials: /Credentials/novell/update.novell.com

Download Filter Data:

The following catalogs are selected using Subscription wizard.	
Catalog Name	
SLES11-SP2-Updates	

Replication Schedule:

Schedule Type:
Day of the Week Specific

Days to run schedule:
Sunday, Monday, Tuesday, Wednesday,
Thursday, Friday, Saturday

Start time:
11:30 PM

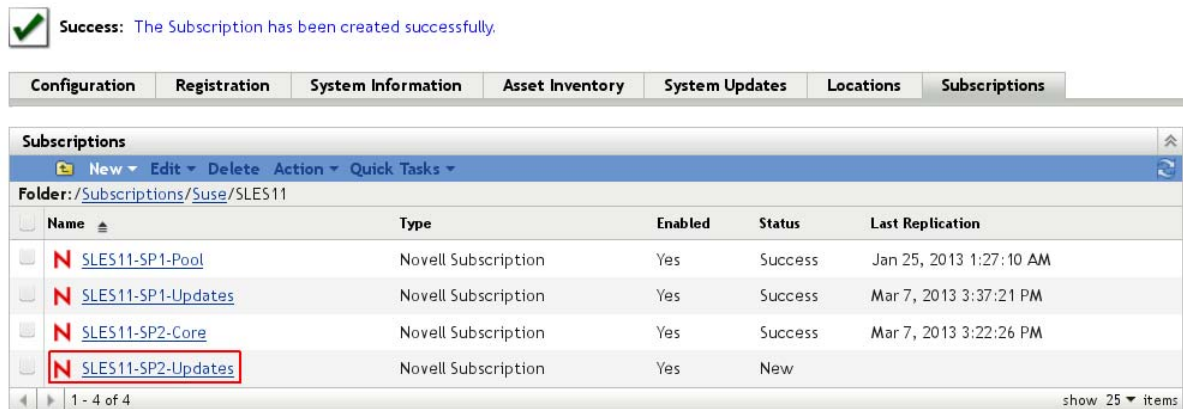
End time:

☐ Define Additional Properties

<< Back Finish Cancel

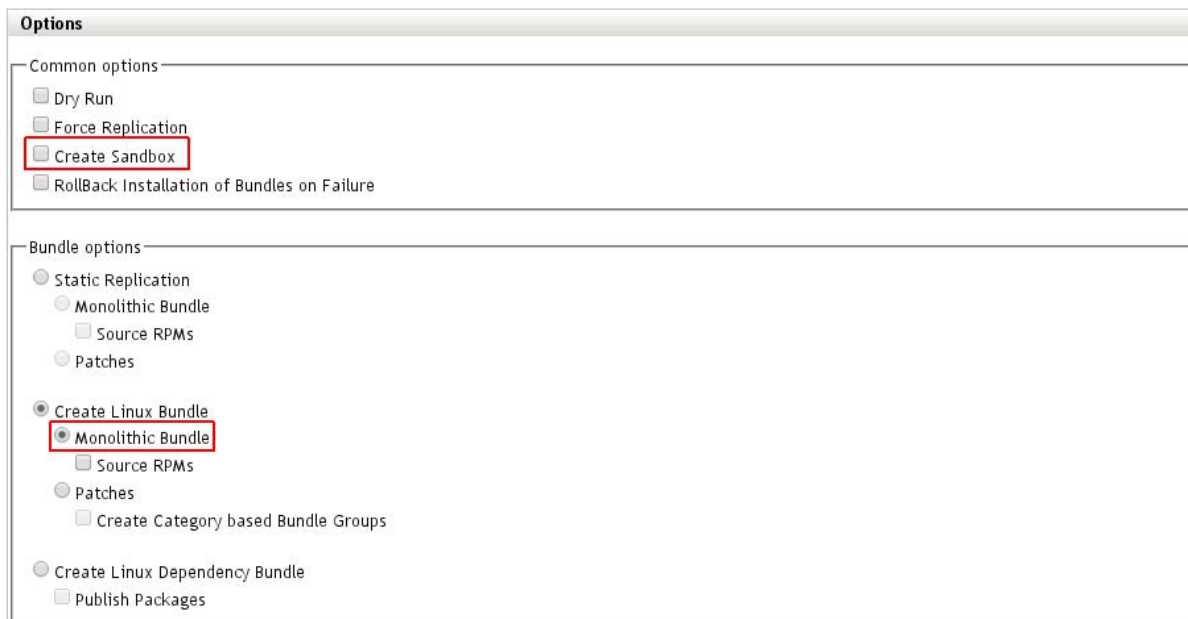
The new subscription is displayed in the following figure.

Figure 11-75 *Creating a Subscription - Subscription Has Been Created*



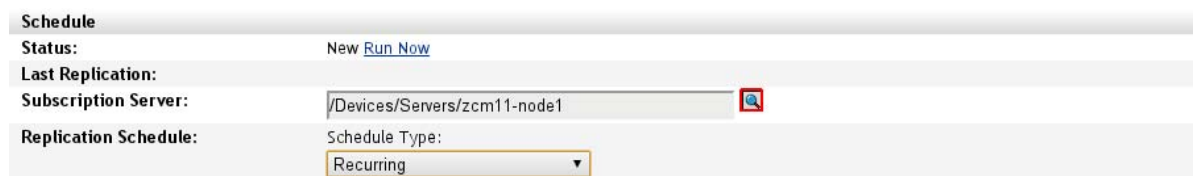
The subscription is not ready to use yet. You need to modify the bundle options of the subscription. No sandbox is needed for the bundle, so the hook must be removed. In addition, for an update bundle you must select *Monolithic Bundle* as the type.

Figure 11-76 *Creating a Subscription - Subscription Bundle Options*



A manual mirror of the subscription previously created can be initiated by clicking *Run Now*.

Figure 11-77 *Initiating a Manual Mirror*



11.5.5 Pool Channels, Core Channels, and Online Channels

In addition to patch channels, ZENworks Configuration Management requires you to download pool catalogs/channels for products to be maintained. Pool channels provide all RPMs contained on the media CD/DVD of the corresponding GA (General Availability) or service pack of a particular product for dependency resolution purposes.

Starting with SLES 11 SP2, a core channel is required for each service pack in addition to the SLES 11 SP1 pool channel. For more information, see [Section 11.1, “Overview,” on page 89](#).

Another less frequently used channel type is the online channel, which is necessary when you upgrade from one SLES/OES release or service pack to the next (this applies to SLES 10 and its OES products only, not to SLES 11).

Subscriptions for all three channel types are needed when you use the patch or upgrade functionality of ZENworks Configuration Management. This means that, in addition to the appropriate patch channels, the pool and online channels must be downloaded for every SLES/OES release and service pack. In addition, SLES 11 SP2 needs its core channel.

Pool, core, and online subscriptions differ at least in two ways from a normal patch channel subscription. First, you need to select the *Create Linux Dependency Bundle* option instead of the *Monolithic Bundle* option. Second, pool, core, and online subscriptions need to be downloaded only once and therefore need no scheduling.

The following figure illustrates how the *Option* tab for a pool catalog must be configured:

Figure 11-78 Subscriptions > Options for Pool and Online Bundles

Options

Common options

- ☐ Dry Run
- ☐ Force Replication
- ☐ Create Sandbox
- ☐ RollBack Installation of Bundles on Failure

Bundle options

- ☐ Static Replication
 - ☐ Monolithic Bundle
 - ☐ Source RPMs
 - ☐ Patches
- ☐ Create Linux Bundle
 - ☐ Monolithic Bundle
 - ☐ Source RPMs
 - ☐ Patches
 - ☐ Create Category based Bundle Groups
- ☒ Create Linux Dependency Bundle

- ☒ Publish Packages

11.6 zman

Most administrative tasks can be performed from the Web interface. However, you might need to perform some actions from the zman command line interface.

When the root account executes zman, you are asked for the administrator user and password. This can be very cumbersome when performing multiple steps with zman. For convenience, you can configure a password file (`.zmanrc`) in the home directory of the root user by issuing the following command from a shell:

```
zman asc administrator
```

An encrypted `/root/.zmanrc` file is created. It should be made read-only by executing the following command:

```
chmod 600 /root/.zmanrc
```

Password-free execution of zman instructions can be tested by executing the following command:

```
zman ql -s F
```

This command displays all failed actions from the main queue of the ZENworks Configuration Management server.

12 Managing Linux Bundles

The previous sections discussed topics such as how to organize ZENworks Configuration Management to have a maintainable object structure, how to configure ZENworks Configuration Management for subscriptions, and Novell Consulting recommendations for naming standards.

This section introduces the patching process itself. It starts with the organization of the different patch levels and details the process until the execution of the patch process at the agent level.

- ♦ [Section 12.1, “Frozen Patch Level,” on page 155](#)
- ♦ [Section 12.2, “Bundle Deployment to SLES/OES Devices,” on page 160](#)
- ♦ [Section 12.3, “YUM Repositories Derived from a Frozen Patch Level,” on page 177](#)

12.1 Frozen Patch Level

A subscription normally ensures a recurring download of a certain patch channel. It equalizes remote and local repositories by downloading only the differences. The resulting bundle objects are updated with every mirror process if the remote repository has been updated.

Production environments need stable and reliable software levels that must be provided unchanged over long time frames. In addition, applications that rely on specific kernel versions and kernel modules might require a patch level that differs from patch levels required in other server environments providing other applications.

This means that the patching components must ensure that a specific software level can be frozen at any point in time and can be provided to its targets as long as needed.

Although there isn't a specific menu within ZENworks Configuration Management to fulfill these requirements, they can be achieved easily within ZENworks Configuration Management by copying an existing update bundle and assigning properties to the copied bundle. The copy process does not increase disk space use because it only uses some internal pointers that are set within the database.

- ♦ [Section 12.1.1, “Naming Standards for Frozen Patch Levels,” on page 155](#)
- ♦ [Section 12.1.2, “Creating a Frozen Patch Level,” on page 156](#)

12.1.1 Naming Standards for Frozen Patch Levels

Novell Consulting recommends a naming scheme for copied bundles (also known as frozen bundles) that relies on following principles:

`%PRODUCT% [%VERSION% [SP%SP-LEVEL%]] - %UPDATE% - %DATE%`


Name	Description
PRODUCT	SLES OES This part of a name identifies which product the frozen bundle is for. If an object name starts with <code>SLES</code> , you need to assign it to pure SUSE Linux Enterprise Server systems (such as SLES 11) as well as to the corresponding Open Enterprise Server system (such as OES 11).

Name	Description
VERSION	2 9 10 11 This part of the name identifies the version (SLES or OES) that the frozen bundle applies to.
SP-LEVEL	GA SP1 SP2 SP3 SP4 The service pack level.
UPDATE	This string informs the administrator about the bundle purpose.
DATE	The date when the frozen bundle was copied from the original update bundle or when a frozen bundle from an earlier stage (DEV, TEST) was created and used as source for this bundle.

