

Novell ZENworks OnDemand Services

2

www.novell.com

INSTALLATION AND ADMINISTRATION

103-000239-001



Novell®

Legal Notices

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

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

You may not export or re-export this product in violation of any applicable laws or regulations including, without limitation, U.S. export regulations or the laws of the country in which you reside.

Copyright © 2000 -2002 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

U.S. Patent Pending.

Novell, Inc.
1800 South Novell Place
Provo, UT 84606
U.S.A.

www.novell.com

Installation and Administration
[December 20, 2002](#)

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

Novell Trademarks

ApprovalFlow is a trademark of Novell, Inc.

ConsoleOne is a registered trademark of Novell, Inc. in the United States and other countries.

DeFrame is a trademark of Novell, Inc.

DirXML is a registered trademark of Novell, Inc. in the United States and other countries.

eDirectory is a trademark of Novell, Inc.

GroupWise is a registered trademark of Novell, Inc. in the United States and other countries.

iChain is a registered trademark of Novell, Inc. in the United States and other countries.

NDPS is a registered trademark of Novell, Inc. in the United States and other countries.

NDS is a registered trademark of Novell, Inc. in the United States and other countries.

NetWare is a registered trademark of Novell, Inc. in the United States and other countries.

NLM is a trademark of Novell, Inc.

Novell is a registered trademark of Novell, Inc. in the United States and other countries.

Novell Application Launcher is a trademark of Novell, Inc.

Novell Client is a trademark of Novell, Inc.

Novell Consulting is a service mark of Novell, Inc.

Novell Distributed Print Services is a trademark of Novell, Inc.

Novell iFolder is a trademark of Novell, Inc.

SnAppShot is a trademark of Novell, Inc.

ZENworks is a registered trademark of Novell, Inc. in the United States and other countries.

ZENworks OnDemand Services is a trademark of Novell, Inc.

Third-Party Trademarks

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

Contents

About This Guide	9
Part I Introduction	
1 What Is ZENworks OnDemand Services?	13
Part II Installation	
2 Preparing for Installation	19
Verifying Software Requirements	19
Web Server, Web Application Server, JVM, and Novell Portal Services	20
NetWare 5.1/6.0 or Windows NT/2000 Server	20
Novell eDirectory and ConsoleOne.	21
DeFrame	22
ZENworks for Desktops	22
Desktop Operating Systems and Web Browsers.	23
Deciding on a Configuration	23
Sample Configuration 1 - Single Server Environment	24
Sample Configuration 2 - Mixed Server Environment	25
Sample Configuration 3 - Mixed Server Environment	26
Determining Installation Order	26
3 Installing OnDemand Services	29
Installing the Software	29
Installing the Software	30
Installing the Software (excluding NPS 1.5 Runtime)	32
Enabling Purchased Desktop or Thin-Client Applications to be Associated with Users	38
Applying Support Pack 1	40
Increasing the Tomcat Servlet Engine's Heap Size	40
Enabling Extended Character Support	41
Restarting the Web Server, Web Application Server, and JVM	42
Configuring the Cost Center	43
Adding Budget Holders	43
Assigning Users	45
Configuring Your Portal	47
Creating Portal Pages	47
Running the OnDemand Services Configuration Wizard.	62
Restarting the Web Server, Web Application Server, and JVM	63

Starting the Maintenance and UsageServer Processes.	64
Starting the Maintenance Process.	64
Starting the UsageServer Process.	64
What's Next?	65

Part III Administration

4 Administration Basics	69
OnDemand Services Gadgets.	69
OnDemand Services Snap-Ins for ConsoleOne.	69
Novell eDirectory Management	70
ODSAdmin Trustee Rights	71
5 Creating Application Packages	73
Preparing the Package's Applications.	74
Desktop Applications	74
Thin-Client Applications	75
Web Applications	75
Creating the Package Object	78
Adding Applications to the Package.	80
Establishing the Package's Cost	81
Enabling the Launch Item Gadget to Display Desktop Applications.	83
6 Distributing Free Applications	85
Configuring an Application for Free Distribution through User Associations	85
Assigning the OnDemand Administrator (ODSAdmin) Rights to the Application Object	86
Integrating with Novell Application Launcher	87
7 Tracking Application Usage	89
Usage Tracking Support for Each Application Type.	89
Enabling Usage Tracking of an Application	90
Ensuring Accurate Usage Tracking for Thin-Client Applications	91
Starting the UsageServer and Maintenance Processes.	92
Starting the UsageServer Process.	92
Starting the Maintenance Process	93
8 Limiting Package Availability	95
Configuring the Package Request Gadget to Limit Package Availability	95
Giving Users Trustee Rights to Packages.	96
Making Users Trustees of a Package Object	96
Making Users Trustees of a Container Object.	97

9	Securing Web Applications With Novell iChain	99
	Setting Up Protected Resources	100
	Creating ACL Rules	101
	Configuring iChain Server ACL List Refresh	102
	Making the ODSAdmin Object a Trustee of ACL Rules	104
	Associating ACL Rules with Web Applications.	105
10	Organizing Packages and Applications	107
	Organizing Packages in the Package Request Gadget	107
	Organizing Applications in the Launch Item Gadget	108
11	Customizing Notification Messages	111
	Modifying the Message Subject	112
	Modifying the Message Body	112
12	Supporting Desktop Application Users	115
13	Supporting Thin-Client Application Users	117
	Installing the DeFrame, ICA, and RDP Web Clients	117
	Updating the ICA Web Client.	118
	Installing the DeFrame Client	119
	Issues for Users to Be Aware Of.	120
14	Creating Cost Centers	121
	Creating a Cost Center and Adding Budget Holders.	121
	Assigning Users to the Cost Center	123
15	Billing for Purchases	125
	Developing Your Own Interface	125
	Getting Development Help	126
16	Changing the Status of a Purchase	127
17	Managing the Maintenance and UsageServer Processes	129
	Understanding the Maintenance Process	129
	Understanding the UsageServer Process	130
	How the UsageServer and UsageClient Work	130
	What to Do If the UsageServer and the UsageClient Are in Different Subnets	131
	Multiple UsageServers and Load Balancing	134
	Explicitly Defining the UsageServer's Address.	134
	Adding the UsageServer's Address to the Launch Item Gadget	135
	Adding the UsageServer's Address to a DeFrame Server's Registry	136
	Installing Additional Maintenance and UsageServer Processes	137
	Modifying the Maintenance and UsageServer Configuration File (Commerceserver.cfg)	140
	Modifying Logging Settings	143

Part IV Appendices

A	Installation Worksheets	147
	OnDemand Services Installation Worksheet	147
	Novell Portal Services Installation Worksheet.	153
B	Gadget Configuration Settings	157
	Approval Gadget (OD_ApprovalGadget)	158
	Configuration Wizard Gadget (OD_ConfigWizardGadget)	158
	Current Budget Holder Gadget (OD_CurrentBHGadget)	159
	Edit Account Gadget (OD_EditAccountGadget).	159
	Launch Item Gadget (OD_LaunchItemGadget)	160
	Package Request Gadget (OD_PackageRequestGadget)	164
	Report Gadget (OD_ReportGadget)	165
	User Administration Gadget (OD_UserAdminGadget)	165
	Workflow Tracking Gadget (OD_WorkflowTrackingGadget)	169
C	Tabbed Portal Pages	171
D	Upgrading from OnDemand Services 1.5	175
	Differences Between OnDemand Services 1.5 and ZENworks OnDemand Services 2	175
	What Is Upgraded	177
	Web Server Environment	177
	eDirectory Information	177
	Maintenance and UsageServer Processes	178
	Preparing for the Upgrade.	179
	Performing the Upgrade.	179
	Removing Obsolete OnDemand Service Objects and Attributes	180
	Removing Unused Software Components	181
E	Documentation Updates	183
	October 7, 2002 (ODS2Patch102002.exe)	183
	December 20, 2002 (Support Pack 1).	186

About This Guide

This guide is designed to help you set up Novell® ZENworks® OnDemand Services™ 2 as quickly and easily as possible. The guide is intended for network administrators who will install and administer OnDemand Services and is divided into the following sections:

- ♦ “Introduction” on page 11
- ♦ “Installation” on page 17
- ♦ “Administration” on page 67

Documentation Updates

For the most recent version of the ZENworks OnDemand Services 2 *Installation and Administration Guide*, visit the [Novell ZENworks OnDemand Services 2 Web site \(http://www.novell.com/documentation/lg/nzods20\)](http://www.novell.com/documentation/lg/nzods20).

Documentation Conventions

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

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

User Comments

We want to hear your comments and suggestions about this manual and the other documentation included with ZENworks OnDemand Services. To contact us, send e-mail to proddoc@novell.com.



Introduction

Chapter 1, “What Is ZENworks OnDemand Services?” on page 13

1

What Is ZENworks OnDemand Services?

Novell® ZENworks® OnDemand Services™ 2 provides directory-based provisioning of applications to users.

If your organization is like most, application provisioning carries with it a significant amount of overhead for your users and IT staff. In many cases, paperwork or telephone calls are required to gain access to an application, directory and file system rights need to be granted to the user, and installation of the application might require one or more service trips to the user's work area. If your IT department needs to bill the user to recover or track the application's purchase and rollout costs, additional overhead is generated.

With OnDemand Services, you can lessen your provisioning overhead by defining available applications in Novell eDirectory™, automating the approval process used to gain access to applications, and automating the delivery of approved applications to users' desktops.

At the same time, you can satisfy your organization's budget management requirements by assigning purchase and usage fees to applications, tracking the application fees, and billing individual departments for the application fees accrued by their users.

The following list highlights some of the features that let you realize the benefits of using OnDemand Services as your application provisioning solution:

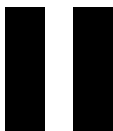
- ♦ **Directory-Based Management:** All application and user information is stored and managed in Novell eDirectory. Using eDirectory makes configuration and management of application provisioning quicker and

easier and enables OnDemand Services to automatically control user access to applications through use of directory rights.

- ♦ **Multi-Type Application Support:** OnDemand Services supports thin-client (terminal server) applications, Web-based applications, and traditional desktop applications. Provisioning of thin-client applications requires the use of DeFrame™ 2.01, included with OnDemand Services. Provisioning of traditional desktop applications requires the use of ZENworks for Desktops 3.2.
- ♦ **Application Packages:** Applications are made available to users through OnDemand Services packages. A package can contain a single application or a suite of applications. You can include desktop, thin-client, and Web applications in the same package.
- ♦ **Application Usage Tracking:** You can track application usage for individual users on a per-minute or per-use basis.
- ♦ **Purchase and Usage Charges:** You can provide application packages to user free of charge, or you can charge purchase and usage fees for the package. The charge can be based on a one-time purchase fee for limited or unlimited use, a subscription fee, a per-use fee, a per-minute fee, or a combination of these fees.
- ♦ **Administrator- or User-Initiated Provisioning:** Administrator-initiated provisioning enables you to push applications to users. User-initiated provisioning enables users to request application packages. Their requests must be approved before the applications can be used.
- ♦ **Automated Approval Workflow:** For user-initiated provisioning, OnDemand Services provides ApprovalFlow™, an automated workflow process for requesting application packages and approving requests. ApprovalFlow enables users to request packages and track their requests. It also enables administrators to approve or deny requests.
- ♦ **Purchase and Usage Charge Tracking:** Purchase and usage charges are tracked against accounting units referred to as cost centers. You can use a single cost center for tracking all users' charges, or, if you need to track application charges by organization, you can create cost centers for each organization.
- ♦ **Delegated Provisioning Authority:** You can delegate provisioning authority to individuals within each cost center. These individuals, called budget holders, have the ability to control the application charges accrued

to their cost center by approving or denying users' requests for applications. They also have the ability to assign applications to the cost center's users, create user accounts assigned to their cost center, and enable or disable the user accounts.

- ◆ **Web-Browser Interface:** Users and budget holders access OnDemand Services through a Web browser interface. From this one location, users can request applications, track their requests, launch applications, and generate reports showing the charges they've accrued for using applications. They can also edit their user information. Budget holders, in addition to having the same capabilities as users, can approve or deny requests for applications, create user accounts, enable and disable user accounts, and generate reports showing the charges their cost center's users have accrued for using applications. The OnDemand Services Web-browser interface is provided through Web portal information windows, called gadgets, that are installed and configured under Novell Portal Services.
- ◆ **ZENworks for Desktops 3.2 Integration:** OnDemand Services requires ZENworks for Desktops (ZfD) 3.2 in order to provision traditional desktop applications. ZfD 3.2 provides the delivery mechanism for these applications. Other integration points include:
 - ◆ Launching both desktop and thin-client applications from Novell Application Launcher™. Launching Web applications through Application Launcher is not supported at this time.
 - ◆ Launching both packaged applications and user-associated applications from the OnDemand Services Web-browser interface. Packaged applications are applications that are included in an OnDemand Services package. To gain access to a packaged application, users must purchase the application's package. User-associated applications are applications that are associated directly with users. Users immediately gain access to the applications without purchasing a package.
- ◆ **Novell iChain Integration:** Novell iChain® provides secure authentication and access to portals, Web-based content, and Web applications. OnDemand Services integrates with Novell iChain 2.x to enable you to secure Web applications you've included in OnDemand Services packages. Through the use of iChain ACL (Access Control List) rules, OnDemand Services enables access to Web applications when a user purchases the Web application's package and disables access when the purchase expires.



Installation

Chapter 2, “Preparing for Installation,” on page 19

Chapter 3, “Installing OnDemand Services,” on page 29

2

Preparing for Installation

This section contains information to help you install a new Novell® ZENworks® OnDemand Services™ system. If you are upgrading from OnDemand Services 1.5/1.5a, please see [Appendix D, “Upgrading from OnDemand Services 1.5,”](#) on page 175.

Before you begin installing OnDemand Services, you need to complete the preparation tasks in the following sections:

- ◆ [“Verifying Software Requirements”](#) on page 19
- ◆ [“Deciding on a Configuration”](#) on page 23
- ◆ [“Determining Installation Order”](#) on page 26

Verifying Software Requirements

The following sections list the requirements for OnDemand Services:

- ◆ [“Web Server, Web Application Server, JVM, and Novell Portal Services”](#) on page 20
- ◆ [“NetWare 5.1/6.0 or Windows NT/2000 Server”](#) on page 20
- ◆ [“Novell eDirectory and ConsoleOne”](#) on page 21
- ◆ [“DeFrame”](#) on page 22
- ◆ [“ZENworks for Desktops”](#) on page 22
- ◆ [“Desktop Operating Systems and Web Browsers”](#) on page 23

Web Server, Web Application Server, JVM, and Novell Portal Services

The OnDemand Services user interface is created through gadgets that plug in to Novell Portal Services (NPS) 1.5. The OnDemand Services gadgets must be installed to one of the server environments listed in the table below.

Operating System	Web Server	Web Application Server	JVM	Novell Portal Services
NetWare® 6.0	Apache 1.3.20	Tomcat 3.3a	Novell JVM* for NetWare 1.3.1	NPS 1.5
Windows 2000 with Support Pack 2 or later	IIS 5.0 or Apache 1.3.20	Tomcat 3.3a	Sun* 1.3.1_01 or IBM* 1.3	NPS 1.5

If you have already used the full version of NPS 1.5 to create an NPS 1.5 portal, you can install the OnDemand Services gadgets to that portal. Otherwise, OnDemand Services includes a runtime version of Novell Portal Services 1.5 that can be installed during installation. The runtime version includes limited functionality and only supports the OnDemand Services gadgets.

NetWare 5.1/6.0 or Windows NT/2000 Server

OnDemand Services includes two Java* processes, Maintenance and UsageServer. The Maintenance process cleans up purchase and usage records in Novell eDirectory™ and closes out usage tracking of applications when the UsageServer becomes unavailable. It needs to be running at all times. The UsageServer process enables application usage to be tracked. It needs to be running if you want to be able to charge for an application based on the number of times or minutes it is used.

The OnDemand Services Maintenance and UsageServer processes can be installed on either a NetWare 5.1/6.0 server or a Windows NT*/2000 server. Because these are Java processes, the server must also have JVM 1.2.2 (or later) installed.

You can install the UsageServer and Maintenance processes to the Web server or to a different network server. Your decision should be based on the Web server's workload and processing power.

Novell eDirectory and ConsoleOne

OnDemand Services requires eDirectory 8.5 or higher. We recommend that you have the latest release of the version you are using (8.5x, 8.6.x, etc.). The latest releases of these versions correct several issues with auxiliary classes (which are used by OnDemand Services) and provide better compatibility between eDirectory versions in a mixed eDirectory environment.

As you prepare your environment for OnDemand Services, please be aware of the following eDirectory requirements and issues:

- ◆ OnDemand Services accesses eDirectory through an LDAP server. The network server where your LDAP server resides needs to have eDirectory 8.5 or higher installed and must have Read/Write replicas of the partitions that hold the following objects:
 - ◆ The OnDemand Services objects.
 - ◆ The Application objects (ZENworks for Desktops or DeFrame) used with OnDemand Services.
 - ◆ The User objects for any OnDemand Services users.
- ◆ The NPS 1.5 server must have eDirectory 8.5 or higher installed. If the NPS server is a NetWare server, the server must have Read/Write replicas of the partitions that hold the Application objects. This enables the OnDemand Services ZENAPP32.NLM to make the proper rights assignments when a user purchases or is assigned an application.
- ◆ Because OnDemand Services uses auxiliary classes that are only supported in NDS[®] 8.x or higher, we recommend that OnDemand Services be installed in a tree that is exclusively NDS 8.x or higher. However, if you must have NDS 6/7 servers in the same tree as OnDemand Services, the NDS 6/7 servers should not hold replicas which contain OnDemand Services objects, Application objects used with OnDemand, or User objects for OnDemand users.

NOTE: NDS 6/7 servers are capable of synchronizing the schema associated with auxiliary classes and can therefore coexist in the same tree. However, objects which have auxiliary classes on them are converted to "Unknown" objects on NDS 6/7 servers. These objects will be displayed as "Unknown" objects and this will prevent you from administering these objects when your primary connection is to a NDS 6/7 server. In addition, OnDemand users whose objects appear as "Unknown" will be unable to authenticate to the NDS 6/7 servers

ConsoleOne

OnDemand Services requires ConsoleOne® 1.3.2 or higher. You can download the latest version of ConsoleOne from the [Novell Downloads site \(http://download.novell.com\)](http://download.novell.com).

eDirectory and LDAP Naming Restrictions

In all areas of the eDirectory tree that will be accessed by OnDemand Services, you should make sure that object names (including container and leaf objects) do not include any of the following characters. These characters, when used in LDAP names (for example, cn=jsmith,ou=users,o=novell), will result in invalid syntax.

" (double quote)	+ (plus)
= (equals)	\ (backslash)
< (open angle bracket)	; (semicolon)
> (close angle bracket)	/ (forwardslash)
, (comma)	# (pound sign)

DeFrame

To distribute thin-client (terminal server) applications, OnDemand Services requires DeFrame™ 2.01. DeFrame 2.01 is included in the OnDemand Services package.

ZENworks for Desktops

To distribute desktop applications, OnDemand Services requires ZENworks for Desktops (ZfD) 3.2.

Desktop Operating Systems and Web Browsers

The following table lists the desktop operating systems and Web browsers that OnDemand Services supports.

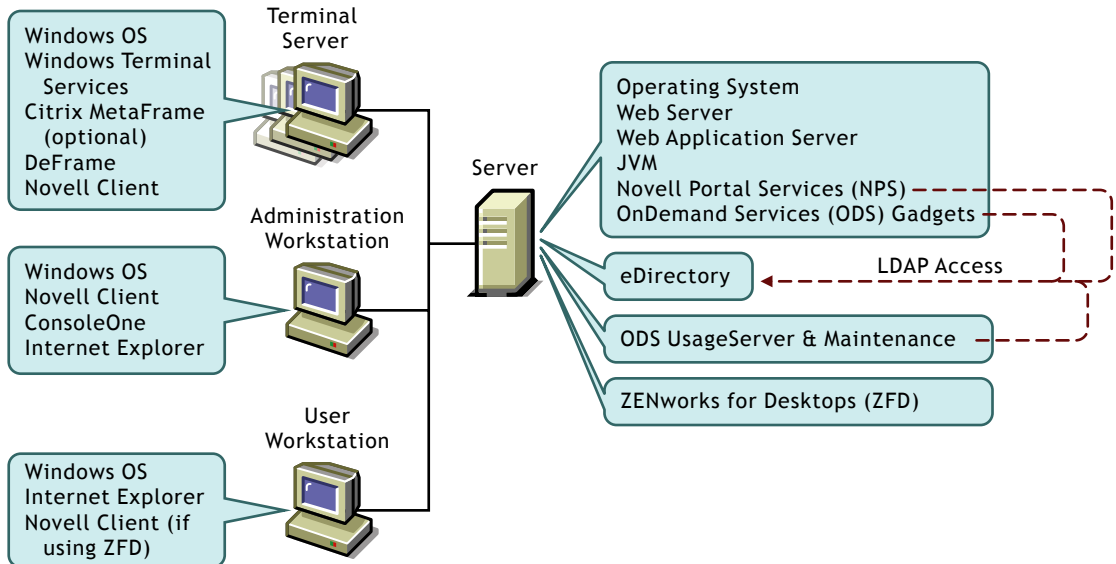
Operating System	Web Browsers
Windows 98 SE	Microsoft* Internet Explorer 5.5 with SP2
Windows NT 4.0 Workstation	Microsoft Internet Explorer 6.0
Windows 2000 Professional	
Windows XP	

Netscape* browsers are not supported by OnDemand Services in this release.

You should make sure that the latest support pack is applied to each of these software components.

Deciding on a Configuration

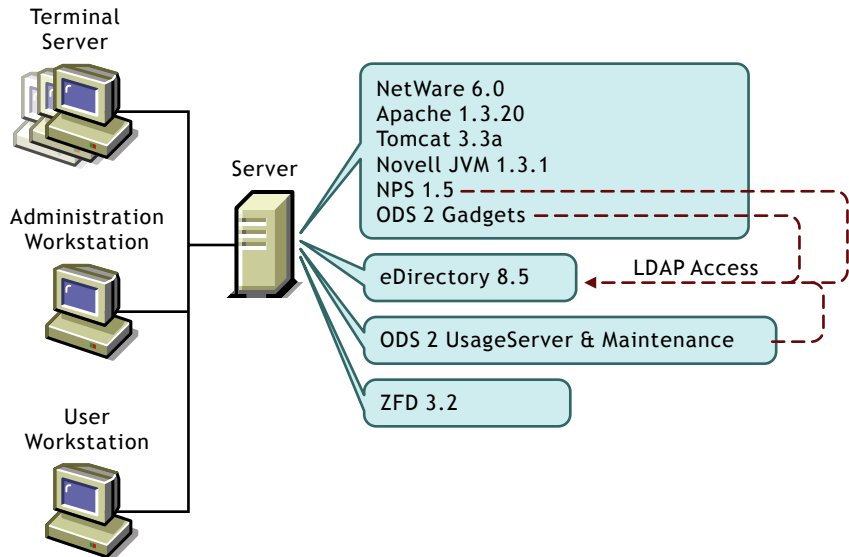
The OnDemand Services environment consists of many different software components, as explained in [“Verifying Software Requirements” on page 19](#) and illustrated in the following diagram.



All software components included in a shaded box must be installed on the same machine, but software components in different boxes (on the server) do not necessarily need to be installed on the same server. Below are some sample configurations you could use.

Sample Configuration 1 - Single Server Environment

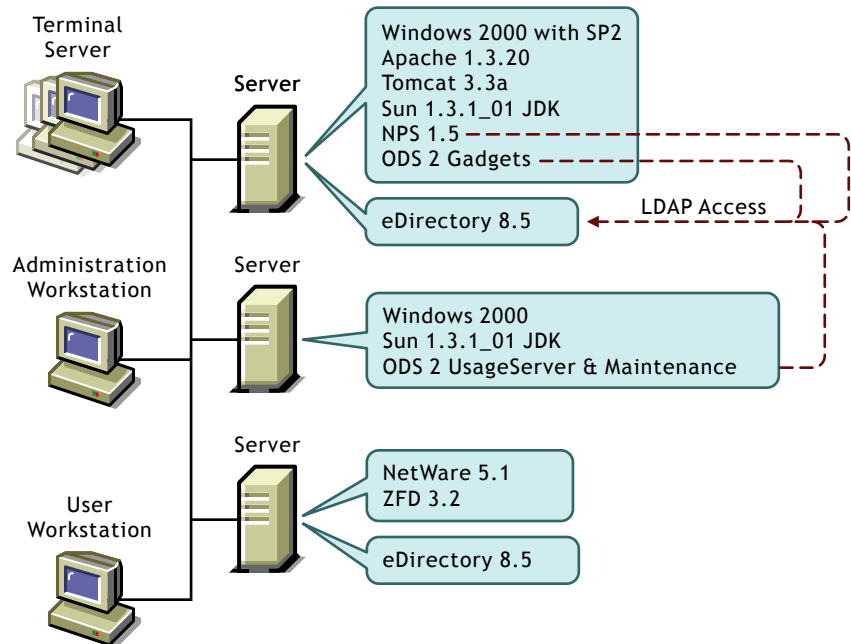
In this configuration, all software components are installed on a NetWare 6 server.



This configuration saves on hardware but might sacrifice performance. Users will be accessing NPS and the ODS gadgets through the Web server, while at the same time ZfD and the ODS UsageServer and Maintenance processes will be running. This configuration is recommended only if the server has a lot of processing power or is servicing a limited number of users.

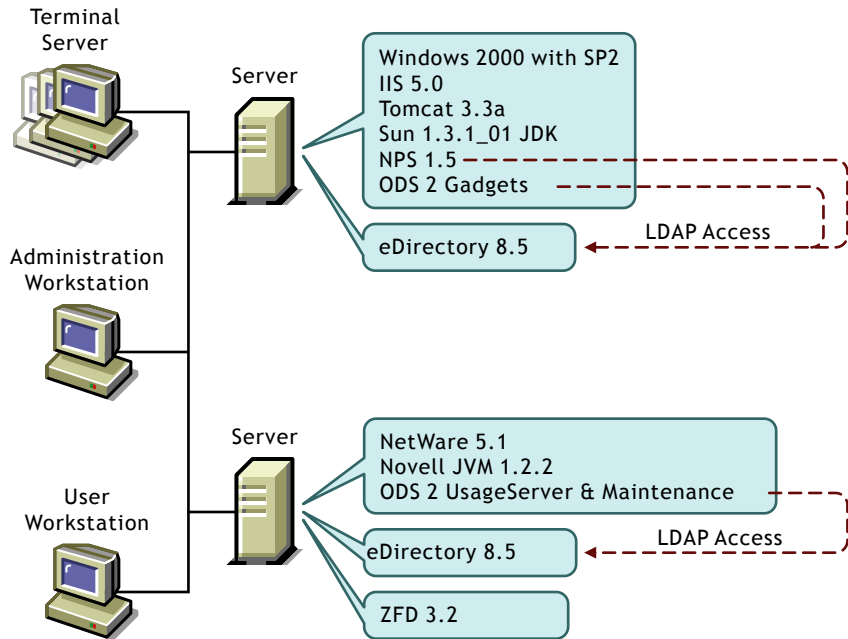
Sample Configuration 2 - Mixed Server Environment

In this configuration, ZfD and the ODS UsageServer and Maintenance processes have been installed to different servers. In addition, NPS, the ODS gadgets, and the ODS UsageServer and Maintenance processes are running on Windows 2000 servers that do not have eDirectory installed. They are accessing eDirectory, located on the NetWare 5.1 server, through LDAP.



Sample Configuration 3 - Mixed Server Environment

This sample configuration has Microsoft Internet Information Services being used as the Web server, ZfD and the ODS UsageServer and Maintenance processes running on the same NetWare 5.1 server, and eDirectory 8.5 installed on both the Windows 2000 server and the NetWare 5.1 server.



Determining Installation Order

If you do not already have some of the required software products installed, the preferred installation order is shown below. If there is an alternative installation order, the product's description gives it.

- ♦ **eDirectory 8.5 (or higher):** Make sure the tree and servers where you will install OnDemand Services components use eDirectory 8.5 or higher.

For information about installing eDirectory 8.5, see the documentation included with the product or see the Novell eDirectory documentation at the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

- ♦ **ZENworks for Desktops (ZfD) 3.2:** OnDemand Services enables usage tracking of desktop applications. If you will use ZfD 3.2, you should install it before OnDemand Services so that OnDemand Services can install its usage tracking components to the ZENworks server. If you choose to install ZfD 3.2 after OnDemand Services, you will need to run the OnDemand Services Installation program again and use the Custom Install option to install the OnDemand Services components for ZfD.

For information about installing ZfD 3.2, see the documentation included with the product or see the ZENworks for Desktops 3.2 documentation at the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

- ♦ **Web Server, Web Application Server, and JVM:** These software components must be installed before Novell Portal Services and OnDemand Services. For supported Web server environments, see “[Web Server, Web Application Server, JVM, and Novell Portal Services](#)” on page 20.
- ♦ **Novell Portal Services (NPS) 1.5:** If you have purchased the full version of NPS 1.5, you must install it before you install OnDemand Services. This allows the OnDemand Services Installation program to copy the OnDemand Services gadgets to your NPS 1.5 portal location and then configure the gadgets.

For information about installing NPS 1.5, see the documentation included with the product or see the Novell Portal Services 1.5 documentation at the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

If you plan to use the runtime version of NPS 1.5 that is included with OnDemand Services, you will be given the opportunity to install it during the OnDemand Services installation process.

- ♦ **OnDemand Services 2:** See the installation instructions in [Chapter 3, “Installing OnDemand Services,”](#) on page 29.
- ♦ **DeFrame 2.01:** If you plan to use DeFrame 2.01 to make thin-client applications available to users through OnDemand Services, you should install DeFrame after installing OnDemand Services.

For information about installing DeFrame, see the *DeFrame Installation and Administration* guide (DEFRA.ME.PDF) on the root of the *Novell ZENworks OnDemand Services 2* CD or see the ZENworks OnDemand Services 2 documentation at the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

3

Installing OnDemand Services

To install Novell® ZENworks® OnDemand Services™, complete the tasks in the following sections:

- ♦ [“Installing the Software” on page 29](#)
- ♦ [“Configuring the Cost Center” on page 43](#)
- ♦ [“Configuring Your Portal” on page 47](#)
- ♦ [“Starting the Maintenance and UsageServer Processes” on page 64](#)

After you’ve completed the above tasks, see [“What’s Next?” on page 65](#) for directions on what you need to do to start using OnDemand Services.