12.1.2 Creating a Frozen Patch Level

As an example, imagine a standard subscription for SUSE Linux Enterprise 11 SP1 servers where the *Monolithic Bundle* option is chosen, as illustrated in the next two figures.

Figure 12-1 Example Subscription


SLES11-SP1-UPDATES

Summary

Download Filter

Settings

General

Name:

SLES11-SP1-UPDATES

Type:

NU

Created By:

Administrator

GUID:

672ac812a942b3de52889361524d6ac6

Description:


[\(Edit\)](#)

Update Channel for SLES11-SP1 from nu.novell.com

Enabled:

Yes [\(Disable\)](#)

Download To Folder:



Subscription Logs:

[View Log](#)

Remote Server

Base URL:

[\(Edit\)](#)
[\(Update Certificate\)](#)

https://nu.novell.com/repo

Server Credentials:




Figure 12-2 Subscription Options for the Example

Bundle options

- Static Replication
 - Monolithic Bundle
 - Source RPMs
 - Patches
- Create Linux Bundle
 - Monolithic Bundle
 - Source RPMs
 - Patches
 - Create Category based Bundle Groups
- Create Linux Dependency Bundle
 - Publish Packages

By using this subscription, a mirror automatically creates a `SLES11-SP1-Updates` folder in the bundle menu below `Bundles/Linux/SLES11/SP1` as displayed in the following figure.

Figure 12-3 Folder Automatically Created by the Mirror Process and Based on Subscription Properties

[Bundles](#) > [Linux](#) > [SLES11](#) > **SP1**

Bundles						
Status	Name	Type	Category	Enabled	Version	Has Sandbox
	SLES11-SP1-Pool (Details)	Folder				
	SLES11-SP1-Updates (Details)	Folder				

1 - 2 of 2 show 25 items

The monolithic bundle is created by the mirror process below the `SLES11-SP1-Updates` folder.

Figure 12-4 Monolithic Bundle Created by Mirror Process

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP1](#) > **SLES11-SP1-Updates**

Bundles						
Status	Name	Type	Category	Enabled	Version	Has Sandbox
	SLES11-SP1-Updates-bundle	Linux Bundle		Yes	0	No

1 - 1 of 1 show 25 items

It contains all patches for SLES 11 SP1 64-bit that have been provided by Novell at the time of the mirror process.

To freeze this bundle, you need to execute the actions illustrated in the following figures. The SLES11-SP1-Updates bundle created and maintained by the mirror process is used as the source for a new frozen patch level.

Figure 12-5 *Creating a Frozen Bundle - Copy Source Bundle*

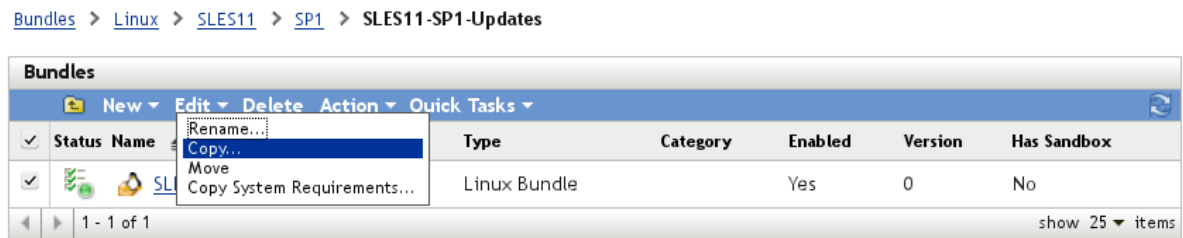
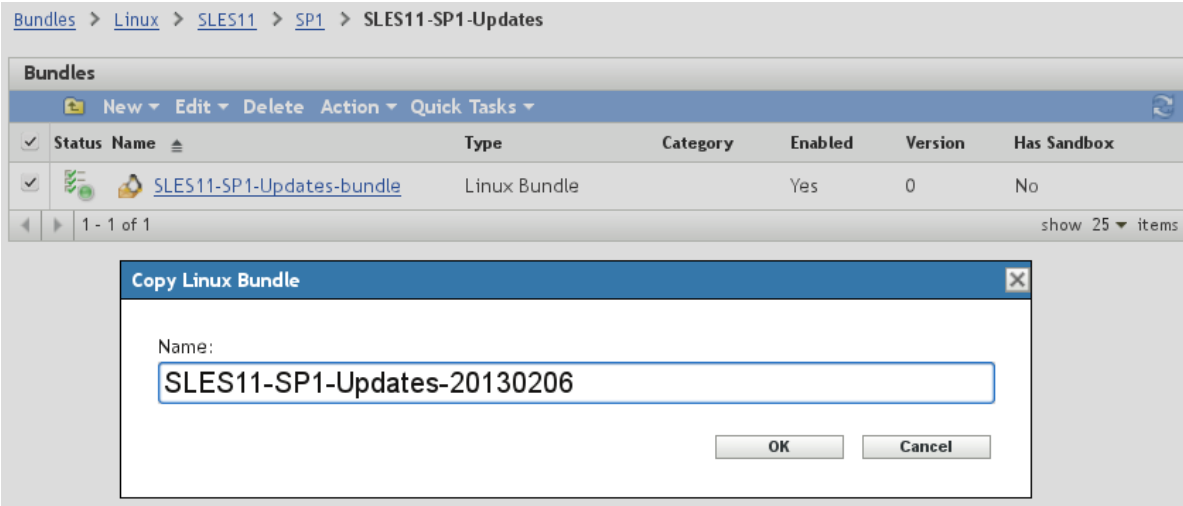


Figure 12-6 *Creating a Frozen Bundle - Assign Name to Bundle Copy*



ZENworks Configuration Management treats most object names as case-insensitive. If you need to change the case of an object name, it must be renamed first and named to the correct spelling after.

The frozen bundle is displayed in the folder view beneath the update bundle created by the mirror process, along with older frozen patch levels and the bundle groups used to assign frozen patch levels to target devices in the different environments.

Figure 12-7 Creating a Frozen Bundle - New Frozen Patch Bundle

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP1](#) > [SLES11-SP1-Updates](#)

Bundles							
New Edit Delete Action Quick Tasks							
<input type="checkbox"/>	Status	Name	Type	Category	Enabled	Version	Has Sandbox
<input type="checkbox"/>		SLES11-SP1-Updates-DEV	Bundle Group				
<input type="checkbox"/>		SLES11-SP1-Updates-PROD	Bundle Group				
<input type="checkbox"/>		SLES11-SP1-Updates-TEST	Bundle Group				
<input type="checkbox"/>		SLES11-SP1-Updates-20120706	Linux Bundle		Yes	0	No
<input type="checkbox"/>		SLES11-SP1-Updates-20121220	Linux Bundle		Yes	0	No
<input type="checkbox"/>		SLES11-SP1-Updates-20130206	Linux Bundle		Yes	0	No
<input type="checkbox"/>		SLES11-SP1-Updates-bundle	Linux Bundle		Yes	0	No
1 - 7 of 7							show 25 items

A glance at the *Packages* tab shows the packages contained in the copied bundle in alphabetical order, and also shows the properties. All properties of the original bundle remain unchanged.

Figure 12-8 Frozen Bundle - Package View

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP1](#) > [SLES11-SP1-Updates](#) > [SLES11-SP1-Updates-20130206](#)

SLES11-SP1-Updates-20130206

Displayed Version:

Summary Relationships Requirements Actions Packages Settings								
RPMs								
Edit Remove								
<input type="checkbox"/>	Name	Epoch	Version	Release	Architecture	Targets	Freshen	Install Type
<input type="checkbox"/>	a2ps	0	4.13	1326.35.1	x86_64	sles-11-x86_64	Yes	Auto
<input type="checkbox"/>	aaa_base	0	11	6.46.46.2	x86_64	sles-11-x86_64	Yes	Auto
<input type="checkbox"/>	acct	0	6.3.5	817.25.1	x86_64	sles-11-x86_64	Yes	Auto
<input type="checkbox"/>	acpid	0	1.0.6	91.16.1	x86_64	sles-11-x86_64	Yes	Auto
<input type="checkbox"/>	alsa	0	1.0.18	16.24.1	x86_64	sles-11-x86_64	Yes	Auto
<input type="checkbox"/>	alsa-docs	0	1.0.18	16.24.1	x86_64	sles-11-x86_64	Yes	Auto
<input type="checkbox"/>	alsa-plugins	0	1.0.18	7.12.23	x86_64	sles-11-x86_64	Yes	Auto
<input type="checkbox"/>	alsa-plugins-pulse	0	1.0.18	7.12.23	x86_64	sles-11-x86_64	Yes	Auto
<input type="checkbox"/>	alsa-plugins-pulse-32bit	0	1.0.18	7.12.23	x86_64	sles-11-x86_64	Yes	Auto
<input type="checkbox"/>	apache2	0	2.2.12	1.30.1	x86_64	sles-11-x86_64	Yes	Auto
1 - 10 of 1,075							show 10 items	

The frozen bundle is ready for assignment and will never change automatically, even with the next mirror cycle.

This procedure must be executed for all products and their respective service packs to gain the advantage of frozen patch bundles.

12.2 Bundle Deployment to SLES/OES Devices

The frozen patch level created in the previous section can now be assigned and deployed to all relevant machines. However, a SLES/OES server usually needs additional bundles assigned. The assignment and deployment of standard patch items needed in all SLES/OES environments is described in the following sections.

- ♦ [Section 12.2.1, “Bundle Assignment,” on page 160](#)
- ♦ [Section 12.2.2, “Manual Bundle Deployment,” on page 168](#)
- ♦ [Section 12.2.3, “Scheduled Bundle Deployment,” on page 175](#)

12.2.1 Bundle Assignment

Novell Consulting recommends that you perform all bundle assignments tasks for Linux devices to device groups instead of single devices or device folders. This makes addition and removal of assignments easier, and the central concepts of simplicity and clarity within ZENworks Configuration Management administration are ensured.

Configuration bundles are collected in bundle groups that are assigned to device groups with identical configuration requirements (see [“The BUNDLE-GROUPS Folder” on page 126](#)).

Each *Update bundle* is a member of a bundle group that is assigned to all device groups whose members are on the same product, version, and support pack, such as all OES 11 SP1 servers (see [Section 11.3.2, “Server Group Objects in the Device Menu,” on page 96](#)).

Because *Pool bundles* and *Core bundles* required for dependency resolution never change, there is no benefit in assigning them through bundle groups and these bundles are directly assigned to the same device groups as the corresponding update bundle groups.

The following bundles and bundle groups need to be assigned to each SLES 11 SP2 device in the production environment:

- ♦ The bundle group `SLES11-SP1-Updates-PROD`
- ♦ The Pool bundle for SLES 11 SP1
- ♦ The bundle group `SLES11-SP2-Updates-PROD`
- ♦ The Core bundle for SLES 11 SP2

In addition, each production OES 11 SP1 device requires the following assignments:


- ♦ The bundle group `OES11-SP1-Updates-PROD`
- ♦ The Pool bundle for OES 11 SP1

For an OES 11 SP1 device in production, you first need to create three bundle groups, one for each of the three update bundles listed above.


Assign the new bundle group a name adhering to the suggested naming standard and provide a description.

Figure 12-9 Creating a Bundle Group for SLES 11 SP 2 Devices - Basic Information

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP2](#) > [SLES11-SP2-Updates](#) > **Create New Group**

Create New Group
 **Step 1: Basic Information**

Group Name: *

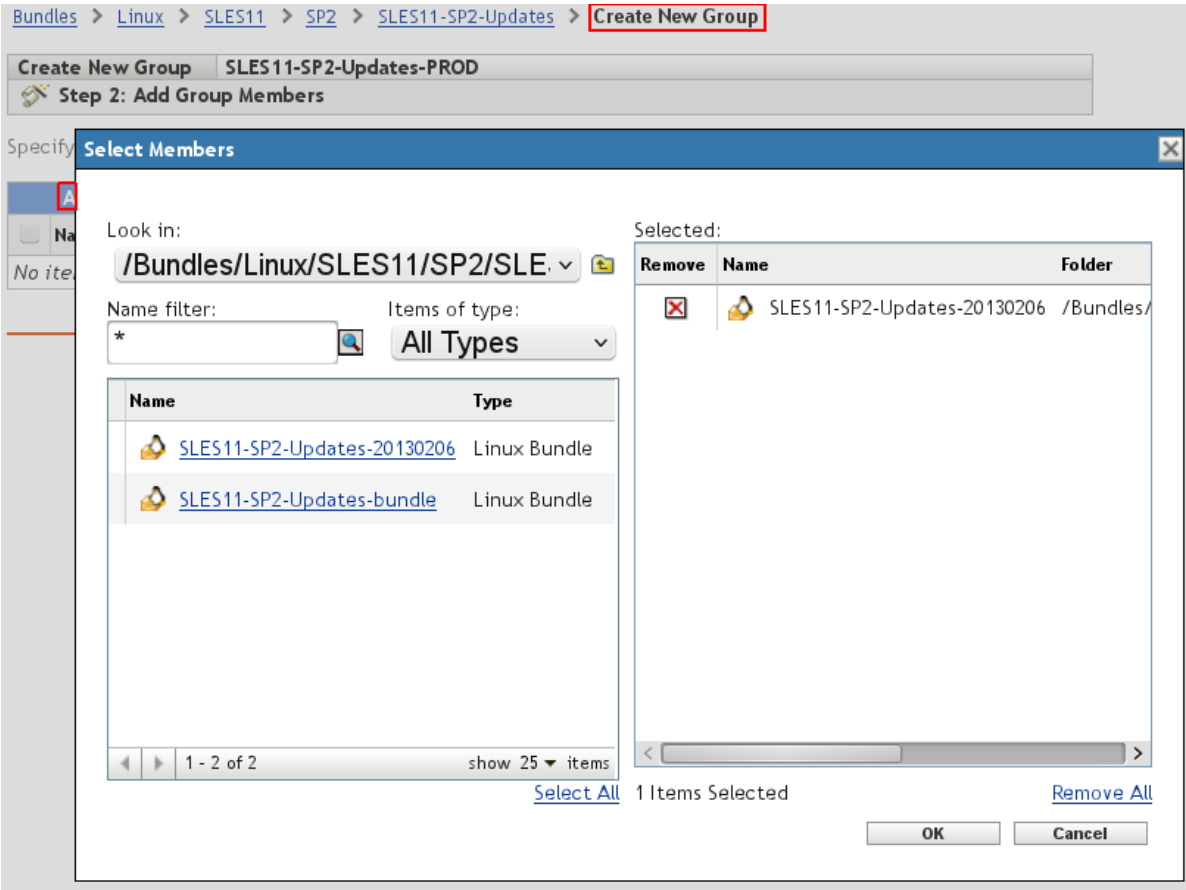
Folder: *


Description:

Fields marked with an asterisk are required.

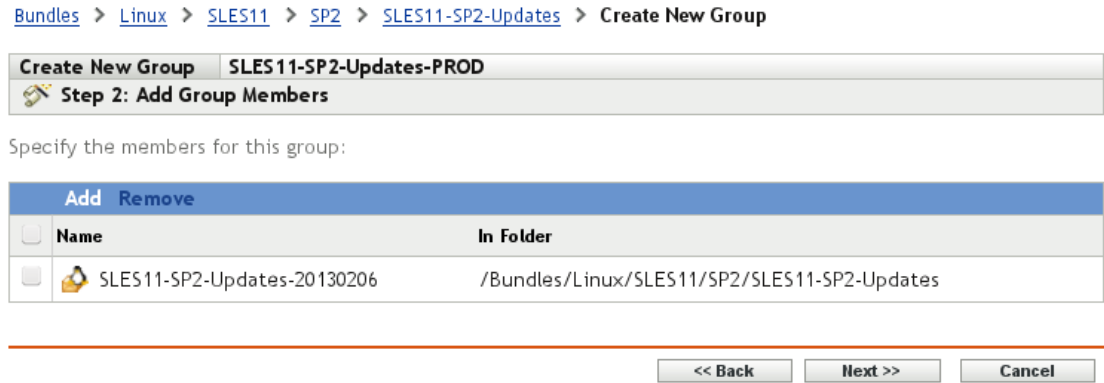
In the next dialog window, select the appropriate group member (SLES 11 SP2 update bundle) by clicking *Add*.

Figure 12-10 *Creating a Bundle Group Needed by SLES 11 SP2 Devices - Select Members*



The update bundle created on February 6, 2013 has been added to the group in this example:

Figure 12-11 *Summary of All Bundle Group Members*



A final summary screen is presented. If the *Define Additional Properties* check box was selected, you can now execute the assignment step.

Figure 12-12 Summary of Bundle Group Creation

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP2](#) > [SLES11-SP2-Updates](#) > [Create New Group](#)

Create New Group

SLES11-SP2-Updates-PROD

Step 3: Summary

Review the information and click "Finish" to create the new Group

Folder: /Bundles/Linux/SLES11/SP2/SLES11-SP2-Updates
Group Name: SLES11-SP2-Updates-PROD
Description: Group for assignment of frozen SLES11-SP2 patch levels

Members: [/Bundles/Linux/SLES11/SP2/SLES11-SP2-Updates/SLES11-SP2-Updates-20130206](#)

☒ Define Additional Properties

<< Back

Finish

Cancel

Click *Add* on the bundle group summary page to select the desired device groups. For this example, the `PROD_OES11SP1` and the `PROD_SLES11SP2` device groups have been selected.

Figure 12-13 Device Group Assignment To a Bundle Group - Device Group Selection

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP2](#) > [SLES11-SP2-Updates](#) > [SLES11-SP2-Updates-PROD](#) > [Assign Bundle](#)

Assign Bundle

Step 1: Devices to be Assigned

Select the devices to be assigned to the previously selected bundle (/Bundles/Linux/SLES11/SP2/SLES11-SP2-Updates/SLES11-SP2-Updates-PROD).

Select Objects

Look in:
/Devices/Servers/LINUX/GROUP

Name filter:
*

Items of type:
All Types

Name	Type
PROD_OES11SP1	Server Group
PROD_OES25P3	Server Group
PROD_SLES10SP3	Server Group
PROD_SLES10SP4	Server Group
PROD_SLES11SP1	Server Group
PROD_SLES11SP2	Server Group

1 - 6 of 6

show 25 items

Select All

Selected:

Remove	Name	Folder
<input checked="" type="checkbox"/>	PROD_OES11SP1	/Devices/Servers/LINUX/GROUP
<input checked="" type="checkbox"/>	PROD_SLES11SP2	/Devices/Servers/LINUX/GROUP

2 Items Selected

Remove All

OK

Cancel

The advantage of a clear folder structure is obvious in this figure. The folder contains only production device groups to update. No single device objects or irrelevant test or configuration groups are interfering with object selection.

The shortcut locations displayed in the next figure are not of any interest for Linux servers. Enabling or disabling them has no effect on the target devices. Novell Consulting recommends that you deselect all check boxes on this page.

Figure 12-14 Device Group Assignment To a Bundle Group - Shortcut Locations Deselected

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP2](#) > [SLES11-SP2-Updates](#) > [SLES11-SP2-Updates-PROD](#) > **Assign Bundle**

Assign Bundle

Step 1: Devices to be Assigned

Select the devices to be assigned to the previously selected bundle (/Bundles/Linux/SLES11/SP2/SLES11-SP2-Updates/SLES11-SP2-Updates-PROD).

Add Remove	
<input type="checkbox"/> Name	In Folder
<input type="checkbox"/> PROD_OES11SP1	/Devices/Servers/LINUX/GROUPS/PROD/UPDATE
<input type="checkbox"/> PROD_SLES11SP2	/Devices/Servers/LINUX/GROUPS/PROD/UPDATE

Shortcut Location

Select the locations to place the shortcut for the bundle.

☐ Application Window ☐ Desktop ☐ Start Menu
☐ Quick Launch ☐ System Tray

The next page is important when the distribution or installation of bundles must be scheduled. Three different scheduling types are available. Only the Distribution Schedule and the Availability Schedule are of interest for Linux devices.

With a Distribution Schedule, you can pre-distribute software bundles to remote devices. For example, assume that you have a large amount of data that must be transported via a slow WAN link. In this case, the Distribution Schedule ensures that the data is delivered to remote devices within the desired time, such as one week. It can then be installed manually or automatically.

The Availability Schedule is a task where device agents are informed at a predefined point in time that new ZENworks Configuration Management bundle objects have been assigned and are ready to be installed. Both scheduling tasks can be modified later.

Figure 12-15 Device Group Assignment To a Bundle Group - Schedule Configuration

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP2](#) > [SLES11-SP2-Updates](#) > [SLES11-SP2-Updates-PROD](#) > **Assign Bundle**

Assign Bundle

Step 2: Schedules and flags

Select one or more of the optional schedules that you would like to configure. Bundle actions can be scheduled along with the time window during which the bundle will be available to users.

☐ Distribution Schedule
☐ Launch Schedule
☐ Availability Schedule

☐ Attempt a dry run (Useful for Linux Bundles)

Clicking *Finish* summarizes the bundle details before the assignment can be completed.

Figure 12-16 Device Group Assignment To a Bundle Group - Selection Overview

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP2](#) > [SLES11-SP2-Updates](#) > [SLES11-SP2-Updates-PROD](#) > **Assign Bundle**

Assign Bundle

Step 3: Finish

Click "Finish" to create the new assignments.