Installing the Software

Before starting the OnDemand Services Installation program, you might want to gather the information you will be prompted for. This information is described in [Appendix A, “Installation Worksheets,” on page 147](#).

You can print the worksheets to record your information, or you can simply review the worksheets to see the information you will be prompted for.

Complete the tasks in the following sections to install the OnDemand Services software:

- ♦ [“Installing the Software” on page 30](#)
- ♦ [“Installing the Software \(excluding NPS 1.5 Runtime\)” on page 32](#)
- ♦ [“Enabling Purchased Desktop or Thin-Client Applications to be Associated with Users” on page 38](#)

- ◆ “Applying Support Pack 1” on page 40
- ◆ “Increasing the Tomcat Servlet Engine’s Heap Size” on page 40
- ◆ “Enabling Extended Character Support” on page 41
- ◆ “Restarting the Web Server, Web Application Server, and JVM” on page 42

Installing the Software

Complete the steps in this section if you don’t already have an existing Novell Portal Services (NPS) 1.5 portal where you want to install the OnDemand Services gadgets.

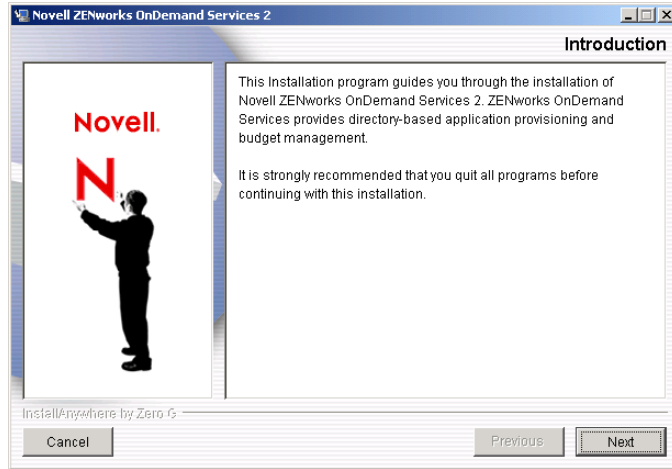
If you do have an existing NPS 1.5 portal, you need to take certain precautions not covered in this section. Skip to “[Installing the Software \(excluding NPS 1.5 Runtime\)](#)” on page 32.

- 1** Make sure you are at a Windows workstation that has drive mappings to the server locations where you will install files.

If you are installing any OnDemand Services components to a Windows 2000 server, we recommend that you install from that server.

- 2** Make sure the LDAP server is configured for Clear Text Passwords. OnDemand Services requires this.
- 3** Insert the *Novell ZENworks OnDemand Services 2* CD.
- 4** Run `odsetup.exe` from the `\ondemand` directory to display the Introduction page.

NOTE: If you copy the `odsetup.exe` file from the CD to another location and attempt to run it from there, it will fail unless you copy the entire `\ondemand` directory structure.



- 5** At the Introduction page, click Next to display the License Agreement page.
- 6** After reviewing the license information, select I Accept the Terms of the License Agreement > click Next to display the Installation Type page.

The Installation program provides two installation options: Typical and Custom. At this point, you should use the Typical option unless you don't have ZENworks for Desktops (ZfD) 3.2 installed. The Typical option installs the OnDemand Services component that provides usage tracking of desktop applications. If you don't have ZfD 3.2, you shouldn't install this component.

- 7** To install all components, click Typical, then click Next to display the Novell Portal Services 1.5 Requirement page.

or

To deselect the OnDemand Components for ZENworks, click Custom, deselect the component, then click Next to display the Novell Portal Services 1.5 Requirement page.

The Novell Portal Services 1.5 Requirement page warns you that OnDemand Services requires NPS 1.5. You should not attempt to install OnDemand Services to an earlier NPS version.

- 8** Click Yes to continue with the installation.

or

Click No to exit the Installation program.

- 9 Follow the prompts to complete the installation. If you filled out the worksheets (see [Appendix A, “Installation Worksheets,” on page 147](#)), use the information you recorded.
- 10 When installation is complete, skip to [“Enabling Purchased Desktop or Thin-Client Applications to be Associated with Users” on page 38](#).

Two installation log files are saved to the c:\documents and settings*username*\local settings\temp directory. The novell_zenworks_ondemand_services_2_installog.xml file contains information about where files were installed. The ondemandinstall.log file contains information about schema extensions added to the Novell eDirectory™ tree.

Installing the Software (excluding NPS 1.5 Runtime)

When using an existing NPS 1.5 portal for OnDemand Services, before beginning the OnDemand Services installation you need to take precautions to ensure that your existing NPS 1.5 portal will not accidentally be overwritten with the NPS 1.5 runtime version included with OnDemand Services. The following sections, [Locating the Nps.pdt File](#) and [“Verifying the Nps.pdt Information” on page 33](#), explain what you need to do. After you’ve reviewed the information in these sections, continue with [“Running the Installation Program” on page 34](#).

Locating the Nps.pdt File

During installation, the OnDemand Services Installation program launches the Novell Portal Services (NPS) 1.5 Installation program. The NPS 1.5 Installation program is designed to install the NPS 1.5 runtime version if it does not detect an existing NPS 1.5 portal. The way it detects an existing installation is through the presence of an NPS.PDT file.

When NPS 1.5 is installed to a NetWare® server, the NPS.PDT file is created in the server’s sys:\system directory regardless of whether the NPS installation was performed from a Windows workstation or from the NetWare server console. If you are installing the OnDemand Services gadgets to an existing NPS portal on a NetWare server, there are no issues with locating the nps.pdt

file. The NPS 1.5 runtime version's Installation program will find the nps.pdt file regardless of the Windows workstation where you perform the OnDemand Services installation.

When NPS1.5 is installed to a Windows server, the nps.pdt file is created on the C: drive of the Windows machine from which you performed the NPS installation. If the installation was performed on the Windows server, the nps.pdt file will be on the server's C: drive. If the installation was performed from another Windows workstation, it will be on that workstation's C: drive. If you are installing the OnDemand Services gadgets to a Windows server, you need to ensure that the correct nps.pdt file is being found. Otherwise, your existing NPS portal will be overwritten by the NPS 1.5 runtime version. To ensure that this does not happen, you have the following options:

- ◆ Run the OnDemand Services Installation program from the same machine used to install NPS 1.5. Check the C:\ directory to ensure that the nps.pdt file is present.
- ◆ Copy the nps.pdt file from the NPS 1.5 installation machine to the C:\ drive of the machine you will use to run the OnDemand Services Installation program.
- ◆ Deselect the Installation program option to install the OnDemand Services gadgets and then manually install the gadgets. This allows you to ignore where the nps.pdt file is located.

Verifying the Nps.pdt Information

The nps.pdt file contains several pieces of information that you need to make sure you correctly match when running the NPS 1.5 runtime version's Installation program. To verify the exact information you need to enter:

- 1** Open the nps.pdt file in a text editor.
- 2** Verify the information in the following properties:

NPSWebServerAddress: When the NPS Installation program prompts you for the Web server's address, you need to enter it exactly as recorded in this property. Otherwise, the Installation program will not recognize the existing NPS portal and will overwrite it. In addition, if the address is a NetWare server name rather than IP address, you need to add the server name to the Windows workstation's hosts file (typically in

c:\winnt\system32\drivers\etc) so that it can resolve the server name to the correct IP address.

NPSDir: Make sure the path listed in this property is correct from the perspective of the machine where you will run the OnDemand Services Installation program. For example, for a NetWare server, you would need to change the directory path from sys:webapps\nps to \\server_name\sys\webapps\nps.

NPSLDAPDir: Make sure the address for the LDAP server that will access eDirectory is correct.

3 Save the file.

Running the Installation Program

To install the software:

1 Make sure you are at a Windows workstation that has drive mappings to the server locations where you will install files.

If you will be installing the OnDemand Services gadgets to an existing NPS 1.5 portal on a Windows 2000 server, we recommend that you install from that server.

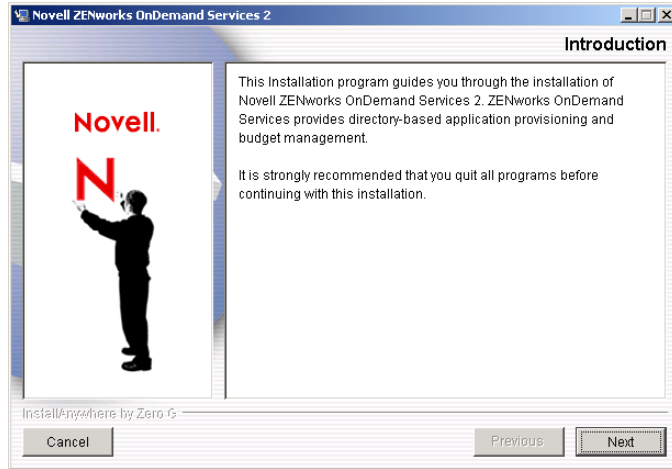
2 Make sure the LDAP server is configured for Clear Text Passwords. OnDemand Services requires this.

3 If you will have the OnDemand Services Installation program install the OnDemand Services gadgets to an existing NPS 1.5 portal, make sure the nps.pdt file, as explained in [Locating the Nps.pdt File](#) above, is in the C:\ directory.

4 Insert the *Novell ZENworks OnDemand Services 2* CD.

5 Run odsetup.exe from the \ondemand directory to display the Introduction page.

NOTE: If you copy the odsetup.exe file from the CD to another location and attempt to run it from there, it will fail unless you copy the entire \ondemand directory structure.



- 6** At the Introduction page, click Next to display the License Agreement page.
- 7** After reviewing the license information, select I Accept the Terms of the License Agreement, then click Next to display the Installation Type page.

The Installation program provides two installation options: Typical and Custom. At this point, you should use the Typical option unless:

- ◆ You want to manually install the OnDemand Services gadgets to an existing NPS 1.5 portal rather than having the Installation program do it. In this case, select Custom and deselect the OnDemand Services Gadgets option.
 - ◆ You don't have ZENworks for Desktops (ZfD) 3.2 installed so you don't need to install the OnDemand Services component that provides usage tracking of desktop applications. In this case, select Custom and deselect the OnDemand Components for ZENworks option.
- 8** To install all components, click Typical, then click Next to display the Novell Portal Services 1.5 Requirement page.

or

To install selected components only, click Custom, click Next, deselect the components you do not want to install, then click Next to display the Novell Portal Services 1.5 Requirement page.

The Novell Portal Services 1.5 Requirement page warns you that OnDemand Services requires NPS 1.5. You should not attempt to install OnDemand Services to an earlier NPS version.

NOTE: If you deselected the OnDemand Services Gadgets component, the Novell Portal Services Requirements page is not displayed. Skip to [Step 10](#).

9 Click Yes to continue with the installation.

or

Click No to exit the Installation program.

10 Follow the prompts to complete the installation. If you filled out the worksheets (see [Appendix A, “Installation Worksheets,” on page 147](#)), use the information you recorded.

IMPORTANT: If you have chosen to install the OnDemand Services gadgets, the NPS Installation program will be launched after you’ve provided the OnDemand Services information. After entering the Web server’s address, you should only be prompted for the Admin user’s context and password and for the context where you want the gadgets to be created. If you are prompted for additional information, such as the Web server’s operating system, type, and Web Application server, either you did not enter the Web server’s address exactly as it is recorded in the NPSWebServerAddress property in the nps.pdt file or the file is not in the expected location. Reenter the Web Server address or verify that the file is in the C:\ directory. If this does not solve the problem, you should cancel the NPS installation. Continuing the installation will result in your existing NPS 1.5 portal being overwritten. After you’ve cancelled the NPS installation and finished the OnDemand Services installation, see [“Manually Installing the OnDemand Services Gadgets” on page 36](#).

11 When installation is complete, if you did not have the Installation program install the OnDemand Services gadgets, continue with the next section, [Manually Installing the OnDemand Services Gadgets](#). Otherwise, skip to [“Enabling Purchased Desktop or Thin-Client Applications to be Associated with Users” on page 38](#).

Manually Installing the OnDemand Services Gadgets

To manually install the gadgets to an existing NPS 1.5 portal:

1 Use a zip utility (for example, WinZip*) to extract the nine OnDemand Services gadget package (.npg) files from the ondemand.npa file. The ondemand.npa file is located in the \ondemand\gadgets\ondemand directory on the *ZENworks OnDemand Services 2* CD.

- 2** Log in to NPS as an administrator.
- 3** Click Administer the Portal.
- 4** Click Gadget, then click Create.
- 5** Enter a name for the gadget you want to create. OnDemand Services includes the following gadgets:

ConfigWizardGadget: Used to store the OnDemand Administrator (ODSADMIN) context and password information in a place where the other gadgets can get it in order to authenticate. You must run the Configuration Wizard before any other OnDemand Services gadgets will work. Recommended gadget name: OD_ConfigWizardGadget.

UserAdminGadget: Used to create, modify, enable, and disable user accounts, and to assign packages to users. Recommended gadget name: OD_UserAdminGadget.

ApprovalGadget: Used to approve or deny user requests for packages. Recommended gadget name: OD_ApprovalGadget.

CurrentBHGadget: Used to select which budget holder is the currently active budget holder and to add or remove budget holders. Recommended gadget name: OD_CurrentBHGadget.

PackageRequestGadget: Used to request packages. Recommended gadget name: OD_PackageRequestGadget.

EditAccountGadget: Used to edit user account information (for example, name, password, and e-mail address). Recommended gadget name: OD_EditAccountGadget.

WorkflowTrackingGadget: Used to check the status of package requests and to cancel requests. Recommended gadget name: OD_WorkflowTrackingGadget.

LaunchItemGadget: Used to launch applications. Recommended gadget name: OD_LaunchItemGadget.

ReportGadget: Used to create reports showing purchase and usage costs for packages. Recommended gadget name: OD_ReportGadget.

- 6** Browse to the location where you extracted the gadget's package file (.npg) and select the file. The following table lists the package filenames for the gadgets.

Gadget	Package Filename
ConfigWizardGadget	OD_ConfigWizardGadget.npg
UserAdminGadget	OD_UserAdminGadget.npg
ApprovalGadget	OD_ApprovalGadget.npg
CurrentBHGadget	OD_CurrentBHGadget.npg
PackageRequestGadget	OD_PackageRequestGadget.npg
EditAccountGadget	OD_EditAccountGadget.npg
WorkflowTrackingGadget	OD_WorkflowTrackingGadget.npg
LaunchItemGadget	OD_LaunchItemGadget.npg
ReportGadget	OD_ReportGadget.npg

- 7** Click Create, then click OK to add the gadget to the gadgets list.
- 8** Repeat **Step 4** through **Step 7** until you've created all of the OnDemand Services gadgets.
- 9** Continue with the next section, **Enabling Purchased Desktop or Thin-Client Applications to be Associated with Users**.

Enabling Purchased Desktop or Thin-Client Applications to be Associated with Users

This section applies only if you are using ZENworks for Desktops (ZfD) 3.2 to provision desktop applications or Novell DeFrame™ 2.01 to provision thin-client applications.