Bundles: /Bundles/Linux/SLES11/SP2/SLES11-SP2-Updates/SLES11-SP2-Updates-PROD

Assigned Objects: /Devices/Servers/LINUX/GROUPS/PROD/UPDATE/PROD_OES11SP1
 /Devices/Servers/LINUX/GROUPS/PROD/UPDATE/PROD_SLES11SP2

<< Back

Finish

Cancel

The bundle group assigning the SLES 11 SP2 frozen patch level to all SLES 11 SP2 devices and OES 11 SP1 devices in production has been created successfully.

Figure 12-17 Device Group Assignment To a Bundle Group - Success

Success: The assignment was successful.

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP2](#) > [SLES11-SP2-Updates](#) > **SLES11-SP2-Updates-PROD**

SLES11-SP2-Updates-PROD

Summary

General

Object type:

Bundle Group

GUID:

43ea999f9f9e2a139d390d3e17955096

Description:

[\(Edit\)](#)

Group for assignment of frozen SLES11-SP2 patch levels

Members

Add Remove Move Up Move Down

<input type="checkbox"/>	Name	In Folder
<input type="checkbox"/>	SLES11-SP2-Updates-20130206	/Bundles/Linux/SLES11/SP2/SLES11-SP2-Updates

1 - 1 of 1

show 10 ▼ items

YUM Service Details

YUM Service:

[\(Create\)](#)

(No YUM Service)

Device Assignments

Add Remove

<input type="checkbox"/>	Name	In Folder	Details
<input type="checkbox"/>	PROD_OES11SP1	/Devices/Servers/LINUX/GROUPS/PROD/UPDATE	Assignment Details
<input type="checkbox"/>	PROD_SLES11SP2	/Devices/Servers/LINUX/GROUPS/PROD/UPDATE	Assignment Details

1 - 2 of 2

show 5 ▼ items

The new bundle group object can be seen in the following figure:

Figure 12-18 Final Bundle Group Object View

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP2](#) > [SLES11-SP2-Updates](#)

Bundles							
New Edit Delete Action Quick Tasks							
<input type="checkbox"/>	Status	Name	Type	Category	Enabled	Version	Has Sandbox
<input type="checkbox"/>		SLES11-SP2-Updates-DEV	Bundle Group				
<input type="checkbox"/>		SLES11-SP2-Updates-PROD	Bundle Group				
<input type="checkbox"/>		SLES11-SP2-Updates-TEST	Bundle Group				
<input type="checkbox"/>		SLES11-SP2-Updates-20130206	Linux Bundle		Yes	0	Yes
<input type="checkbox"/>		SLES11-SP2-Updates-bundle	Linux Bundle		Yes	3	No
1 - 5 of 5							show 25 items

The assignment process now needs to be replicated for the SLES 11 SP1 updates as well as for the OES 11 SP1 updates. In addition, the core and pool bundles also need to be assigned to the same device groups as the SLES11-SP2-Updates-PROD bundle group.

Whenever you need to assign multiple objects to a device or device group, it is more efficient to start the process from the *Relationships* tab of the device / device group object then from the different objects you need to assign to the devices. The following figures illustrate this approach for the assignment of the outstanding bundles and bundle groups to the PROD_OES11SP1 device group.

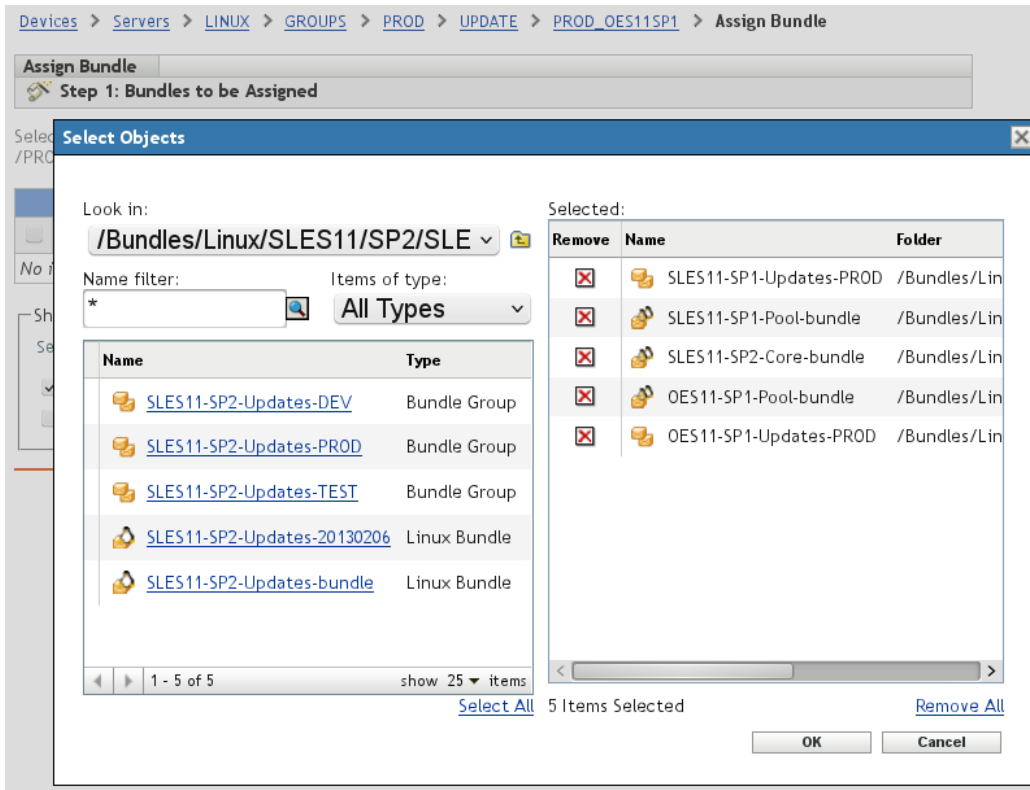
Select the *Relationships* tab of the PROD_OES11SP1 device group in the /Devices/Servers/Linux/GROUPS/PROD/UPDATE folder and then select *Add* to add the missing bundles and bundle groups to the PROD_OES11SP1 device group.

Figure 12-19 Bundle Assignment To a Device Group

[Devices](#) > [Servers](#) > [LINUX](#) > [GROUPS](#) > [PROD](#) > [UPDATE](#) > [PROD_OES11SP1](#)

PROD_OES11SP1			
Summary	Relationships	Patches	
Assigned Bundles			
Direct		Inherited	All
Add Remove Move Up Move Down			
Name	Type	Source	Details
SLES11-SP2-Updates-PROD	Bundle Group	PROD_OES11SP1	Assignment Details
SLES11-SP2-Updates-20130206	Linux Bundle		

Figure 12-20 Bundle Assignment To a Device Group - Selecting Bundles And Bundle Groups



The following figures provides and overview of the six bundles that have been assigned to this device group, either directly or through bundle groups for the update bundles.

Figure 12-21 Bundle Assignment To a Device Group - Overview of Assigned Bundles

Devices > Servers > LINUX > GROUPS > PROD > UPDATE > PROD_OES11SP1

PROD_OES11SP1

Summary Relationships Patches

Assigned Bundles			
Direct		Inherited	All
Add	Remove	Move Up	Move Down
Name	Type	Source	Details
<input checked="" type="checkbox"/> SLES11-SP1-Updates-PROD	Bundle Group	PROD_OES11SP1	Assignment Details
<input checked="" type="checkbox"/> SLES11-SP1-Updates-bundle-20130228	Linux Bundle		
<input checked="" type="checkbox"/> SLES11-SP1-Pool-bundle	Linux Dependency Bundle	PROD_OES11SP1	Assignment Details
<input checked="" type="checkbox"/> SLES11-SP2-Core-bundle	Linux Dependency Bundle	PROD_OES11SP1	Assignment Details
<input checked="" type="checkbox"/> OES11-SP1-Pool-bundle	Linux Dependency Bundle	PROD_OES11SP1	Assignment Details
<input checked="" type="checkbox"/> OES11-SP1-Updates-PROD	Bundle Group	PROD_OES11SP1	Assignment Details
<input checked="" type="checkbox"/> OES11-SP1-Updates-bundle-20130125	Linux Bundle		
<input checked="" type="checkbox"/> SLES11-SP2-Updates-PROD	Bundle Group	PROD_OES11SP1	Assignment Details
<input checked="" type="checkbox"/> SLES11-SP2-Updates-20130206	Linux Bundle		

Assigned Policies		
Direct		All
Add	Remove	Move Up
Name	Type	Source

Typically a new frozen patch level is first assigned to development devices through the corresponding bundle groups.

After successful initial testing, the next step is to replace the current update bundle in the bundle groups for the test environment with the new frozen patch level.

Finally, after additional testing has not revealed any issues, the new frozen patch level will also be added to the bundle groups for the production.

New frozen patch levels can be created in parallel at any time and the whole cycle can be repeated. Old frozen patch bundles that are no longer a member of any bundle group should be kept for a while, so that the corresponding patch levels can be re-activated if the need should arise.

12.2.2 Manual Bundle Deployment

Deployment of the bundles assigned in [Section 12.2.1, “Bundle Assignment,” on page 160](#) can be achieved in several ways. The simplest form is an assignment without scheduling:

Figure 12-22 Simple Assignment without Scheduling

Bundles > Linux > OES11 > GA > Assign Bundle

Assign Bundle

Step 3: Schedules and flags

Select one or more of the optional schedules that you would like to configure. Bundle actions can be scheduled along with the time window during which the bundle will be available to users.

☐ Distribution Schedule

☐ Launch Schedule

☐ Availability Schedule

☐ Attempt a dry run (Useful for Linux Bundles)

<< Back Next >> Cancel

This method of assignment is recommended by Novell Consulting if no more than 20–30 devices must be managed by ZENworks Configuration Management. The patching process itself must be initiated manually on each device by an administrator.

First, look at an example of how a ZENworks Configuration Management agent displays an assignment of a newly created bundle group. To have an up-to-date agent view, you need to execute the following command at the device:

```
zac ref
```

The `zac bl` (bundle list) command lists all assigned bundles and their states. The bundle groups are not displayed:

Figure 12-23 Agent View of the Bundle Group Assignment

```
edir04:/etc/opt/novell/zenworks # zac bl
```

Processing Command: bl

Display Name	Version	Bundle Type	Status	Assigned
-----+-----+-----+-----+-----				
OES11-Pool-bundle	0	Linux Dependency Bundle	Available	Device
OES11-SP1-UPDATE-20120224-PROD	1	Linux Bundle	Downloaded	Device
SLES11-SP1-Pool-bundle	0	Linux Dependency Bundle	Available	Device
SLES11-SP1-UPDATE-20120224-PROD	1	Linux Bundle	Downloaded	Device
-----+-----+-----+-----+-----				
Total Bundles Assigned: 4				

```
edir04:/etc/opt/novell/zenworks #
```

There are two different states displayed in the fourth column in [Figure 12-23](#). A status of `Available` means the bundle is installed. The status for dependency bundles is always `Available`, even if a package has not been installed to the device from these bundles. This explains why this status is displayed immediately after the first agent refresh after the dependency bundle has been assigned.

The term `Downloaded` does not mean that the bundle content is already present on the target device. It means that all metadata, such as package lists and dependencies, is known to the agent.

Available updates are listed by executing the following command:

```
zac lu (list updates)
```

Figure 12-24 Listing Updates through zac lu

```
edir04:/etc/opt/novell/zenworks # zac lu
```

Processing Command: lu

Source Name	Source Type	Package Name	
Version	Old Version		Arch
-----+-----+-----+-----			
SLES11-SP1-UPDATE-20120224-PROD	ZenBundle	a2ps	
4.13-1326.35.1:0	4.13-1326.33:0		x86_64
SLES11-SP1-UPDATE-20120224-PROD	ZenBundle	aaa_base	
11-6.46.42.1:0	11-6.28.5:0		x86_64
SLES11-SP1-UPDATE-20120224-PROD	ZenBundle	acpid	
1.0.6-91.16.1:0	1.0.6-91.6:0		x86_64
SLES11-SP1-UPDATE-20120224-PROD	ZenBundle	alsa-plugins-pulse	
1.0.18-7.12.23:0	1.0.18-7.5:0		x86_64
SLES11-SP1-UPDATE-20120224-PROD	ZenBundle	apache2	
2.2.12-1.30.1:0	2.2.10-2.24.5:0		x86_64
SLES11-SP1-UPDATE-20120224-PROD	ZenBundle	apache2-doc	
2.2.12-1.30.1:0	2.2.10-2.24.5:0		x86_64

Updates from a particular bundle are displayed by using the following command:

```
zac lu <bundle name>
```

The output is shown in the following figure:

Figure 12-25 Updates for a Particular Frozen Patch Bundle

```
edir04:/var/opt/novell/zenworks # zac lu OES11-UPDATE-20120224-PROD

Verarbeitung von Kommando: lu

Quellenname          | Quelltyp   | Paketname
| Version            | Alte Version | Arch
-----+-----+-----
OES11-UPDATE-20120224-PROD | ZenBundle | novell-dclient
| 8.8.6.5-3.14:0      | 8.8.6.4-1.13:0 | x86_64
OES11-UPDATE-20120224-PROD | ZenBundle | novell-dclient-32bit
| 8.8.6.5-3.14:0      | 8.8.6.4-1.13:0 | x86_64
OES11-UPDATE-20120224-PROD | ZenBundle | novell-edirectory-jc1nt
| 8.8.6.5-3.29:0      | 8.8.6.4-3.15:0 | x86_64
OES11-UPDATE-20120224-PROD | ZenBundle | novell-edirectory-tsands
| 8.8.6.5-3.29:0      | 8.8.6.4-3.15:0 | x86_64
OES11-UPDATE-20120224-PROD | ZenBundle | novell-edirectory-tsands-32bit
| 8.8.6.5-3.29:0      | 8.8.6.4-3.15:0 | x86_64
OES11-UPDATE-20120224-PROD | ZenBundle | novell-migration
| 1.0.3-4.472:0       | 1.0.3-4.165:0  | x86_64
OES11-UPDATE-20120224-PROD | ZenBundle | novell-NDSbase
| 8.8.6.5-3.29:0      | 8.8.6.4-3.15:0 | x86_64
OES11-UPDATE-20120224-PROD | ZenBundle | novell-NDSbase-32bit
| 8.8.6.5-3.29:0      | 8.8.6.4-3.15:0 | x86_64
```

Novell Consulting recommends that you initiate a particular update by executing the `zac bundle-install (zac bin)` command:

```
zac bin <bundle_name>
```

Although a server update can be achieved by executing `zac update (zac up)`, using `zac bin` gives a better overview of whether a particular bundle has been installed.

The following figure illustrates the different stages in the patch process. In the first stage, the patches are distributed from the ZCM server to the device. The installation takes place in the second stage.

Figure 12-26 Example Installation of a Frozen Patch Bundle with `zac bin`

```
edir04:/etc/opt/novell/zenworks # zac bin SLES11-SP1-UPDATE-20120224-PROD

Processing Command: bin

Starting Distribution of SLES11-SP1-UPDATE-20120224-PROD

Finished Distributing SLES11-SP1-UPDATE-20120224-PROD

Starting Install of SLES11-SP1-UPDATE-20120224-PROD

Finished Installing SLES11-SP1-UPDATE-20120224-PROD

edir04:/etc/opt/novell/zenworks # █
```

The new bundle view is seen in the following figure, where `zac bl` has been executed again after installation of the `SLES11-SP1-UPDATE-20120224-PROD` bundle. All installed bundles are now flagged as Available. The `OES11-UPDATE-20120224-PROD` bundle still remains in the Downloaded state.

Figure 12-27 Bundle View after Installation of a Frozen Bundle

```
edir04:~ # zac bl
```

Processing Command: bl

Display Name	Version	Bundle Type	Status	Assigned
OES11-Pool-bundle	0	Linux Dependency Bundle	Available	Device
OES11-UPDATE-20120224-PROD	5	Linux Bundle	Downloaded	Device
SLES11-SP1-Pool-bundle	0	Linux Dependency Bundle	Available	Device
SLES11-SP1-UPDATE-20120224-PROD 1		Linux Bundle	Available	Device

Total Bundles Assigned: 4

```
edir04:~ #
```

You can also use `zac up` for installing patches from a frozen bundle. The example in this section uses the frozen OES 11 update bundle.

You can use the `zac list-updates (zac lu)` command to verify if there are still patches that have not yet been applied to the device. In the following example, there are still uninstalled patches because the previous patch update did not include the OES 11 patches:

Figure 12-28 Updates Listed after Applying SLES 11 SP1 Patches

```
edir04:~ # zac lu
```

Processing Command: lu

Source Name	Source Type	Package Name
Version Old Version Arch		
OES11-UPDATE-20120224-PROD	ZenBundle	novell-dclient
8.8.6.5-3.14:0 8.8.6.4-1.13:0 x86_64		
OES11-UPDATE-20120224-PROD	ZenBundle	novell-dclient-32bit
8.8.6.5-3.14:0 8.8.6.4-1.13:0 x86_64		
OES11-UPDATE-20120224-PROD	ZenBundle	novell-edirectory-jcInt
8.8.6.5-3.29:0 8.8.6.4-3.15:0 x86_64		
OES11-UPDATE-20120224-PROD	ZenBundle	novell-edirectory-tsands
8.8.6.5-3.29:0 8.8.6.4-3.15:0 x86_64		
OES11-UPDATE-20120224-PROD	ZenBundle	novell-edirectory-tsands-32bit
8.8.6.5-3.29:0 8.8.6.4-3.15:0 x86_64		
OES11-UPDATE-20120224-PROD	ZenBundle	novell-migration
1.0.3-4.472:0 1.0.3-4.165:0 x86_64		

The `zac update (zac up)` command also allows you to optionally specify one or more bundles to be processed, as shown in the next figure:

Figure 12-29 Patching by Using `zac up` (1)

```
edir04:~ # zac up OES11-UPDATE-20120224-PROD

Processing Command: up

Following 21 package(s) will be installed.

novell-NDSbase-32bit-8.8.6.5-3.29.x86_64 (OES11-UPDATE-20120224-PROD)
novell-oes-samba-libsmbclient0-3.4.3-1.36.17.x86_64
(OES11-UPDATE-20120224-PROD)
novell-oes-samba-libtalloc1-32bit-3.4.3-1.36.17.x86_64
(OES11-UPDATE-20120224-PROD)
novell-NOVLice-8.8.6.5-3.29.x86_64 (OES11-UPDATE-20120224-PROD)
novell-edirectory-jcInt-8.8.6.5-3.29.x86_64 (OES11-UPDATE-20120224-PROD)

novell-NDSbase-8.8.6.5-3.29.x86_64 (OES11-UPDATE-20120224-PROD)
novell-oes-samba-libtdb1-32bit-3.4.3-1.36.17.x86_64
(OES11-UPDATE-20120224-PROD)
novell-oes-samba-libsmbsharemodes0-3.4.3-1.36.17.x86_64
(OES11-UPDATE-20120224-PROD)
novell-migration-1.0.3-4.472.x86_64 (OES11-UPDATE-20120224-PROD)
novell-NDScommon-8.8.6.5-3.29.x86_64 (OES11-UPDATE-20120224-PROD)
novell-NOVLembox-8.8.6.5-3.90.x86_64 (OES11-UPDATE-20120224-PROD)
sles-release-11.1-3.1.x86_64 (OES11-UPDATE-20120224-PROD)
novell-edirectory-tsands-32bit-8.8.6.5-3.29.x86_64
(OES11-UPDATE-20120224-PROD)
novell-sasl-gssapi-method-2.8.3.3-4.1.x86_64 (OES11-UPDATE-20120224-PROD)

novell-NOVLsnmp-8.8.6.5-3.16.x86_64 (OES11-UPDATE-20120224-PROD)
novell-edirectory-tsands-8.8.6.5-3.29.x86_64 (OES11-UPDATE-20120224-PROD)

novell-NDSimon-8.8.6.5-3.124.x86_64 (OES11-UPDATE-20120224-PROD)
novell-NOVLsubag-8.8.6.5-3.16.x86_64 (OES11-UPDATE-20120224-PROD)
novell-dclient-8.8.6.5-3.14.x86_64 (OES11-UPDATE-20120224-PROD)
novell-dclient-32bit-8.8.6.5-3.14.x86_64 (OES11-UPDATE-20120224-PROD)
novell-NDSserv-8.8.6.5-3.29.x86_64 (OES11-UPDATE-20120224-PROD)

Continue Yes/No [N]:
```

In contrast to `zac bin`, using `zac up` does display the individual patch RPMs being processed:

Figure 12-30 Patching by Using `zac up` (2)

```
Continue Yes/No [N]:Yes
Downloading novell-oes-samba-libsmbclient0 [100%] Done
Downloading novell-oes-samba-libsmbclient0 [100%] Done
Installing sles-release-11.1-3.1.x86_64.rpm [100%] Done

21 packages installed 0 packages removed.
```

However, the bundle listing still shows that the bundle is not installed. Although all packages that have been installed are part of the frozen OES 11 patch bundle, it still remains in the `Downloaded` state.