OnDemand Services uses a ZfD 3.2 support library to associate desktop and thin-client applications with users who have purchased the applications. On a NetWare server, the support library is zenapp32.nlm. On a Windows server, it is zenapp32.dll. This support library must reside on the server where your NPS 1.5 portal (with the OnDemand Services gadgets) is located.

If your NPS 1.5 portal is on a NetWare server, you should have installed the zenapp32.nlm to the sys:\system directory on that server when you ran the OnDemand Services Installation program. If you did not or you installed the OnDemand Services gadgets to multiple NPS NetWare servers, you need to

copy the zenapp32.nlm file to the NetWare server or servers. See [Copying Zenapp32.dll or Zenapp32.nlm to the NPS Server](#) below for instructions.

If your NPS 1.5 portal is on a Windows server, you need to complete the tasks in the following sections to ensure that OnDemand Services has access to zenapp32.dll and that the DLL has appropriate eDirectory access:

- ◆ [“Logging in as a User Who Has Trustee Rights to Application Objects” on page 39](#)
- ◆ [“Copying Zenapp32.dll or Zenapp32.nlm to the NPS Server” on page 39](#)

Logging in as a User Who Has Trustee Rights to Application Objects

To associate an application with a user, zenapp32.dll needs to have rights to the Application object in eDirectory. This requires the following:

- ◆ The Novell Client™ to be installed on the Windows server.
- ◆ A user login from the Windows server to eDirectory. The eDirectory user account must provide rights to all Application objects. By default, the ODSAdmin user (created during installation) has all required rights. You can log in as the ODSAdmin user or any other user with the required rights (Admin, for example).

Copying Zenapp32.dll or Zenapp32.nlm to the NPS Server

If zenapp32.dll is not already installed on the Windows server where your NPS portal and OnDemand Services gadgets reside, you can use either of the following methods to install it:

- ◆ Copy the file from the sys:\public directory of an existing ZENworks server to the NPS Windows server and then add the zenapp32.dll directory to the server's PATH variable (if necessary).
- ◆ Modify the ODSAdmin user's login script (or the login script of the user account the Windows server is using to log into eDirectory) to launch Novell Application Launcher™. This will cause Novell Application Launcher, including zenapp32.dll, to be copied from the specified ZENworks server to the NPS Windows server. For instructions, see Application Management in the ZENworks for Desktops 3.2 *Administration Guide* at the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

IMPORTANT: You must use a version of zenapp32.dll dated 1/31/2002 or later. Using an earlier version will cause launched applications to fail.

If zenapp32.nlm is not already installed on the NetWare server where your NPS portal and OnDemand Services gadgets reside, you need to:

- ♦ Copy the file from the ondemand\services\system directory on the *Novell ZENworks OnDemand Services 2* CD to the sys:\system directory of the NetWare server.

Applying Support Pack 1

OnDemand Services 2 Support Pack 1 resolves a variety of issues and increases the reliability and stability of OnDemand Services 2. We recommend that you apply the support pack to the software components you just installed.

You can download Support Pack 1 from the [Novell Support site \(http://support.novell.com/servlet/tidfinder/2964553\)](http://support.novell.com/servlet/tidfinder/2964553).

To update the OnDemand Services software, follow the instructions included in the support pack's Readme_ods2sp1.txt file.

Increasing the Tomcat Servlet Engine's Heap Size

To ensure that the Tomcat Java servlet engine has sufficient memory to operate properly, you need to increase its initial heap size.

The amount you increase the Java servlet engine's heap size is dependent on the server's available memory, the number of OnDemand Services users that will be supported, and so forth. We recommend that the minimum heap size memory allocation be no less than 128 MB for the first 200 users and that you increase the allocation by 64 MB for every 200 additional users. The maximum heap size memory allocation should be limited only by the amount of the server's memory you want allocated to the Java process.

NetWare Server

To modify the Java servlet engine's heap size:

- 1** Open the tomcat33.ncf file, located in the sys:\tomcat\33\bin directory.
- 2** On the Java line, change the initial heap size memory allocation parameter (Xms), the maximum heap size memory allocation parameter (Xmx), and the heap size buffer allocation parameter (Xss).

The following example shows the recommended parameter settings for supporting up to 200 users.

```
-Xms128m -Xmx400m -Xss256k
```

The maximum heap size parameter (Xmx) should reflect the maximum amount of the server's memory you want used by the Java servlet engine. On NetWare, this amount cannot exceed 400 MB; if you enter a value greater than 400, the Java servlet engine will revert to using only 32 MB.

- 3** Save the file.
- 4** Skip to [Enabling Extended Character Support](#) below.

Windows 2000 Server

To modify the Java servlet engine's heap size:

- 1** Open the tomcat.bat file, located in the c:\jakarta_tomcat-3.3x\bin directory.
- 2** Add the following settings to the TOMCAT_OPTS environment variable. If the file doesn't include a TOMCAT_OPTS environment variable, you will need to add it.

```
set TOMCAT_OPTS= -Xms128m -Xmx512m -Xss256k
```

The above example shows the recommended initial heap size memory parameter (Xms) setting for supporting up to 200 users. The maximum heap size parameter (Xmx) should reflect the maximum amount of the server's memory you want used by the Java servlet engine.

- 3** Save the file.
- 4** Continue with the next section, [Enabling Extended Character Support](#).

Enabling Extended Character Support

If you want to support extended characters, you need to configure the Tomcat servlet engine and Novell Portal Services to support extended characters.

Configuring the Tomcat Servlet Engine

For the Tomcat servlet engine to support extended characters (for example, in package and application names displayed in the OnDemand Services gadgets), you need to enable extended character support.

- 1** On a NetWare server, open the `nwserver.xml` file, located in the `sys:\tomcat\33\conf` directory.

or

On a Windows 2000 server, open the `server.xml` file, located in the `c:\tomcat\conf` directory.

- 2** Locate the `DecodeInterceptor` tag.
- 3** Add the following attributes to the tag:

```
defaultEncoding="UTF-8" useSessionEncoding="False"
```

The resulting tag should be similar to this:

```
<DecodeInterceptor defaultEncoding="UTF-8"  
useSessionEncoding="False" />
```

- 4** Save the file.
- 5** Continue with the next section, [Restarting the Web Server, Web Application Server, and JVM](#).

Configuring the Portal

- 1** Open the `NPS PortalServlet.properties` file. It is located in the `webapps\nps\WEB-INF` directory
- 2** Add the following entry

```
System.DecodeRequest=true
```

Restarting the Web Server, Web Application Server, and JVM

To ensure that changes made during installation of the software are recognized by your Web server:

- 1** Restart the Web server, Web Application server, and Java Virtual Machine. If necessary, refer to your product's documentation for instructions.
- 2** Continue with the next section, [Configuring the Cost Center](#).

Configuring the Cost Center

OnDemand Services requires a cost center in order to record purchase and usage charges for packages. The OnDemand Services Installation program helped you create your first cost center.

You need to configure the cost center by adding budget holders and assigning users. The cost center must have at least one budget holder, and any users whose purchases should be recorded against the cost center need to be assigned to the cost center. When a user requests a package, the budget holder for the user's cost center can approve or deny the request. If the request is approved, the charges associated with the package are recorded against the cost center.

The following sections explain how to assign budget holders and users to the cost center created during installation. You might want to include yourself as a budget holder and user for testing purposes.

- ◆ [“Adding Budget Holders” on page 43](#)
- ◆ [“Assigning Users” on page 45](#)

Unless you don't need to bill package charges to individual budgets within your organization, you'll probably want to create additional cost centers before making OnDemand Services available to your users. Instructions for doing so are provided in [Chapter 14, “Creating Cost Centers,” on page 121](#).

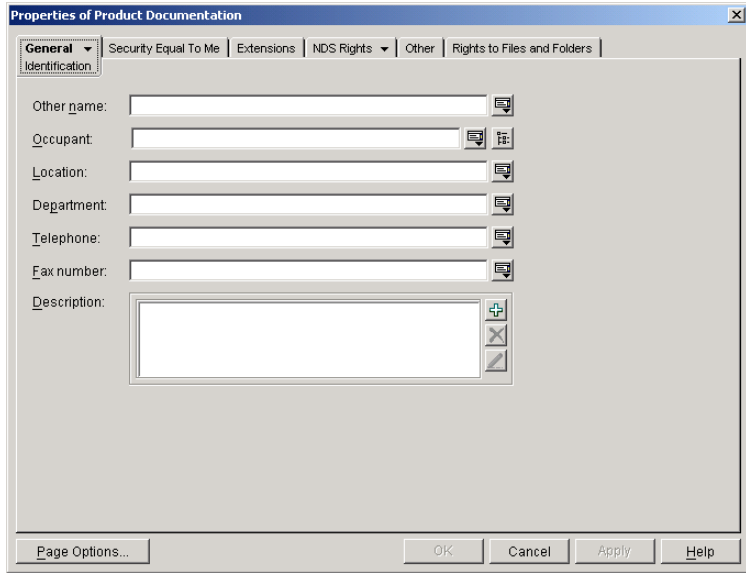
Adding Budget Holders

A cost center must have at least one budget holder. The cost center's budget holder is responsible for approving or denying package purchase requests from the cost center's users. A cost center can have multiple budget holders, but only one of them, referred to as the current budget holder, is active at one time. All other budget holders are referred to as backup budget holders. A backup budget holder can be assigned as the current budget holder at any time.

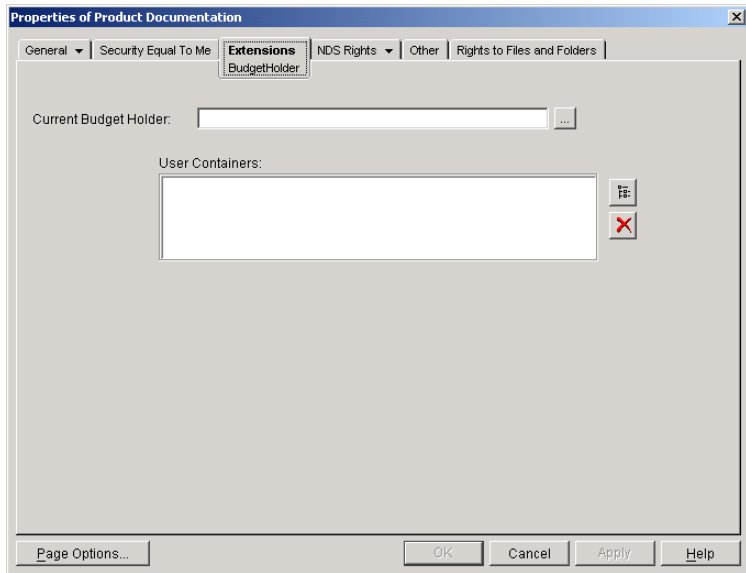
To add budget holders and assign the current budget holder:

- 1 In ConsoleOne, right-click the cost center's object > click Properties.

The cost center is an Organizational Role object. You can find it by looking for the object in the context specified during installation.



- 2 On the Identification page, use the Occupant field to add the users who will be the cost center's budget holders.
- 3 Click the Extensions tab to display the Budget Holder page.



4 Fill in the following fields:

Current Budget Holder: The cost center can have more than one budget holder. This allows for reassignment of the budget holder responsibilities from one person to another (for example, if a budget holder goes on vacation). The current budget holder is the budget holder who is currently active. He or she will receive package requests, can assign packages to users, and can add or delete users. Select the user you want to assign as the current budget holder.

You can add a user who has not been defined as an Occupant on the Identification page (see [Step 2](#)). If you do, you need to manually assign the user to the cost center's Security Equal to Me list.

NOTE: Budget holders can use the Current Budget Holder gadget to change the current budget holder assignment. Changing it each time does not require you to edit this page.

User Containers: Budget holders can create user accounts in eDirectory. By default, the user accounts will be created in the same container as the cost center object. If you add one container to this list, the users will be created in that container. If you add two or more containers, the budget holder will be able to select which container a user is created in.

5 Click OK to save your budget holder changes.

6 Continue with the next section, [Assigning Users](#).

Assigning Users

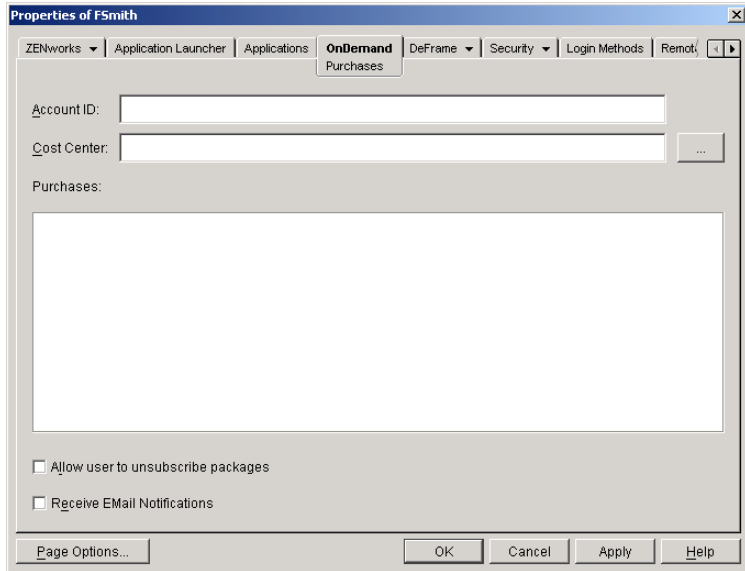
Any users whose purchase charges should be accrued to the cost center need to be assigned to the cost center.

If a user already exists in eDirectory, you must use ConsoleOne to assign him or her to the cost center. If the user doesn't exist in eDirectory, one of the cost center's budget holders can create the user's account through the OnDemand Services User Administration gadget, at which time the user will automatically be assigned to the cost center.

To assign an existing user to the cost center:

1 In ConsoleOne, right-click the User object > click Properties.

2 Click the OnDemand tab to display the Purchases page.



3 Fill in the following fields:

Account ID: The user's account ID is recorded on each Purchase and Usage object that he or she generates. If you have a third-party billing system or reporting system, you can base your billing or reporting on the account ID.

The account ID can be a unique identifier, such as the user's employee workforce number, or it can be a common identifier, such as the user's cost center ID as it is defined in the billing or reporting system.

Cost Center: Browse to and select the cost center (represented by an Organizational Role object) that you want the user assigned to.

Allow User to Unsubscribe to Packages: If you want the user to be able to cancel his or her subscription to a package, select this option. If you do so, the user can use the Workflow Tracking gadget to view an approved subscription purchase and cancel the subscription.

Receive E-Mail Notifications: Select this option to enable the user to receive e-mail messages notifying him or her of approved or denied purchases.

4 Click OK to save the information.

5 Continue with the next section, [Configuring Your Portal](#).

Configuring Your Portal

Before OnDemand Services can be used, you need to complete the tasks in the following sections to configure your portal.

- ♦ “Creating Portal Pages” on page 47
- ♦ “Running the OnDemand Services Configuration Wizard” on page 62
- ♦ “Restarting the Web Server, Web Application Server, and JVM” on page 63

Creating Portal Pages

OnDemand Services includes nine gadgets. Each of these gadgets represents a window to specific content or functionality provided by OnDemand Services. Some of the gadgets are intended for users, some for budget holders, and some for administrators. The following list describes the gadgets and their intended audience:

ConfigWizardGadget (Administrator): Used to store the OnDemand Administrator (ODSAdmin) context and password in a place where the other gadgets can get it in order to authenticate to eDirectory. The Configuration Wizard must be run before any other OnDemand Services gadgets will work.