Figure 12-31 Bundle View after `zac up OES11-SP1-UPDATE-20120224-PROD`

```
edir04:~ # zac bl

Processing Command: bl

Display Name          Version Bundle Type      Status      Assigned
-----+-----+-----+-----+-----
OES11-Pool-bundle     0      Linux Dependency Bundle   Available   Device
OES11-UPDATE-20120224-PROD 5      Linux Bundle             Downloaded  Device
SLES11-SP1-Pool-bundle 0      Linux Dependency Bundle   Available   Device
SLES11-SP1-UPDATE-20120224-PROD 1      Linux Bundle             Available   Device
-----+-----+-----+-----+-----
Total Bundles Assigned: 4
edir04:~ #
```

For this reason, Novell Consulting strongly recommends using `zac bin` as shown in the following figure. The commands displayed there have been executed after `zac up` to obtain the correct state of the bundle view.

Figure 12-32 After Correcting the Frozen OES 11 Patch Bundle

```
edir04:~ # zac bin OES11-UPDATE-20120224-PROD

Processing Command: bin

Starting Distribution of OES11-UPDATE-20120224-PROD

Finished Distributing OES11-UPDATE-20120224-PROD

Starting Install of OES11-UPDATE-20120224-PROD

Finished Installing OES11-UPDATE-20120224-PROD

edir04:~ # zac bl

Processing Command: bl
```

Display Name	Version	Bundle Type	Status	Assigned
OES11-Pool-bundle	0	Linux Dependency Bundle	Available	Device
OES11-UPDATE-20120224-PROD	5	Linux Bundle	Available	Device
SLES11-SP1-Pool-bundle	0	Linux Dependency Bundle	Available	Device
SLES11-SP1-UPDATE-20120224-PROD	1	Linux Bundle	Available	Device

```
-----+-----+-----+-----+-----
Total Bundles Assigned: 4
edir04:~ #
```

No real action happens, so no software is installed at the device. However, the state of the bundle changes from the agent point of view and is updated from `Downloaded` to `Available`.

12.2.3 Scheduled Bundle Deployment

Manual patch deployments are not suitable if large numbers of devices must be managed via ZENworks Configuration Management. For such environments, bundle assignments can be scheduled either for a particular date and time, a random time, after the next device refresh, or a time relative to the next device refresh.

The following two figures illustrate the use of an Availability Schedule for automatic bundle deployment. At the scheduled time, the agent retrieves and installs all assigned bundles.:

Figure 12-33 Schedules and Flags

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP2](#) > [SLES11-SP2-Updates](#) > [Assign Bundle](#)

Assign Bundle
 Step 2: Schedules and flags

Select one or more of the optional schedules that you would like to configure. Bundle actions can be scheduled along with the time window during which the bundle will be available to users.

☐ Distribution Schedule
☐ Launch Schedule
☒ Availability Schedule
☐ Attempt a dry run (Useful for Linux Bundles)

<< Back Next >> Cancel

Figure 12-34 Bundle Availability Schedule

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP2](#) > [SLES11-SP2-Updates](#) > [Assign Bundle](#)

Assign Bundle
 Step 3: Bundle Availability Schedule

Configure the time period when the bundle should be made available to run on managed devices.

Schedule Type:
Date Specific ▾

Start Date(s): *

Start Time: : End Time: :

☐ Use Coordinated Universal Time (Current UTC 7:31 PM)

<< Back Next >> Cancel

Another schedule type for automatic patch deployments is the distribution schedule. This schedule type is primarily intended to predistribute larger bundles to the target devices; for example, if a slow WAN link prevents a direct installation. The bundle content is preloaded to the devices and installed from a local cache directory:

There are many possibilities for specifying a particular point in time when the distribution should be started. The Recurring type has been chosen for the following example:

Figure 12-35 *Schedule Types for Distribution Schedules*

Schedule Type:

- No Schedule
- Date Specific
- Recurring**
- Event

fresh: 0 Days 0 Hours 0 Minutes

The option to start this distribution immediately after the next device refresh is also selected:

Figure 12-36 *Distribution Schedule*

Schedule Type: Recurring

When a device is refreshed

☐ Delay execution after refresh: 0 Days 0 Hours 0 Minutes

Days of the week

Sun	Mon	Tue	Wed	Thu	Fri	Sat
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Start Time: 1 : 00

[More Options](#)

Monthly

☒ Day of the month: 1

☐ Last day of the month

☐ First Sunday

Start Time: 1 : 00

[More Options](#)

Fixed Interval

0 Months 0 Weeks 0 Days 0 Hours 0 Minutes

Start Date: 5/21/2012 Start Time: 1 : 00

[More Options](#)

☐ Wake-on-LAN (Applies to Devices only) [Options](#)

☒ **Install Immediately after Distribution**

☐ Launch Immediately after Installation

Automatic bundle installation can be achieved by selecting the *Install Immediately after Distribution* check box as shown in the figure above. The installation starts immediately after the particular bundles are distributed to the device.

12.3 YUM Repositories Derived from a Frozen Patch Level

Frozen patch levels created as described in [Section 12.1.2, “Creating a Frozen Patch Level,” on page 156](#) require a ZENworks Configuration Management agent on the target device. If a ZENworks Configuration Management agent cannot be installed or if the frozen patch level is to be deployed as part of the initial installation, an update source of type ZENworks cannot be used because of the non-standard APIs used by ZENworks Configuration Management.

However, ZENworks Configuration Management has the ability to convert patch bundles into an open format understandable by the standard zypp libraries. This is called a YUM repository.

A YUM repository can be used by zypper, which is the main patch management tool provided by SLES 11. It allows you to update devices without a full agent. In addition, YUM repositories can be consumed during the initial installation phase of a device because the libraries that are used (libzypp) do understand the repository format.

Novell Consulting recommends that you use YUM repositories to deploy frozen patch levels during unattended installations (AutoYaST) to reduce administrative overhead. A device installed in this way does not need to be patched until the next patching cycle. For more information, see [“Add-On Products” on page 52](#).


A YUM repository should always be created from a bundle group such as `SLES11-SP1-Updates-PROD`, containing a generic frozen bundle that represents the actual patch level used in production. Date strings or other dynamic identifiers must be avoided as part of the name for YUM repositories because the URL accessed by zypper or within AutoYaST must always remain the same. The alternative is to synchronize scripts or control files every time the name of the repository changes, which is not recommended.

[Figure 12-37](#) through [Figure 12-40](#) demonstrate how to create a YUM repository from a frozen patch bundle.

The creation process must be initiated from the summary page of a Linux bundle group.

Figure 12-37 Creating a YUM Service for a Bundle Group

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP2](#) > [SLES11-SP2-Updates_org](#) > **SLES11-SP2-Updates-PROD**

 **SLES11-SP2-Updates-PROD**

Summary

General

Object type: Bundle Group
GUID: 43ea999f9f9e2a139d390d3e17955096
Description: [\(Edit\)](#) Group for assignment of frozen SLES11-SP2 patch levels

Members

[Add](#) [Remove](#) [Move Up](#) [Move Down](#)

<input type="checkbox"/> Name	In Folder
<input type="checkbox"/> SLES11-SP2-Updates-20130206	/Bundles/Linux/SLES11/SP2/SLES11-SP2-Updates_org

1 - 1 of 1

YUM Service Details


YUM Service: [\(Create\)](#) (No YUM Service)

In the following dialog window, leave the *Auto update the repository when bundle is published* check box selected. In addition, at least one Primary Server that hosts the YUM service must be selected before you click *Finish*.

Figure 12-38 Creating a YUM Service for a Bundle Group - Select Auto Update Option and Primary Servers

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP1](#) > [SLES11-SP1-Updates](#) > [SLES11-SP1-Updates-DEV](#) > **Bundle YUM Service**

Bundle YUM Service

 **Step 1: Create YUM Service**

Service Name:

☒ Auto update the repository when bundle is published

Primary Servers

[Add](#) [Remove](#)

<input type="checkbox"/> Name	In Folder
<input checked="" type="checkbox"/> zcm11-node1	/Devices/Servers

1 - 1 of 1

show 5 items

<< Back

Finish

Cancel

An icon like a bouncing ball is displayed for a time, depending on the bundle size

Figure 12-39 Creating a YUM Service for a Bundle Group - Summary

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP1](#) > [SLES11-SP1-Updates](#) > **SLES11-SP1-Updates-DEV**

SLES11-SP1-Updates-DEV

Summary

General

Object type: Bundle Group
GUID: 1adf48009ab9bc782ac1ddd0f9db62d6
Description: [\(Edit\)](#)


Members

[Add](#) [Remove](#) [Move Up](#) [Move Down](#)

<input type="checkbox"/>	Name	In Folder
<input type="checkbox"/>	SLES11-SP1-Updates-20130206	/Bundles/Linux/SLES11/SP1/SLES11-SP1-Updates

1 - 1 of 1

YUM Service Details

YUM Service: [\(Edit\)](#) [\(Remove\)](#)  </zenworks-yumrepo/SLES11-SP1-Updates-DEV>

The YUM repository has now been created and the URL through which the new repository can be accessed is displayed.

Figure 12-40 Creating a YUM Service for a Bundle Group - Status And URL

[Bundles](#) > [Linux](#) > [SLES11](#) > [SP1](#) > [SLES11-SP1-Updates](#) > **SLES11-SP1-Updates-DEV**

SLES11-SP1-Updates-DEV

Summary

General

Object type: Bundle Group
GUID: 1adf48009ab9bc782ac1ddd0f9db62d6
Description: [\(Edit\)](#)


Members

[Add](#) [Remove](#) [Move Up](#) [Move Down](#)

<input type="checkbox"/>	Name	In Folder
<input type="checkbox"/>	SLES11-SP1-Updates-20130206	/Bundles/Linux/SLES11/SP1/SLES11-SP1-Updates

1 - 1 of 1

YUM Service Details

YUM Service: [\(Edit\)](#) [\(Remove\)](#)  </zenworks-yumrepo/SLES11-SP1-Updates-DEV>

The YUM repository created in the example above can be accessed by zypper, using the following command:

```
zypper ar https://<servername>/zenworks-yumrepo/SLES11-SP1-UPDATE-PROD sles11sp1_prod
```

To use the YUM repository within AutoYaST, an `add_on_products` section must be specified, similar to the following control file:

```
<add-on>
  <add_on_products config:type="list">
    <listentry>
      <media_url>%%YUM_SERVER%%/zenworks-yumrepo/SLES11-SP1-UPDATE-PROD</media_url>
      <product>SLES11-SP1-UPDATES</product>
      <product_name>sles11sp1_prod</product_name>
      <product_dir>/</product_dir>
    </listentry>
  </add_on_products>
</add-on>
```

IMPORTANT: YUM repositories created by ZENworks Configuration Management are not signed by the current ZENworks Configuration Management versions (ZCM 11 SP2). In practice, this means that every access to the YUM repository creates a warning issued by the client (for example, zypper or YaST) about an unsigned repository.