UserAdminGadget (Budget Holder): Used to create, modify, enable, and disable user accounts, and to assign packages to users.

ApprovalGadget (Budget Holder): Used to approve or deny package purchase requests submitted by users.

CurrentBHGadget (Budget Holder): Used to select which budget holder is the current budget holder and to add or remove budget holders.

PackageRequestGadget (User): Used to submit a purchase request for a package.

EditAccountGadget (User): Used to edit user account information such as name, password, and e-mail address.

WorkflowTrackingGadget (User): Used to check the status of package purchase requests and to cancel requests.

LaunchItemGadget (User): Used to launch purchased applications.

ReportGadget (Administrator, Budget Holder, User): Used to generate expense reports based on package purchases and usage.

TabbedContainerGadget (Administrator, Budget Holder, User): Used to enable the other gadgets to be displayed in a preconfigured tabbed-page format. This is an alternate page design to the one explained below. For information about implementing tabbed pages, see [Appendix C, “Tabbed Portal Pages,” on page 171](#).

You need to design your portal to give users, budget holders, and administrators access to the appropriate gadgets. There are many designs you can use. The instructions in the following sections help you create a simple portal. If you have already installed the full version of NPS 1.5 and implemented a portal design, you should still review the following sections to see how you need to configure the gadgets as you add them to your existing portal.

- ◆ [“Logging in to NPS” on page 48](#)
- ◆ [“Creating and Assigning User Pages” on page 49](#)
- ◆ [“Creating and Assigning Budget Holder Pages” on page 55](#)
- ◆ [“Creating and Assigning the Administration Page” on page 60](#)

Logging in to NPS

You need to log in to NPS as a portal administrator. By default, the Admin user you entered when installing NPS is configured as a portal administrator.

To log in to NPS:

- 1** In your Web browser, enter the following URL:

```
http://web_server_address/nps
```

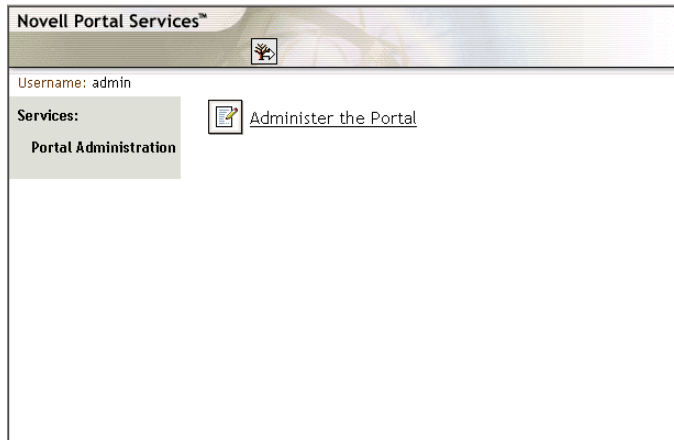
where *web_server_address* is the IP address or host name of the Web server running NPS. If you are using SSL, make sure to enter https rather than http.

- 2** Enter the Admin user’s name and password.

If the Admin user is not in one of the User contexts you specified when installing NPS, you will need to enter the Admin user’s distinguished name using LDAP syntax. For example:

```
cn=admin,ou=users,o=novell
```


3 Click Login to display the Services page.



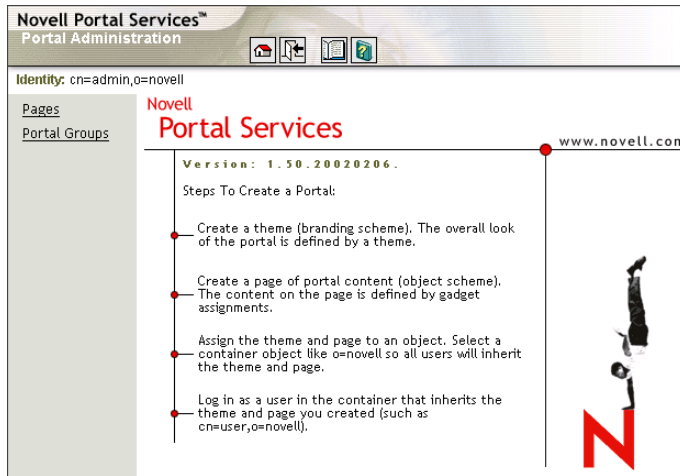
4 Continue with the next section, [Creating and Assigning User Pages](#).

Creating and Assigning User Pages

You will create three pages that all users will have access to:

Example Page Name	Gadgets Included on Page
Applications	OD_LaunchItemGadget OD_PackageRequestGadget
My Account	OD_EditAccountGadget OD_WorkflowTrackingGadget
Reporting	OD_ReportGadget

1 Click Administer the Portal to display the Portal Administration page.



- 2 Click Pages.
- 3 Continue with the next section, [Creating and Assigning the Applications Page](#).

Creating and Assigning the Applications Page

- 1 In the list, select Create New Top Level Page, then click Create to display the Create a New Page page.
- 2 In the Name field, enter Applications, click Create to display the Page Saved page, then click Edit to edit the Applications page.
- 3 Add the Launch Item gadget to the page. To do so:
 - 3a Click Add, select OD_LaunchItemGadget in the gadgets list, then click Add to display the gadget's configuration page.
 - 3b In the Display Name field, enter the gadget's name as you want it displayed on the page. For example, you could use Launch Applications.
 - 3c If you want to enable users to launch desktop applications (ZENworks for Desktops applications), select TRUE for the Display Workstation Applications setting.
 - 3d In the Secure Sensitive Pages field, select TRUE if the Web server supports SSL; otherwise, select FALSE.

3e Modify any other configuration settings.

To get a short description of each field, click the Descriptions button at the bottom of the page. For more detailed information about the fields, see [“Gadget Configuration Settings” on page 157](#).

3f Click Continue to add the configured gadget to the Applications page.

4 Add the Package Request gadget to the Applications page. To do so:

4a Click Add, select OD_PackageRequestGadget, then click Add to display the gadget’s configuration page.

4b In the Display Name field, enter the gadget’s name as you want it displayed on the page. For example, you could use Request Applications.

4c In the OnDemandService Object field, enter the distinguished name of the OnDemandService object that was created in eDirectory during installation. Use LDAP syntax to specify the name (for example, cn=ondemandservice,ou=services,o=novell).

4d Modify any other configuration settings.

To get a short description of each field, click the Descriptions button at the bottom of the page. For more detailed information about the fields, see [“Gadget Configuration Settings” on page 157](#).

4e Click Continue to add the configured gadget to the Applications page.

5 Click Save to save the page, then click Assign to display the Assign Page to Objects page.

6 Assign the page to users. To do so:

6a Click Add.

You can assign the page to users by adding individual User objects, Group objects in which the users are members, or container objects where the users reside. By default, no objects are listed. You need to search for the objects that you want to add.

6b In the Search For field, select the type of object you want to search for, enter the search criteria in the Named field, then click Search to display all objects that match the search criteria.

- 6c** Select the objects you want to assign the page to, then click Assign.
- 6d** Repeat **Step 6a** through **Step 6c** to assign the page to additional objects.
- 6e** Click Close.
- 7** Configure the layout of the page. To do so:
 - 7a** In the list of pages, select the Applications page, then click Edit.
 - 7b** Click Layout Gadget Assignments.

Both the Launch Applications and Request Applications gadgets are listed under column 1. To display the gadgets in two-column format:
 - 7c** Select the Request Application gadget, then click the To Column 2 button.
 - 7d** Click Continue.
- 8** Click Save to save your changes.
- 9** Continue with the next section, **Creating and Assigning the My Account Page**.

Creating and Assigning the My Account Page

- 1** In the Pages list, select Create New Top Level Page, then click Create to display the Create a New Page page.
- 2** In the Name field, enter My Account, click Create to display the Page Saved page, then click Edit to edit the My Account page.
- 3** Add the Edit Account Information gadget to the My Account page. To do so:
 - 3a** Click Add, select OD_EditAccountGadget in the gadgets list, then click Add to display the gadget's configuration page.
 - 3b** In the Display Name field, enter the gadget's name as you want it displayed on the page. For example, you could use Edit Account Information.
 - 3c** In the Secure Sensitive Pages field, select TRUE if the Web server supports SSL; otherwise, select FALSE.

3d Modify any other configuration settings.

To get a short description of each field, click the Descriptions button at the bottom of the page. For more detailed information about the fields, see [“Gadget Configuration Settings” on page 157](#).

3e Click Continue to add the configured gadget to the My Account page.

4 Add the Workflow Tracking gadget to the My Account page. To do so:

4a Click Add, select OD_WorkflowTrackingGadget, then click Add to display the gadget’s configuration page.

4b In the Display Name field, enter the gadget’s name as you want it displayed on the page. For example, you could use Track Requests.

4c In the OnDemandService Object field, enter the distinguished name of the OnDemandService object that was created in eDirectory during installation. Use LDAP syntax to specify the name (for example, cn=ondemandservice,ou=services,o=novell).

4d Modify any other configuration settings.

To get a short description of each field, click the Descriptions button at the bottom of the page. For more detailed information about the fields, see [“Gadget Configuration Settings” on page 157](#).

4e Click Continue to add the configured gadget to the My Account page.

5 Click Save to save the page, then click Assign to display the Assign Page to Objects page.

6 Assign the page to users. If you need instructions, see [Step 6a through Step 6d on page 52](#).

7 When finished assigning the page to users, click Close.

8 Continue with the next section, [Creating and Assigning the Reporting Page](#).

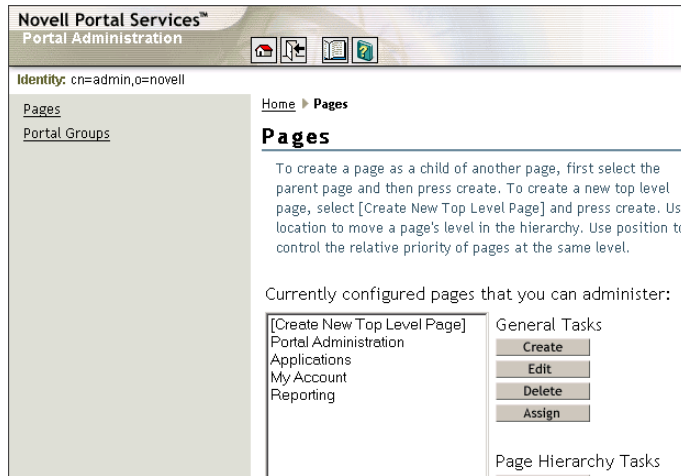
Creating and Assigning the Reporting Page

1 In the list, select Create New Top Level Page, then click Create to display the Create a New Page page.

- 2** In the Name field, enter Reporting, click Create to display the Page Saved page, then click Edit to edit the Reporting page.
- 3** Add the Reporting gadget to the Reporting page. To do so:
 - 3a** Click Add,select OD_ReportingGadget in the gadgets list, then click Add to display the gadget’s configuration page.
 - 3b** In the Display Name field, enter the gadget’s name as you want it displayed on the page. For example, you could use Create Purchase Reports.
 - 3c** In the OnDemandService Object field, enter the distinguished name of the OnDemandService object that was created in eDirectory during installation. Use LDAP syntax to specify the name (for example, cn=ondemandservice,ou=services,o=novell).
 - 3d** Modify any other configuration settings.

To get a short description of each field, click the Descriptions button at the bottom of the page. For more detailed information about the fields, see [“Gadget Configuration Settings” on page 157](#).
 - 3e** Click Continue to add the configured gadget to the Reporting page.
- 4** Click Save to save the page, then click Assign to display the Assign Page to Objects page.
- 5** Assign the page to users. If you need instructions, see [Step 6a](#) through [Step 6d](#).
- 6** When finished assigning the page to users, click Close.

The three pages you created should now be displayed in the Pages list.



7 To create the budget holder pages, continue with the next section, [Creating and Assigning Budget Holder Pages](#).

Creating and Assigning Budget Holder Pages

You will create two pages that all budget holders will have access to.

Example Page Name	Gadgets Included on Page
Cost Center Management	OD_UserAdminGadget OD_CurrentBHGadget
Purchase Approval	OD_ApprovalGadget

Creating a Budget Holder Portal Group

A budget holder portal group lets NPS automatically grant page access to any users who are defined as cost center budget holders. A portal group is the best way to assign your budget holder pages because it is updated on-the-fly as users are added or removed as budget holders.

- 1** Click Portal Groups > click Create to display the Create Portal Group page.
- 2** In the Name field, enter a name for the portal group (for example, BudgetHolders).

3 In the Query Filter String, enter the following:

```
( |(securityequals=cn=cost_center1_context)
(securityequals=cn=cost_center2_context)
(securityequals=cn=cost_center3_context)... )
```

where *cost_center_context* is the context (using LDAP syntax) of the Organizational Role objects that are serving as OnDemand Services cost centers. For example:

```
( |(securityequals=cn=SalesCC,ou=services,o=novell)
(securityequals=cn=MarketingCC,ou=services,o=novell) )
```

In this example, all users who have been added as budget holders on the SalesCC and MarketingCC cost centers will be included in the portal group.

4 Click Create, then click OK to create the portal group

5 To make sure the query filter string returns the desired results, you should test it. To do so:

5a Select the portal group, click Edit, then click Properties.

5b Click Text (next to the Query Filter String field).

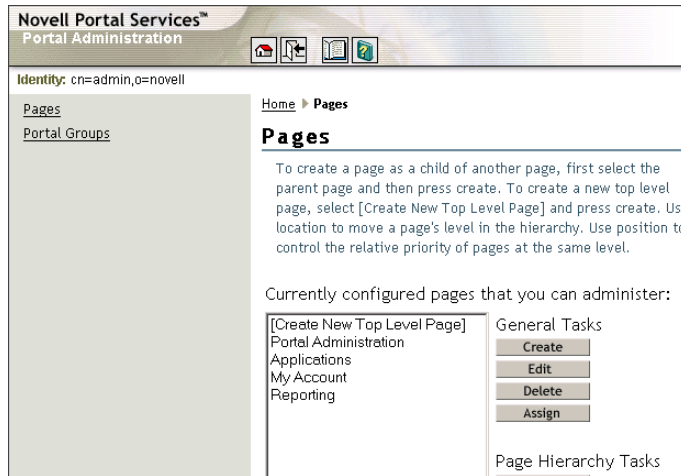
5c If the query returned the correct results, click Cancel, then continue with the next section, [Creating and Assigning the Cost Center Management Page](#).

or

If the test does not return any matches, click Cance, verify that you've entered the correct information in the query filter strin, then retest the query.

Creating and Assigning the Cost Center Management Page

1 Click Pages.



- 2** In the list, select **Create New Top Level Page**, then click **Create** to display the **Create a New Page** page.
- 3** In the **Name** field, enter **Cost Center Management**, click **Create** to display the **Page Saved** page, then click **Edit** to edit the **Cost Center Management** page.
- 4** Add the **User Administration** gadget to the **Cost Center Management** page. To do so:
 - 4a** Click **Add**, select **OD_UserAdminGadget** in the gadgets list, then click **Add** to display the gadget's configuration page.
 - 4b** In the **Display Name** field, enter the gadget's name as you want it displayed on the page. For example, you could use **Manage Users**.
 - 4c** In the **Secure Sensitive Pages** field, select **TRUE** if the Web server supports **SSL**; otherwise, select **FALSE**.
 - 4d** In the **OnDemandService Object** field, enter the distinguished name of the **OnDemandService** object that was created in **eDirectory** during installation. Use **LDAP** syntax to specify the name (for example, **cn=ondemandservice,ou=services,o=novell**).
 - 4e** Modify any other configuration settings.

To get a short description of each field, click the Descriptions button at the bottom of the page. For more detailed information about the fields, see [“Gadget Configuration Settings” on page 157](#).

- 4f** Click Continue to add the configured gadget to the Cost Center Management page.
- 5** Add the Current Budget Holder gadget to the Cost Center Management page. To do so:
 - 5a** Click Add, select OD_CurrentBHGadget, then click Add to display the gadget’s configuration page.
 - 5b** In the Display Name field, enter the gadget’s name as you want it displayed on the page. For example, you could use Change Current Budget Holder.
 - 5c** In the OnDemandService Object field, enter the distinguished name of the OnDemandService object that was created in eDirectory during installation. Use LDAP syntax to specify the name (for example, cn=ondemandservice,ou=services,o=novell).
 - 5d** Modify any other configuration settings.

To get a short description of each field, click the Descriptions button at the bottom of the page. For more detailed information about the fields, see [“Gadget Configuration Settings” on page 157](#).
 - 5e** Click Continue to add the configured gadget to the Cost Center Management page.
- 6** Click Save to save the page, then click Assign to display the Assign Page to Objects page.
- 7** Assign the Cost Center Management page to budget holders. To do so:
 - 7a** Click Add.

You need to assign the page to the budget holders portal group you created in [“Creating a Budget Holder Portal Group” on page 55](#).
 - 7b** In the Search For field, select Portal Groups > enter the name of the portal group in the Named field, then click Search to display the portal group’s context.
 - 7c** Select the portal group, then click Assign.

7d Click Close.

8 Continue with the next section, [Creating and Assigning the Purchase Approval Page](#).

Creating and Assigning the Purchase Approval Page

1 In the list, select Create New Top Level Page, then click Create to display the Create a New Page page.

2 In the Name field, enter Purchase Approval > click Create to display the Page Saved page, then click Edit to edit the Purchase Approval page.

3 Add the Approval gadget to the page. To do so:

3a Click Add, select OD_ApprovalGadget in the gadgets list, then click Add to display the gadget's configuration page.

3b In the Display Name field, enter the gadget's name as you want it displayed on the page. For example, you could use Approve/Deny Purchase Requests

3c In the OnDemandService Object field, enter the distinguished name of the OnDemandService object that was created in eDirectory during installation. Use LDAP syntax to specify the name (for example, cn=ondemandservice,ou=services,o=novell).

3d Modify any other configuration settings.

To get a short description of each field, click the Descriptions button at the bottom of the page. For more detailed information about the fields, see [“Gadget Configuration Settings” on page 157](#).

3e Click Continue to add the configured gadget to the Purchase Approval page.

4 Click Save to save the page, then click Assign to display the Assign Page to Objects page.

4a Assign the Purchase Approval page to budget holders. To do so:

4b Click Add.

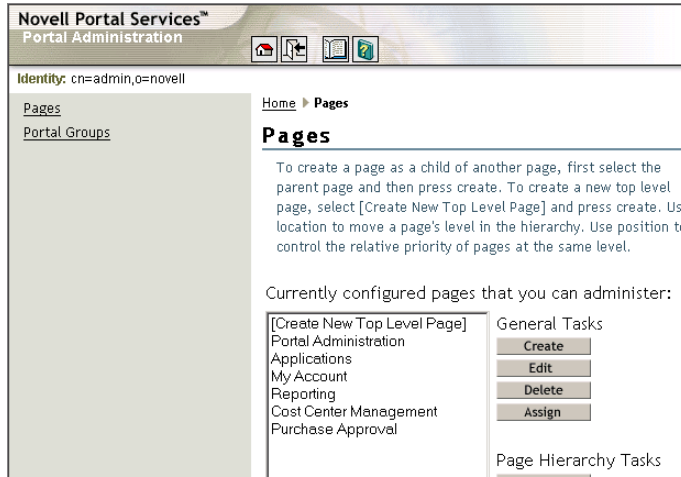
You need to assign the page to the budget holders portal group you created in [“Creating a Budget Holder Portal Group” on page 55](#).

4c In the Search For field, select Portal Groups, enter the name of the portal group in the Named field, then click Search to display the portal group's context.

4d Select the portal group, then click Assign.

4e Click Close.

The two budget holder pages you created should now be displayed in the Pages list.



5 To create the OnDemand Administration page, continue with the next section, **Creating and Assigning the Administration Page**.

Creating and Assigning the Administration Page

You will create an OnDemand Administration page that the Admin user will have access to.

Example Page Name	Gadgets
OnDemand Administration	OD_ConfigWizardGadget

1 In the list, select Create New Top Level Page, then click Create to display the Create a New Page page.

- 2** In the Name field, enter OnDemand Administration, click Create to display the Page Saved page, then click Edit to edit the OnDemand Administration page.
- 3** Add the Configuration Wizard gadget to the OnDemand Administration page. To do so:
 - 3a** Click Add > select OD_ConfigWizardGadget in the gadgets list > click Add to display the gadget's configuration page.
 - 3b** In the Display Name field, enter the gadget's name as you want it displayed on the page. For example, you could use Configuration Wizard.
 - 3c** In the Secure Sensitive Pages field, select TRUE if the Web server supports SSL; otherwise, select FALSE.
 - 3d** Modify any other configuration settings.

To get a short description of each field, click the Descriptions button at the bottom of the page. For more detailed information about the fields, see [“Gadget Configuration Settings” on page 157](#).
 - 3e** Click Continue to add the configured gadget to the OnDemand Administration page.
- 4** Click Save to save the page, then click Assign to display the Assign Page to Objects page.
- 5** Assign the OnDemand Administration page to the Admin user.
 - 5a** Click Add.
 - 5b** In the Object Name field, enter the context of the Admin user (for example, cn=admin,o=novell) > click Assign.
 - 5c** Click Close.

The administrator page should now be displayed in the Pages list.



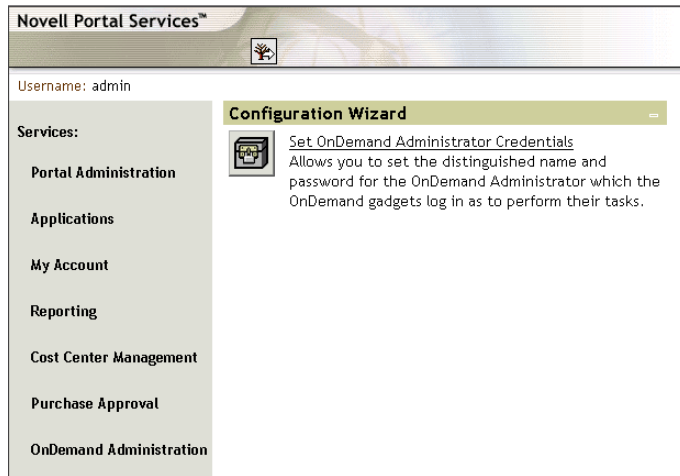
- 6** Continue with the next section, **Running the OnDemand Services Configuration Wizard**.

Running the OnDemand Services Configuration Wizard

You must run the Configuration Wizard before the other OnDemand Services gadgets will function properly. The OnDemand Services gadgets authenticate to eDirectory through the OnDemand Administrator (ODSAdmin) object created during installation. The OnDemand Administrator object has the rights required by the gadgets to perform their tasks in eDirectory.

When you run the Configuration Wizard, it stores the OnDemand Administrator context and password in a location that the other gadgets can access.

- 1** Log out of the portal and log back in to refresh the portal pages.
- 2** Open the page where you added the Configuration Wizard.



- 3** Click [Set OnDemand Administrator Credentials](#) to display the Set Credentials window.
- 4** In the Distinguished Name field, enter the full context of the OnDemand Administrator (ODSAdmin) object (for example, `cn=ODSAdmin,ou=services,o=novell`).
- 5** In the Password field, enter the OnDemand Administrator password. This password must be the password that was entered during installation.
- 6** Click Update.
- 7** Continue with the next section, [Restarting the Web Server, Web Application Server, and JVM](#).

Restarting the Web Server, Web Application Server, and JVM

To ensure that the configuration changes made during installation are recognized:

- 1** If you did not already change the Tomcat servlet engine's configuration to increase its heap size, do so now. For instructions, see [“Increasing the Tomcat Servlet Engine's Heap Size”](#) on page 40.
- 2** Restart the Web server, Web Application server, and Java Virtual Machine. If necessary, refer to your product's documentation for instructions.
- 3** Continue with the next section, [Starting the Maintenance and UsageServer Processes](#).

Starting the Maintenance and UsageServer Processes

The Maintenance process cleans up purchase and usage records in eDirectory and closes out usage tracking of applications when the UsageServer becomes unavailable. It needs to be running at all times.

The UsageServer process enables application usage to be tracked. It needs to be running if you want to be able to charge for an application based on the number of times or minutes it is used.

- ♦ “Starting the Maintenance Process” on page 64
- ♦ “Starting the UsageServer Process” on page 64

Starting the Maintenance Process

1 At the NetWare server console, enter `commercemaintenance.ncf`.

or

At the Windows server command prompt, change to the directory where the Maintenance process was installed (by default, `\ondemand\bin`), then enter `commercemaintenance.bat`.

2 Enter the password for the Admin or Admin-equivalent user that the Usage Server will use to authenticate to eDirectory through the LDAP server.

If you can't remember which user is being used for authentication, the user's context is stored in the `commerceserver.cfg` file, located in the `\ondemand` directory.

You can install additional Maintenance processes or change the default configuration for this process. For information, see [Chapter 17, “Managing the Maintenance and UsageServer Processes,”](#) on page 129.

Starting the UsageServer Process

1 At the NetWare server console, enter `usageserver.ncf`.

or

At the Windows server command prompt, change to the directory where the Usage Server was installed (by default, `\ondemand\bin`), then enter `usageserver.bat`.

- 2 Enter the password for the Admin or Admin-equivalent user that the Usage Server will use to authenticate to eDirectory through the LDAP server.

If you can't remember which user is being used for authentication, the user's context is stored in the `commerceserver.cfg` file, located in the `\ondemand` directory.

You can install additional UsageServer processes or change the default configuration for this process. For information, see [Chapter 17, "Managing the Maintenance and UsageServer Processes,"](#) on page 129.

What's Next?

OnDemand Services should now be functional. You, your users, and your budget holders can log in to your NPS portal to access the OnDemand Services gadgets.

At this point, however, there should not be any application packages available to users in OnDemand Services. To make applications available, you can:

- ♦ Create the application packages. See [Chapter 5, "Creating Application Packages,"](#) on page 73.
- ♦ Deliver applications free of charge. See [Chapter 6, "Distributing Free Applications,"](#) on page 85.

In addition, you can:

- ♦ Review the issues you should be aware of as you support users who purchase packages that include desktop and thin-client applications. See [Chapter 12, "Supporting Desktop Application Users,"](#) on page 115.
- ♦ Limit the availability of application packages so that only specific users have access to them. See [Chapter 8, "Limiting Package Availability,"](#) on page 95.
- ♦ Create additional cost centers and add users to the cost centers. See [Chapter 14, "Creating Cost Centers,"](#) on page 121.
- ♦ Install Novell DeFrame 2.01 in your terminal server environment so that you can include thin-client applications in your packages. See the *DeFrame Installation and Administration Guide* (`deframe_en.pdf`) in the `\DEFRAME` on the *Novell ZENworks OnDemand Services 2* CD. It is also available under the ZENworks OnDemand Services 2 documentation at the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).



Administration

- Chapter 4, “Administration Basics,” on page 69
- Chapter 5, “Creating Application Packages,” on page 73
- Chapter 6, “Distributing Free Applications,” on page 85
- Chapter 7, “Tracking Application Usage,” on page 89
- Chapter 8, “Limiting Package Availability,” on page 95
- Chapter 9, “Securing Web Applications With Novell iChain,” on page 99
- Chapter 10, “Organizing Packages and Applications,” on page 107
- Chapter 11, “Customizing Notification Messages,” on page 111
- Chapter 12, “Supporting Desktop Application Users,” on page 115
- Chapter 13, “Supporting Thin-Client Application Users,” on page 117
- Chapter 14, “Creating Cost Centers,” on page 121
- Chapter 15, “Billing for Purchases,” on page 125
- Chapter 16, “Changing the Status of a Purchase,” on page 127
- Chapter 17, “Managing the Maintenance and UsageServer Processes,” on page 129

4

Administration Basics

The following sections provide basic information that will help as you administer Novell® ZENworks® OnDemand Services™.

- ◆ “OnDemand Services Gadgets” on page 69
- ◆ “OnDemand Services Snap-Ins for ConsoleOne” on page 69
- ◆ “Novell eDirectory Management” on page 70
- ◆ “ODSAdmin Trustee Rights” on page 71

OnDemand Services Gadgets

Much of the OnDemand Services functionality is enabled through the OnDemand Services gadgets. When appropriate, the administration tasks in this documentation cover the enabling and configuring of the gadgets associated with the tasks. However, not all configuration settings are discussed in the administration tasks. You should review [Appendix B, “Gadget Configuration Settings,” on page 157](#) so that you are familiar with the functionality provided by the gadget configuration settings.

OnDemand Services Snap-Ins for ConsoleOne

You can install the OnDemand Services snap-ins to as many ConsoleOne® locations as you need.

- 1 Launch the OnDemand Services Installation program from the \ondemand directory on *Novell ZENworks OnDemand Services 2 CD*.

- 2 Follow the prompts until you reach the Installation Type dialog box.
- 3 Select Custom, then click Next.
- 4 Deselect all components except the ConsoleOne Snapins component.
- 5 Click Next, then follow the prompts to install the snap-ins.

Novell eDirectory Management

The OnDemand Services components perform tasks in Novell eDirectory™ to manage applications, packages, and users' access to packages. In general, if OnDemand Services configures a setting in eDirectory, you should let OnDemand Services manage that setting. Listed below are some things you shouldn't do in eDirectory because they adversely affect the way OnDemand Services operates.

Disassociating Desktop and Thin-Client Applications: When a user purchases a package that includes desktop or thin-client applications, OnDemand Services associates the user with the applications through the Application object's Association page. When the user's purchase expires, OnDemand Services disassociates the user from the applications.

You should not manually disassociate a user from a packaged desktop or thin-client application. Doing so does not set the "Expired" flag on the Purchase object, so the application continues to appear in the user's Launch Item gadget and also in the budget holder's User Administration gadget. The user can continue to launch the application at no charge and the budget holder doesn't know that the user has been disassociated with the application.

For more information about application packages, see [Chapter 5, "Creating Application Packages," on page 73](#).

Removing Users from iChain ACL Rules' Apply To Lists: When a user purchases a package that includes a Web application that is secured through Novell iChain™, OnDemand Services adds the user to the Apply To list for the Web application's ACL rule. You should not manually remove a user from the Apply To list. Doing so causes the user to be unable to launch the Web application, but the Web application still shows up in the user's Launch Item gadget.