To avoid these messages, the ZCM repository can be manually signed by using `gpg`, as explained in the Cool Solutions article “[How to digitally sign a YUM repository created with ZCM11](http://www.novell.com/communities/node/13593/how-digitally-sign-yum-repository-created-zcm11)” (<http://www.novell.com/communities/node/13593/how-digitally-sign-yum-repository-created-zcm11>).

If a new patch cycle occurs, the frozen patch bundle representing the new patch level must be added to the bundle group assigned to the target devices. At the same time, the bundle representing the old frozen patch level must be removed from this same bundle group. This changes the status of the YUM service from green to yellow, indicating that the YUM service is not updated with the latest content on one or more servers. The update happens automatically at the time configured in your YUM Service Settings (see [Section 10.4, “Configuring the Inventory Schedule,” on page 83](#)).

If you want to start the update immediately, just select the *Edit* link and the *Finish* button. This triggers the update process indicated by the bouncing ball (see [Figure 12-39 on page 179](#)).

13 Agent Deployment

The ZENworks Adaptive Agent (further referred to as ZCM agent) can be deployed in various ways.

- ♦ [Section 13.1, “Manual Deployment,” on page 181](#)
- ♦ [Section 13.2, “Deployment Task,” on page 183](#)
- ♦ [Section 13.3, “Automated Deployment,” on page 190](#)

13.1 Manual Deployment

For manual deployments, a ZCM agent binary is needed, which can be downloaded from the ZCM server. Depending on the architecture and the Java environment installed on the target system, four different ZCM agent versions are available. In addition, a ZCM agent can be self-created, based on one of the four existing ZCM agents, so you can manually configure registration keys and zone settings.

The agent can be downloaded from ZENworks Configuration Management by using the *Download ZENworks Tools* link in the left frame. This link appears only if the *Configuration* entry has already been selected.

Figure 13-1 Agent Download



The next figure displays the four different agent versions. To reduce the amount of traffic, the Network Agent is preferred. This version of the agent requires a fully functional JRE already installed on the target system. You need to test whether the existing JRE at the devices is agent compatible before deploying this version of the agent.

Figure 13-2 Agent Download Page at the ZCM Server

Welcome to the ZENworks download page

Here you can download specific ZENworks components and tools. To view the listing of available tools, click the desired menu item to the left.

The ZENworks Adaptive Agent can be downloaded from the list below by clicking on the link entitled "Default". Please note that the "Network" install type requires Microsoft .NET be installed prior to running the installation.

Package Name	Target Platform	Target Architecture	Install Type	Size
Default Agent (x86_64 Complete)	Microsoft Windows	x86_64 Architecture (64 bit)	Standalone (.NET required)	71.8 MB
Default Agent (x86_64 Complete)	Microsoft Windows	x86_64 Architecture (64 bit)	Standalone	303.3 MB
Default Agent (x86_64 Network)	Microsoft Windows	x86_64 Architecture (64 bit)	Network (.NET required)	1.1 MB
Default Agent (x86 Complete)	Microsoft Windows	x86 Architecture (32 bit)	Standalone (.NET required)	70.2 MB
Default Agent (x86 Complete)	Microsoft Windows	x86 Architecture (32 bit)	Standalone	301.7 MB
Default Agent (x86 Network)	Microsoft Windows	x86 Architecture (32 bit)	Network (.NET required)	1.1 MB
Default Agent (x86_64 Network)	Linux	x86_64 Architecture (64 bit)	Network (JRE required)	1.5 MB
Default Agent (x86_64 Complete)	Linux	x86_64 Architecture (64 bit)	Standalone	123.8 MB
Default Agent (x86 Network)	Linux	x86 Architecture (32 bit)	Network (JRE required)	1.5 MB
Default Agent (x86 Complete)	Linux	x86 Architecture (32 bit)	Standalone	121.8 MB

1 - 10 of 10 show 25 items

The downloaded binary must be flagged executable and must be called from the command line:

```
./PreAgentPkg_AgentLinuxComplete.bin -G -k <location_key>
```

Several installation options are available. For instance, you can use the -k option to specify a registration key during installation. In addition, Novell Consulting recommends that you use the -G option to prevent the installation of packages that require X.

Figure 13-3 Command Line Options for ZCM Agent Installation

```
Usage: PreAgentPkg_AgentLinuxComplete.bin <options>

BUILDTIME <options> are:
  -f <file>      = add <file> to extractor
  -o <outf>      = create extractor named <outf>
                  (default is wrapped)

RUNTIME <options> are:
  -l             = list contents only (do not extract)
  -n             = no command execution (extract only)
  -S             = disable SELinux for RHEL devices in case installation fails
  -G             = do not install packages which require X or GUI
  -k <registration key> = register with the specified registration key after install
  -c <file>      = custom standalone install file
  -d <dest>      = extract files to <dest>
                  (default is /opt/novell/zenworks/stage/)

OTHER <options> are:
  -v             = be verbose
  -h             = help (show this message)
```

13.2 Deployment Task

The deployment task is a method where the ZCM agent software is distributed by the ZCM server. The task is initiated from the ZENworks Configuration Management Web interface. This method can be used for installing the agent at many devices simultaneously. The underlying operating system must be available for this kind of installation, unlike an automatic installation procedure, such as via AutoYaST.

The deployment task can be combined with several actions. The interface allows you to specify registration keys and whether to execute pre-installation and post-installation tasks.

The *Deployment Task* menu is accessible from the left main menu frame within ZENworks Configuration Management.

Figure 13-4 Deployment Task

The screenshot displays the ZENworks Configuration Management web interface. On the left is a navigation menu with the following items: Home, Devices, Users, Policies, Bundles, Patch Management, **Deployment** (highlighted with a red box), Reports, and Configuration. Below these are 'Deployment Activities' with links: Deploy Managed Device, Edit Deployment Package, Schedule Discovery Task, Discover Advertised Devices, and Import Deployable Devices. At the bottom is a 'Frequently Used' section with a link: SLES11SP1-POOL.

The main content area contains three panels:

- Discovery Tasks**: A table with columns Name, Schedule, Status, Devices Found, and Last Scan. It shows 'No items available.'
- Deployment Tasks**: A table with columns Name, Schedule, and Status. It contains one entry: 't1' with a schedule of 'Not Scheduled' and a status of 'Registering: 1 (Total: 1)'. The 'New' button in the header is highlighted with a red box. The footer shows '1 - 1 of 1' and 'show 5 items'.
- Deployable Devices**: A table with columns Name, IP Address, Operating System, Initial Discovery, and Deployment Status. It shows 'No items available.'

Click *New* to initiate a new deployment task. The next form asks for a task name and a description.

Figure 13-5 Creating a Deployment Task - Enter Task Name

The screenshot shows a dialog box titled "Deploy Device Wizard" with a sub-header "Step 1: Enter Deployment Task Name". It contains a "Name:" field with an asterisk, containing the text "ZCM_LX_AGENT_64BIT-DUS". Below it is a "Description:" text area containing the text "Deploys a standard linux agent and registers devices in Devices/Servers /SERVERS/PROD/DUS". At the bottom, there is a note: "* Fields marked with an asterisk are required." and three buttons: "<< Back", "Next >>", and "Cancel".

The following dialog window asks the user whether the IP address or DNS name of the target devices is being used:

Figure 13-6 Creating a Deployment Task - Select Devices (1)

The screenshot shows a dialog box titled "Deploy Device Wizard" with a sub-header "Step 2: Select Devices". The task name "ZCM_LX_AGENT_64BIT-DUS" is displayed in the header. The main text says "Select the devices where you would like the ZENworks Management Agent software deployed." Below this is a large empty list box with a scrollbar. To the right of the list box are three buttons: "Add...", "Remove", and "Clear". At the bottom, there is a section titled "Select whether you want to deploy to the target devices by using IP Address or DNS Name" with two radio buttons: "DNS Name" and "IP Address", where "IP Address" is selected. At the very bottom are three buttons: "<< Back", "Next >>", and "Cancel".

Click *Add* to display multiple options to specify devices to which the agent should be deployed. For simplicity, the example in the following figures demonstrates a deployment by IP address.

Figure 13-7 *Creating a Deployment Task - Select Devices (2)*

The 'Source' dropdown menu is open, showing the following options:

- IP Address (highlighted)
- Discovered Devices
- IP Address
- Add new CSV
- Add new LDAP source

Note: All IP address(es) entered are validated upon selecting "OK" button. Larger ranges will take longer to validate.

The 'Selected Devices' table is empty.

Remove	Name	Source:
--------	------	---------

0 Selected Items [Remove All](#)

You can specify either single IP address or an IP address range.

Figure 13-8 *Creating a Deployment Task - Select Devices (3)*

The 'Source' dropdown menu is set to 'IP Address'.

The 'IP Address Range/Host Name' field contains '10.0.0.255' and the 'Add' button is visible.

Note: All IP address(es) entered are validated upon selecting "OK" button. Larger ranges will take longer to validate.

The 'Selected Devices' table now contains one item:

Remove	Name	Source:
<input checked="" type="checkbox"/>	10.0.0.255	/IP Address

1 Selected Item [Remove All](#)

OK Cancel

A list is displayed after one or more IP addresses have been added.

Figure 13-9 Creating a Deployment Task - Select Devices (4)

[Deployment Tasks](#) > **Deploy Device Wizard**

Deploy Device Wizard ZCM_LX_AGENT_64BIT-DUS

Step 2: Select Devices

Select the devices where you would like the ZENworks Management Agent software deployed.

10.0.4.200 (10.0.4.200)

Add...

Remove

Clear

Select whether you want to deploy to the target devices by using IP Address or DNS Name

☐ DNS Name

☒ IP Address

[<< Back](#) [Next >>](#) [Cancel](#)

The deployment of the ZCM agent is executed via port 22 (SSH) on Linux devices. An admin user name with sufficient rights to install the agent (usually root) and the corresponding password are required by ZENworks Configuration Management for login and installation at the target devices:

Figure 13-10 Creating a Deployment Task - Enter Credentials

[Deployment Tasks](#) > **Deploy Device Wizard**

Deploy Device Wizard ZCM_LX_AGENT_64BIT-DUS

Step 3: Enter Credentials

Enter the credentials to be used v

☐ Save credentials to datastore

Credentials

[Add](#) [Edit](#) [Delete](#)

☐ Username:

No items available.

Enter Credential Information ? x

Type:
Linux v

Username:
root

Password:
.....

Reenter Password:
.....

OK

Cancel

The deployment task can be scheduled like most other tasks in ZENworks Configuration Management:

Figure 13-11 Creating a Deployment Task - Select Schedule

The screenshot shows the 'Deploy Device Wizard' window for 'ZCM_LX_AGENT_64BIT-DUS'. The current step is 'Step 4: Select Schedule'. The instruction reads: 'Create a schedule specifying when you want the deployment to occur.' There are two radio buttons: 'Now' (selected) and 'Scheduled'. Below them is a 'Schedule Type:' dropdown menu currently set to 'No Schedule'. At the bottom are three buttons: '<< Back', 'Next >>', and 'Cancel'.