For more information about OnDemand Services integration with iChain, see [Chapter 9, “Securing Web Applications With Novell iChain,” on page 99](#).

Removing Purchase Objects: When a user purchases a package, OnDemand Services creates a Purchase object for the user under the Package object. The Launch Item gadget uses this object to know which applications to display to the user. After the purchase has expired, OnDemand Services will mark the Expired flag on the Purchase object and revoke the user’s rights to the package’s applications. At that point, the package’s applications will no longer appear in the Launch Item gadget.

You should not manually remove the Purchase object. If you want to remove an expired Purchase object, make sure the Expired and Billed flags are marked. The Maintenance process will then remove the object (and any Usage objects associated with the Purchase object) the next time it processes the eDirectory tree.

For more information about Purchase objects, see [Chapter 16, “Changing the Status of a Purchase,” on page 127](#). For more information about the Maintenance process, see [Chapter 17, “Managing the Maintenance and UsageServer Processes,” on page 129](#).

ODSAdmin Trustee Rights

The OnDemand Services components authenticate to eDirectory through the OnDemand Administrator (ODSAdmin) object. The table below summarizes all trustee rights required by ODSAdmin.

During installation, ODSAdmin is assigned Supervisor rights to the company root container. As long as any of the other objects listed in the table reside beneath the company root container, ODSAdmin will have the required rights. Otherwise, you need to manually assign the rights.

Object	Trustee Rights	Explanation
Company root container	[Entry Rights] - Browse [All Attributes Rights] - Compare, Read	The company root container (as defined on the OnDemandService object) and all subcontainers.
User containers	[Entry Rights] - Browse, Create [All Attributes Rights] - Compare, Read, Write	All containers where User objects reside.

Object	Trustee Rights	Explanation
Catalog container	[Entry Rights] - Browse, Create [All Attributes Rights] - Read, Compare	The catalog root container (as defined on the OnDemandService object) and all subcontainers.
Application objects containers	[Entry Rights] - Browse [All Attributes Rights] - Compare, Read, Write	All containers where Application objects (ZfD and DeFrame) reside.
Commerce Item objects containers	[Entry Rights] - Browse [All Attributes Rights] - Compare, Read	All containers where Commerce Item (Web application) objects reside.
iChain ACL Rule objects	[Entry Rights] - Browse, Create [All Attributes Rights] - Compare, Read, Write	All iChain ACL rules that control access to OnDemand Services Web applications.
OnDemandService object	[Entry Rights] - Browse [All Attributes Rights] - Compare, Read	
Cost Center objects	[Entry Rights] - Browse [All Attributes Rights] - Compare, Read, Write	All cost center (Organizational Role) objects

5

Creating Application Packages

You make applications available to users by including them in a Novell® ZENworks® OnDemand Services™ package. After an application has been included in a package and you've enabled the package for purchase, users can gain access to the application by requesting the package or having their cost center's budget holder assign the package to them.

An OnDemand Services package can consist of a single application or a suite of applications. For example, you could create a package that consists of Microsoft Word. Or, you could create a package that includes all the Microsoft Office applications (Word, Excel, and so forth). Users who purchase a package gain access to all applications included in the package.

A package can include purchase and usage charges, or you can make the package available for free.

Refer to the following sections for information about packaging applications to make them available to users:

- ♦ “Preparing the Package’s Applications” on page 74
- ♦ “Creating the Package Object” on page 78
- ♦ “Adding Applications to the Package” on page 80
- ♦ “Establishing the Package’s Cost” on page 81
- ♦ “Enabling the Launch Item Gadget to Display Desktop Applications” on page 83

IMPORTANT: If a user purchases a package, but the package contents are not successfully delivered to the user, the user will still be charged for the purchase. Users will need to contact you to request that the purchase be removed.

Preparing the Package's Applications

Before you create the package, you need to make sure that the package's applications are ready. Refer to the following sections:

- ♦ “Desktop Applications” on page 74
- ♦ “Thin-Client Applications” on page 75
- ♦ “Web Applications” on page 75

Desktop Applications

OnDemand Services uses ZENworks for Desktops (ZfD) 3.2 to distribute desktop applications. Before you can add a desktop application to the package, you need to use ZfD to prepare the application for distribution. For information about using ZfD to distribute an application, see the ZfD 3.2 documentation on the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

As you prepare a desktop application that will be used with OnDemand Services, keep in mind the following requirements that are specific to OnDemand Services:

- ♦ When creating and configuring the Application object, make sure you use a UNC path (for example, \\server_hostname\vol1\snapshot\appname) in any locations where you are prompted for a path, and that you use the server's hostname in all locations where the server name is required.
- ♦ Do not use any of the following characters in the Application object's name:

" (double quote)	+ (plus)
= (equals)	\ (backslash)
< (open angle bracket)	; (semicolon)
> (close angle bracket)	/ (forwardslash)
, (comma)	# (pound sign)
- ♦ Do not use the Application object's Association page to associate the application with users. OnDemand Services will create this association when a user purchases a package that includes the application.

- ◆ If you plan to use a pricing scheme that requires the application's usage to be tracked (see **“Establishing the Package's Cost” on page 81**), turn on the Enable OnDemand Services to Track Usage of This Application option. This option is located on the Application object's OnDemand tab.
- ◆ If the application's usage will be tracked (as discussed in the previous bulleted item) and the application's executable is a stub program that simply launches the real executable (for example, write.exe launches wordpad.exe), make sure that you've entered the real executable filename in the Application object's Monitor Module Name field (Run Options tab > Environment page). Otherwise, OnDemand Services will only track usage of the stub program, which will result in a very brief, inaccurate usage tracking record.

Thin-Client Applications

OnDemand Services uses Novell DeFrame™ 2.01 to distribute thin-client (terminal server) applications. Before you can add a thin-client application to the package, you need to use DeFrame to prepare the application for distribution. For information, see the DeFrame 2.01 *Installation and Administration Guide* (deframe.pdf) on the root of the *Novell ZENworks OnDemand Services 2* CD.

Web Applications

To include a Web application in a package, you need to 1) make sure the application is located on a Web server that users can access and 2) create a Commerce Item object to define the application in Novell eDirectory™.

You can also include Web content, such as document, video, or audio files, in addition to Web applications. The same process is required if you want to include Web content.

To create a Commerce Item object in eDirectory:

- 1 In ConsoleOne®, right-click the container where you want to create the object > click New > click Object.

You should create the Commerce Item object in the company container (as defined in the OnDemandService object) or one of its subcontainers. This ensures that OnDemand Services has sufficient rights to the object.

If you create it outside of the company container or its subcontainers, you will need to make the OnDemand Administrator (ODSAdmin) object a trustee of the Commerce Item object. For details about the required trustee rights, see [Appendix , “ODSAdmin Trustee Rights,” on page 71.](#)

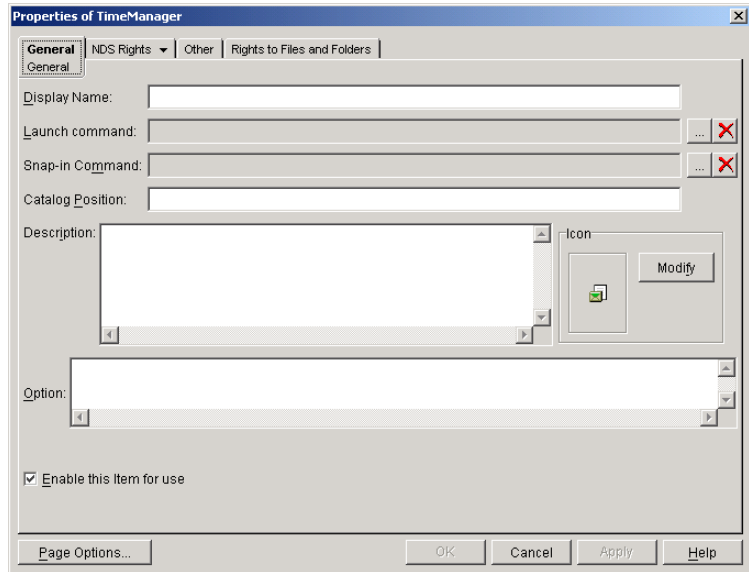
2 In the New Object list, select the commerceItem object, then click OK.

3 Provide the new item with a name.

The name you supply will be used for the object’s eDirectory name and for display in the OnDemand Services gadgets if you do not supply a Display Name later. Do not use any of the following characters in the name.


" (double quote)	+ (plus)
= (equals)	\ (backslash)
< (open angle bracket)	; (semicolon)
> (close angle bracket)	/ (forwardslash)
, (comma)	


4 Select the Define Additional Properties box, then click OK.



5 On the General tab, fill in the following properties:

Display Name: Enter the name of the item as you want it displayed to users. If you don't enter a display name, the Commerce Item object's name is used.

Launch Command: Click the Define Command button () to display the Launch Command dialog box. Enter the URL for the Web application or content, then click OK.

Snapin Command: This field applies only if you are using Novell iChain[®] to secure access to the Web application. If so, click the Define Command button () to display the Snapin Command dialog box. Browse for and select the iChain ACL rule that will provide access to the secured Web application, then click OK. For more information about iChain and iChain ACL rules, see [Chapter 9, "Securing Web Applications With Novell iChain,"](#) on page 99.

Catalog Position: You can use this field to define a folder structure to display the item in the Launch Item gadget. The syntax uses forward slashes, as in the following example:

/Software/Wordprocessors

Description: Enter a description to be displayed to the user.

Icon: If you want to change the display icon, click Modify. You can select an icon from an executable file or .ico file.

Option: This field lets you specify a URL to a Web site that describes or promotes the item. Use the following syntax:

```
<attribute name='brandingURL' value='<URL>' />
```

Replace <URL> with the appropriate URL. Make sure to include the protocol in the URL (for example, <http://www.novell.com>).

Enable this Item for Use: Select this option to make the item available to users. If you deselect this option, the item will not be available, even if a user has already purchased the package.

- 6 Click OK to save the information.

Creating the Package Object

The Package object is used to add package items and determine the package price. During installation, a Software container was created in the location you specified as the packages container. You need to create your Package objects in the Software container.

To create a Package object in eDirectory:

- 1 In ConsoleOne, right-click the a container in the OnDemand Services package catalog > click New > click Object.

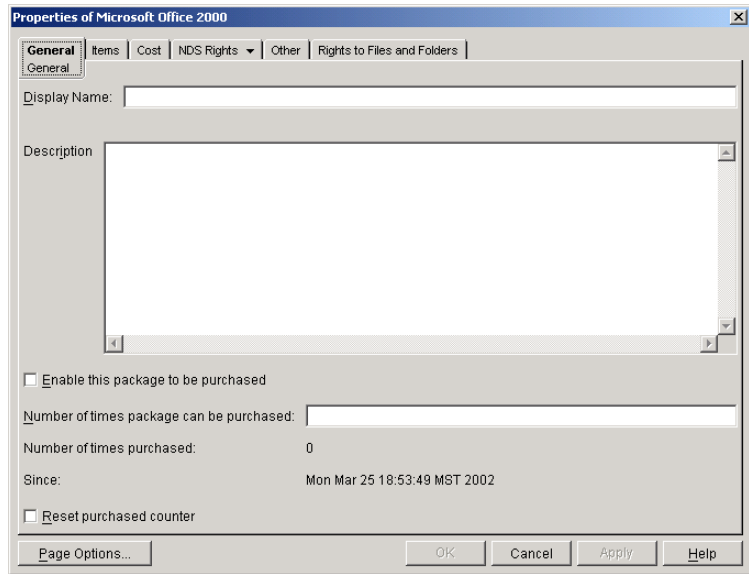
You can create the Package object anywhere in the catalog root container (as defined on the OnDemandService object) or its subcontainers. For information about how your package organization impacts users, see [“Organizing Packages and Applications” on page 107](#).

- 2 In the New Object list, select the commercePackage object, then click OK.
- 3 Provide the new package with a name.

The name you supply will be used for the object’s eDirectory name and for display in the OnDemand Services gadgets if you do not supply a Display Name later. Do not use any of the following characters in the name.

" (double quote)	+ (plus)
= (equals)	\ (backslash)
< (open angle bracket)	; (semicolon)
> (close angle bracket)	/ (forwardslash)
, (comma)	

4 Select the Define Additional Properties box, then click OK.



5 On the General tab, fill in the following properties:

Display Name: Enter the name of the package as you want it displayed to customers. If you don't enter a display name, the Package object's name is used.

Description: Enter a description to be displayed to the user.

Enable This Package to Be Purchased: Turn on this option when you are ready to allow users to purchase the package.

Number of Times Package Can Be Purchased: If you want to restrict the number of times the package can be purchased, enter the appropriate number. Leave the field empty to enable an unlimited number of purchases.

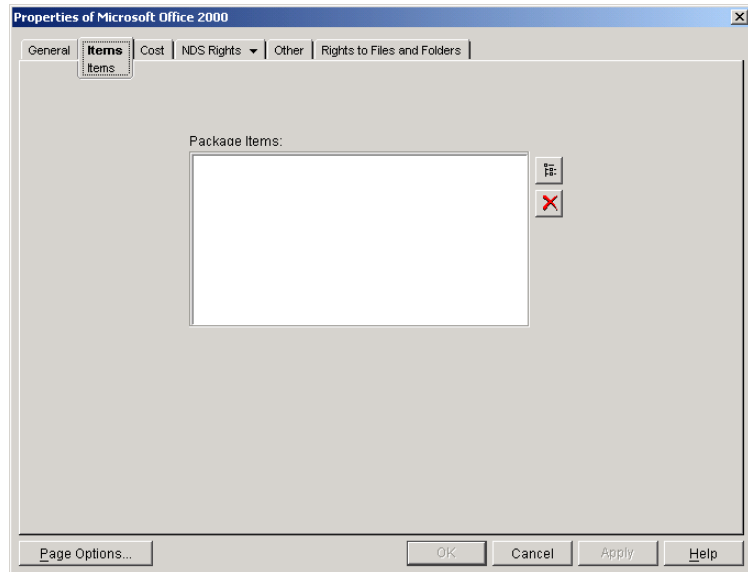
Reset Purchased Counter: This option applies only if you've set a limit on the number of times the package can be purchased.

Click this option, then click Apply to reset the purchase counter. This is useful if the purchase limit has been reached and you want to enable the package to be purchased again, or if you want to start tracking the number of purchases from the current day and time.

- 6 Leave the Package object's dialog box open, then continue with the next section, [Adding Applications to the Package](#).

Adding Applications to the Package

- 1 Click the Items tab.



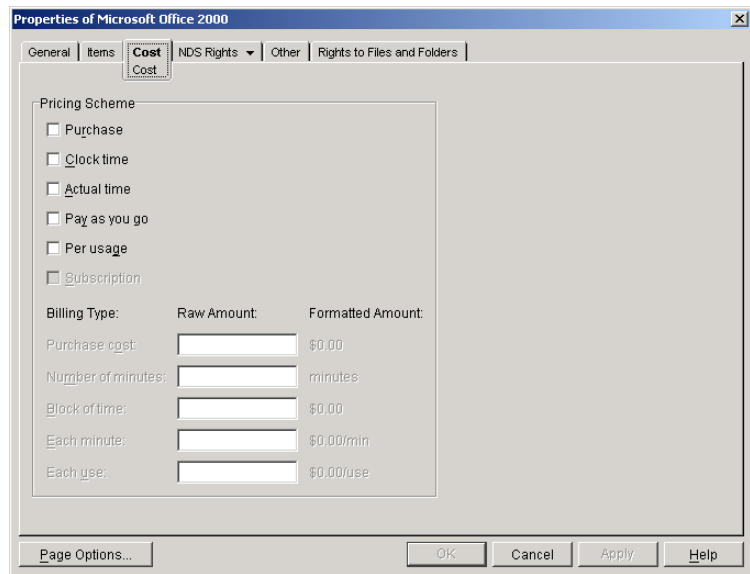
- 2 Click the browse button next to the Package Items list, then browse to and select the Application object or Commerce Item object to add to the package.