You need to select the server to perform the task. This is important in environments with multiple primary servers.

Figure 13-12 Creating a Deployment Task - Select Primary Server

The screenshot shows the 'Deploy Device Wizard' window for 'ZCM_LX_AGENT_64BIT-DUS'. The current step is 'Step 5: Select Primary Server'. The instruction reads: 'Select the server that will perform the deployment task.' There is a text field labeled 'Primary Server: *' containing the path '/Devices/Servers/zcm-docu-node01'. To the right of the text field is a magnifying glass icon. Below the text field is a note: '* Fields marked with an asterisk are required.' At the bottom are three buttons: '<< Back', 'Next >>', and 'Cancel'.

The proxy asked for in the next dialog window is not meant as a Web proxy but rather another Primary or Satellite server inside the ZENworks Configuration Management zone. The box can be left empty in single-server environments or environments where Primary Server and target devices are located within the same network.

Figure 13-13 Creating a Deployment Task - Select or Edit Proxy Device

The screenshot shows the 'Deploy Device Wizard' window for 'ZCM_LX_AGENT_64BIT-DUS'. The current step is 'Step 6: Select or Edit a Proxy Device'. The instruction reads: 'Select the proxy that should perform the Task.' Below the instruction is a table titled 'Select or Edit a Proxy Device'.

Type	Selected Device
Windows Proxy	None
Linux Proxy	None

At the bottom are four buttons: '<< Back', 'Next >>' (highlighted with a red border), 'Finish', and 'Cancel'.

If you click *Finish* at this stage, some of the deployment properties are set to their defaults. To select a particular agent deployment package, prevent the installation of packages running on X (GUI mode) only, and select a registration key, click *Next* instead.

The Default Standalone agent is used in the current example:

Figure 13-14 Creating a Deployment Task - Install Without GUI Packages

Deployment Package

Depending upon the processor architecture of the managed linux device, choose the deployment package to be used for installing ZENworks Adaptive Agent on the device. If you are not sure about the device's processor architecture, choose the package with target architecture as All, which applies to 32-bit and 64-bit platforms. If the selected package has been deleted from the Primary Server, then the default deployment package is deployed.

Default Agent (Target Architecture : All, Install Type: Network (JRE required))

Default Agent (Target Architecture : All, Install Type: Network (JRE required))

Default Agent (Target Architecture : All, Install Type: Standalone)

Default Agent (Target Architecture : x86 Architecture (32 bit), Install Type: Network (JRE required))

Default Agent (Target Architecture : x86 Architecture (32 bit), Install Type: Standalone)

Default Agent (Target Architecture : x86_64 Architecture (64 bit), Install Type: Network (JRE required))

Default Agent (Target Architecture : x86_64 Architecture (64 bit), Install Type: Standalone)

Select this option if you do not want to install the GUI packages of ZENworks Adaptive Agent.

☒ **Do Not Install the GUI Packages**

Select this option to disable SELinux. SELinux is temporarily disabled if the agent is unable to open the ports required by ZENworks.

☐ Disable SELinux for Red Hat Devices

Note : The ZENworks Adaptive Agent is installed by default to /opt/novell/zenworks/ on the target machine.

<< Back **Next >>** Finish Cancel

Only one registration key can be specified in this dialog. If you want to add additional device group registration keys, you can do this in the summary screen where pre-installation and post-installation tasks are added.

Figure 13-15 Creating a Deployment Task - Select Registration Key

Select Registration Key

Look in: /Keys/PROD/LOCATION

Name filter: * Items of type: All Types

Name	Type
PROD_PROVO_EDIR	Registration

1 - 1 of 1 show 25 items

OK Cancel

In this example, the device is registered with the additional ZENworks Configuration Management PROD_OES11GA registration key:

Figure 13-16 Creating a Deployment Task - Define Pre-Deployment and Post-Deployment Tasks

[Deployment Tasks](#) > **Deploy Device Wizard**

Deploy Device Wizard **ZCM_LX_AGENT_64BIT-DUS**

Step 9: Pre/Post Deployment

Instructions for command pack.

Linux Pre-deployment Commands ☐ Continue on error

Add Edit Remove

Command	Arguments
No items available.	

Linux Post-deployment Commands ☒ Continue on error

Add Edit Remove

Command	Arguments
/opt/novell/zenworks/bin/zac ark PROD_OES11GA	

<< Back Finish Cancel

The deployment task starts at the scheduled time:

Figure 13-17 Creating a Deployment Task - Success

Success: The Deployment Task was successfully created.

Discovery Tasks

New Edit Delete Action

Name	Schedule	Status	Devices Found Last Scan
No items available.			

Deployment Tasks

New Edit Delete Action

Name	Schedule	Status
t1	Not Scheduled	Inactive
ZCM_LX_AGENT_64BIT-DUS	Run on: 3/29	Scheduled

1 - 2 of 2 show 5 items

Deployable Devices **Advanced**

Delete Action View

Name	IP Address	Operating System	Initial Discovery	Deployment Status
10.0.4.200	10.0.4.200	Unknown OS	Mar 19, 2013 8:44 PM	Scheduled
demo01ds.demo01.com	10.0.10.11	SuSE Linux Enterprise Server	Jan 25, 2013 5:47 PM	Inactive

1 - 2 of 2 show 25 items

13.3 Automated Deployment

For automated deployments, any method that can launch the ZCM agent deployment package is suitable. This means that the steps in [Section 13.1, “Manual Deployment,” on page 181](#) must be executed via a scripting method.

Novell Consulting recommends that you install the ZCM agent via AutoYaST within a post-installation script and that you use the same procedure for registering the device at its target location and server groups. For more information, see [“The zcm-install.sh Script” on page 54](#).

14 Agent Commands

Some agent commands have been introduced in preceding sections. Detailed instructions about applicable agent commands and their usage can be found in the man page or by using `zac --help` to call the agent help.

A summary of the most important agent commands is given below. Use `zac <command> --help` for specific help on any subcommand.

- ♦ [Section 14.1, “Core Commands,” on page 191](#)
- ♦ [Section 14.2, “Linux Package Commands,” on page 191](#)
- ♦ [Section 14.3, “Bundle Commands,” on page 192](#)

14.1 Core Commands

Command	Description
<code>zac zc (zone-config)</code>	Shows zone properties
<code>zac get (get-preferences)</code>	Lists agent settings such as proxy, rollback, and log settings
<code>zac set</code>	Changes settings listed with <code>zac get</code>
<code>zac ref (refresh)</code>	Refreshes all agent information
<code>zac agp (agent properties)</code>	Displays a summary about the agent state, operating system details, and component versions
<code>zac reg (register)</code>	Registers an agent at the ZENworks Configuration Management zone
<code>zac ark (add-registration-key)</code>	Adds a key for group registration
<code>zac logger</code>	Changes agent log configurations
<code>zac cc (clean cache)</code>	Clears all cached agent information

14.2 Linux Package Commands

Command	Description
<code>zac lu (list updates)</code>	Lists all updates
<code>zac up (update)</code>	Updates packages where higher versions are applicable in any assigned channel
<code>zac se (search)</code>	Searches for a given package by package name or expression
<code>zac in (install)</code>	Installs a package and its dependencies

Command	Description
<code>zac rm (remove)</code>	Removes a package and its dependencies

14.3 Bundle Commands

Command	Description
<code>zac bl (bundle list)</code>	Lists all assigned bundles
<code>zac bin (bundle install)</code>	Installs a given bundle
<code>zac bu (bundle uninstall)</code>	Uninstalls a given bundle (use with caution)

15 Fault Diagnostics and Debugging

ZENworks Configuration Management consists of many different components. Most of them are well explained in the [Novell Knowledgebase Document 3418069](http://www.novell.com/support/kb/doc.php?id=3418069). The configuration and log location for some important components are explained in this section.

- ♦ [Section 15.1, “ZCM Server,” on page 193](#)
- ♦ [Section 15.2, “ZCM Agent,” on page 194](#)

15.1 ZCM Server

- ♦ [Section 15.1.1, “ZENworks Control Center,” on page 193](#)
- ♦ [Section 15.1.2, “ZENworks Web Service,” on page 193](#)
- ♦ [Section 15.1.3, “CASA,” on page 193](#)
- ♦ [Section 15.1.4, “Deployment,” on page 193](#)
- ♦ [Section 15.1.5, “Discovery,” on page 194](#)
- ♦ [Section 15.1.6, “zman,” on page 194](#)

15.1.1 ZENworks Control Center

Configuration: `/opt/novell/zenworks/share/tomcat/webapps/zenworks/web-inf/config.xml`

Setting ID: `debug.tags`

This setting might need to be modified for more logging as requested by Novell Technical Support.

Log Location: `/var/opt/novell/log/zenworks/zcc.log`

15.1.2 ZENworks Web Service

Log Location: `/var/opt/novell/log/zenworks/services-messages.log`

15.1.3 CASA

Configuration: `/etc/CASA/authtoken/svc/log4j.properties`

`set log4j.rootLogger=` (change warn to debug and restart the service)

Log Location: `/srv/www/casaats/logs/*`

15.1.4 Deployment

Configuration: `/etc/opt/novell/zenworks/loader/deployment.xml`

Log Location: `/var/opt/Novell/log/zenworks/loader-messages.log`

15.1.5 Discovery

Configuration: /etc/opt/novell/zenworks/loader/discovery.xml

Log Location: /var/opt/novell/log/zenworks/loader-messages.log

15.1.6 zman

Configuration: /etc/opt/novell/zenworks/zman/properties/zman-config.properties

Set `DEBUG_LEVEL=4`

Log Location: /var/opt/novell/log/zenworks/zman.log

15.2 ZCM Agent

- [Section 15.2.1, “Changing the Log Level,” on page 194](#)
- [Section 15.2.2, “Log Files,” on page 194](#)

15.2.1 Changing the Log Level

```
zac log level debug
zac set CaptureAxis2Logs true
```

15.2.2 Log Files

File	Description
/var/opt/novell/zenworks/novell-zenworks-xplat-zmd.out	Contains agent start and stop messages.
/opt/novell/zenworks/logs/LocalStore/zmd-messages.log	Most relevant agent information is found here.
/var/log/zmd-satbackend.log	This log deals only with actions taken by the agent to install an RPM. It also shows the communication that happens (essentially API calls) between the agent and SATLIB. SATLIB is an interface to the RPM that actually does the install.
/var/opt/novell/log/zenworks/preboot/novell-zislnx.log	Contains information about the managed ZIS partition of a device. The information is written or read from the ZIS sector.