IMPORTANT: If you add the same application to two different packages and a user purchases both packages, only one application icon will be displayed in the Launch Item gadget. In addition, any charges associated with using the application will accrue to the most recent purchase. After that purchase expires, the charges will revert to the previous purchase.

- 3 Leave the Package object's dialog box open, then continue with the next section, **Establishing the Package's Cost**.

Establishing the Package's Cost

- 1 Click the Cost tab.



- 2 Select one or more of the following pricing schemes:
 - ♦ **Purchase:** The user pays a one-time fee for the package. The user can use the package whenever necessary.
 - ♦ **Clock Time:** The user purchases a set amount of time to use the package. The time begins when the user purchases the package, runs continuously, and expires after the set amount of time.

- ◆ **Actual Time:** The user purchases a set amount of time to use the package. Only the actual time spent using the item is recorded. If the user purchases 5 hours, he or she can use 1 hour today, 3 hours tomorrow, and 1 hour next week.
- ◆ **Pay-As-You-Go:** The user pays by the minute to use the package. The time begins when the user launches a package item and ends when the item is exited.
- ◆ **Per Usage:** The user pays each time he or she uses the package.
- ◆ **Subscription:** The user pays a set amount for a subscription to the package. You determine the subscription period (7 days, 14 days, 30 days, etc.). At the end of the subscription period, the user's subscription is automatically renewed. By default, users can cancel a subscription. If you want to disable a user's ability to cancel a subscription, see [“Assigning Users” on page 45](#).

You can use a combination of these pricing schemes. For example, you could have a one-time purchase fee (Purchase pricing scheme) for a package, charge a fee for every time one of the package's applications is used (Per Usage pricing scheme), and also charge a per-minute fee for using the application. (Pay-As-You-Go pricing scheme). For information about which pricing schemes can be used together, click the Help button on the Cost page.

The Pay-As-You-Go and Actual Time schemes should not be used with Web applications because the OnDemand Services UsageServer is unable to track the actual time Web applications are in use. For details, see [“Usage Tracking Support for Each Application Type” on page 89](#).

IMPORTANT: All schemes, other than the Purchase scheme, require OnDemand Services to track usage of the package's items. If you select a usage-based scheme, you need to make sure that usage tracking is enabled and configured properly. For information, see [Chapter 7, “Tracking Application Usage,” on page 89](#).

- 3** Fill in the cost information for the selected schemes.
- 4** Click OK to save the information.

Enabling the Launch Item Gadget to Display Desktop Applications

This section applies only if you are including desktop (ZENworks for Desktops) applications in any of your packages.

By default, the Launch Item gadget is configured to not display desktop applications so as not to cause problems with users who do not have the Novell Client™ and Novell Application Launcher™ installed on their workstations. To change this configuration setting:

- 1** Log in to Novell Portal Services (NPS) as a portal administrator.
- 2** Click Administer the Portal.
- 3** Click Pages.
- 4** Select the page that contains the Launch Item gadget, then click Edit.

The gadget list includes all the gadgets that are included on the page. The Launch Item gadget should be in this list. However, it will be listed by its display name (for example, Launch Applications), not its gadget name (OD_LaunchItemGadget).

- 5** In the list of gadgets, select the gadget, then click Edit.
- 6** Change the Display Workstation Applications setting from FALSE to TRUE.
- 7** Click Continue, click Save, then click OK.

6

Distributing Free Applications

You can distribute desktop applications and thin-client applications by associating Application objects with users. Because the applications are distributed outside of a Novell® ZENworks® OnDemand Services™ package, the applications are free and you will not be able to track usage. If you want to track the usage of a free application, you need to include it in a package and deselect all pricing schemes. For information, see [Chapter 5, “Creating Application Packages,”](#) on page 73.

The following sections provide instructions to help you distribute free applications:

- ♦ [“Configuring an Application for Free Distribution through User Associations”](#) on page 85
- ♦ [“Assigning the OnDemand Administrator \(ODSAdmin\) Rights to the Application Object”](#) on page 86
- ♦ [“Integrating with Novell Application Launcher”](#) on page 87

Configuring an Application for Free Distribution through User Associations

- 1** In ConsoleOne®, right-click the Application object for the application you want to distribute freely > click Properties.
- 2** Click the Associations tab.
- 3** Click Add to add the users you want.

You can add users through their User objects, through Group objects, or through container objects.

- 4 Click the OnDemand tab, then make sure the Enable OnDemand Services to Track Usage of This Application option is disabled.

IMPORTANT: This option must be disabled. Otherwise, when launching the application users will receive an error message stating that there are no active purchases for the application and the application will fail to launch.

- 5 Click OK to save the associations.

Assigning the OnDemand Administrator (ODSAdmin) Rights to the Application Object

This section applies only if the Application object resides outside of the company root container (as defined by the Company Location setting in the OnDemandService object).

The OnDemand Administrator (ODSAdmin) object has rights to all Application objects and Package objects in the company root container and its subcontainers. If you are distributing an application whose Application object resides in a container outside of the company root, you need to manually assign the ODSAdmin object rights to the Application object or its container.

- 1 Right-click the Application object or its container object > click Trustees of This Object.
- 2 Click Add Trustee, browse for and select the ODSAdmin User object, then click OK to display the Rights Assigned to Selected Objects dialog box.
- 3 Leave the [Entry Rights] set to the defaults.
- 4 Select [All Attributes Rights], then click Write (so that Compare, Read, and Write are all selected).
- 5 Click OK to add the ODSAdmin user to the list of trustees.
- 6 Click OK to save your changes.

Integrating with Novell Application Launcher

If a user runs Novell Application Launcher™ (ZENworks for Desktops), the desktop and thin-client applications will be displayed not only in the OnDemand Services Launch Item gadget but also in Application Launcher.

Users can launch the desktop applications from Application Launcher with no problems. However, before a thin-client application can be launched, the Novell DeFrame™ client must be installed. Otherwise, the user will receive the following error:

```
Could not launch <appname> using  
c:\winnt\system32\dappx\dappx.exe...
```

To install the DeFrame client to a workstation you can do either of the following:

- ♦ Run the DeFrame Workstation Setup program (setup.exe). The Setup program is located in the deframe\wksetup directory on the *Novell ZENworks OnDemand Services 2* CD.
- ♦ Use ZENworks for Desktops Application Management to distribute the DeFrame client to workstations. To do so, you need to use the snAppShot™ utility to record the DeFrame client installation on a clean workstation, use the resulting .aot file to create an Application object, configure the Application object so that Application Launcher will automatically run it when the user logs in, and associate the Application object with the users you want it distributed to. For detailed instructions, see Application Management in the [ZENworks for Desktops 3.2 Administration guide \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

Novell Application Launcher and Usage-Based Applications

Thin-client applications that have been enabled for usage tracking cannot be launched from Novell Application Launcher. Usage tracking of thin-client applications requires them to be launched from the OnDemand Services Launch Item gadget.

Desktop applications that have been enabled for usage tracking can be launched from Novell Application Launcher, but only after having been launched from the OnDemand Services Launch Item gadget.

7

Tracking Application Usage

Novell® ZENworks® OnDemand Services™ enables you to track the usage of applications. Usage tracking is required if you want to implement package pricing schemes that are based on the number of times the package is used or the number of minutes it is used. If you don't want to implement usage-based pricing schemes for your packages, usage information can still help you decide if you need to increase or decrease the resources (hardware, Help Desk personnel, etc.) supporting applications.

- ♦ [“Usage Tracking Support for Each Application Type” on page 89](#)
- ♦ [“Enabling Usage Tracking of an Application” on page 90](#)
- ♦ [“Ensuring Accurate Usage Tracking for Thin-Client Applications” on page 91](#)
- ♦ [“Starting the UsageServer and Maintenance Processes” on page 92](#)

Usage Tracking Support for Each Application Type

OnDemand Services provides full usage tracking support for thin-client and desktop applications. You can track both the number of times (per-use) and the number of minutes (per-minute) a thin-client or desktop application is used.

OnDemand Services provides partial usage tracking support for Web applications. You can track per-use but not per-minute usage.

Because per-minute usage is not supported for Web applications, you should not include Web applications in any package whose usage-based pricing scheme is reliant on tracking time that the application is open. The Actual Time and Pay-As-You-Go pricing schemes fall into this category. The

following table summarizes the application types and the package pricing schemes you can use with packages that contain those application types.

	Web App	Thin-Client App	Desktop App
Clock Time	Yes	Yes	Yes
Actual Time	No*	Yes	Yes
Pay-As-You-Go	No*	Yes	Yes
Per Usage	Yes	Yes	Yes
Subscription	Yes	Yes	Yes

* You can actually include Web applications in packages that use these pricing schemes. With Actual Time, however, using the package's Web applications will not increment the time accrued; only thin-client and desktop applications included in the package will increment the time. With Pay-As-You-Go, use of the package's Web applications will be free; only thin-client and desktop applications included in the package will be charged a per-minute usage fee.

Enabling Usage Tracking of an Application

By default, desktop and thin-client applications are not enabled for usage tracking, but Web applications are. To enable a desktop or thin-client application:

- 1 In ConsoleOne[®], right-click the Application object.
- 2 Click the OnDemand tab to display the Usage Tracking page.
- 3 Click the Enable OnDemand Services to Track Usage of This Application check box to turn on the option.

IMPORTANT: Thin-client applications that have been enabled for usage tracking cannot be launched from Novell Application Launcher[™] (ZENworks for Desktops). Desktop applications that have been enabled for usage tracking can be launched from Novell Application Launcher, but only after having been launched from the OnDemand Services Launch Item gadget.

4 If you want OnDemand Services to terminate the application if its usage expires while the user is running it, complete the following substeps. Otherwise, the user will be able to use the application until he or she exits.

4a Click the Availability tab, then click Termination.

4b Select the desired termination behavior, then click OK. .

For a description of each behavior, click the Help button on the Termination page

5 Click OK to save your changes.

Ensuring Accurate Usage Tracking for Thin-Client Applications

To properly terminate usage tracking of a thin-client application, users need to use the application's Exit feature (for example, File menu > Exit) rather than clicking the Close button in the upper-right corner of the session window. Using Exit correctly exits the application, logs the user out of the session, and terminates usage tracking.

If, rather than using the application's Exit feature, the user closes the terminal server session (by clicking the Close button in the upper-right corner of the session window), the user is disconnected from the session but not logged out. The session remains open on the terminal server, and the user can reconnect to the same session by launching the application again. This means that the application was never really exited, so usage tracking continues.

To ensure that disconnected sessions are closed out, you can configure the terminal server to end disconnected sessions after a certain period of time. For instructions, see your terminal server documentation.

If you are using ZENworks for Desktops, you can enable a Terminal Server policy (on a User package) and apply the User package to the appropriate users. The Terminal Server policy should configure a timeout interval for disconnected sessions (Disconnection Timeout setting) and reset the connection on timed-out sessions (Reset Broken or Timed-Out Sessions setting). For more information, see the [ZENworks for Desktops 3.2 documentation](http://www.novell.com/documentation) (<http://www.novell.com/documentation>).

Starting the UsageServer and Maintenance Processes

For usage tracking to occur, you need to make sure the UsageServer process is running. If the UsageServer process is not installed, or for information about configuring the process, refer to [“Managing the Maintenance and UsageServer Processes” on page 129](#).

In addition, the Maintenance process also needs to be running. If the UsageServer becomes unavailable, the Maintenance process closes the usage records for any applications that were being tracked. Even if the users continue to use the applications, they are only charged for the time up to the point that the UsageServer became unavailable. This ensures that they will not be charged for time they did not use.

- ♦ [“Starting the UsageServer Process” on page 92](#)
- ♦ [“Starting the Maintenance Process” on page 93](#)

Starting the UsageServer Process

To start the UsageServer process:

1 At the NetWare[®] server console, enter `usageserver.ncf`.

or

At the Windows server command prompt, change to the directory where the Usage Server was installed (by default, `\ondemand\bin`), then enter `usageserver.bat`.

2 Enter the password for the Admin or Admin-equivalent user that the Usage Server will use to authenticate to Novell eDirectory™ through the LDAP server.

If you can't remember which user is being used for authentication, the user's context is stored in the `commerceserver.cfg` file, located in the `\ondemand` directory.

You can install additional UsageServer processes or change the default configuration for this process. For information, see [Chapter 17, “Managing the Maintenance and UsageServer Processes,” on page 129](#).

Starting the Maintenance Process

1 At the NetWare server console, enter `commercemaintenance.ncf`.

or

At the Windows server command prompt, change to the directory where the Maintenance process was installed (by default, `\ondemand\bin`), then enter `commercemaintenance.bat`.

2 Enter the password for the Admin or Admin-equivalent user that the Usage Server will use to authenticate to eDirectory through the LDAP server.

If you can't remember which user is being used for authentication, the user's context is stored in the `commerceserver.cfg` file, located in the `\ondemand` directory.

You can install additional Maintenance processes or change the default configuration for this process. For information, see [Chapter 17, "Managing the Maintenance and UsageServer Processes,"](#) on page 129.

8

Limiting Package Availability

By default, Novell® ZENworks® OnDemand Services™ is configured to allow all packages to be available to all users. You can, however, configure OnDemand Services so that packages are available only to the users who have been granted rights specifically to the packages.

The OnDemand Administrator (ODSAdmin) User object is a trustee of the OnDemand Services catalog root container. The OnDemand Services Package Request gadget is configured to access the catalog as the ODSAdmin user, which means that it has rights to see all packages contained within the catalog. To limit package availability, you configure the Package Request gadget to receive its catalog rights through the logged-in user rather than the ODSAdmin user. The user will then see only the packages that you specifically give him or her rights to.

The following sections provide instructions for completing the tasks discussed above:

- ♦ [“Configuring the Package Request Gadget to Limit Package Availability” on page 95](#)
- ♦ [“Giving Users Trustee Rights to Packages” on page 96](#)

Configuring the Package Request Gadget to Limit Package Availability

The Package Request gadget’s Limit Package Availability configuration setting determines whether the gadget displays all packages or only the packages the user has been specifically granted rights to.

To configure the gadget to limit availability:

- 1** Log in to Novell Portal Services (NPS) as a portal administrator.
- 2** Click Administer the Portal.
- 3** Click Pages.
- 4** Select the page that contains the Package Request gadget, then click Edit.

The gadget list includes all the gadgets that are included on the page. The Package Request gadget should be in this list. However, it will be listed by its display name (for example, Request Applications), not its gadget name (OD_PackageRequestGadget).
- 5** In the list of gadgets, select the gadget, then click Edit.
- 6** Change the Limit Package Availability setting from FALSE to TRUE.
- 7** Click Continue, click Save, then click OK.

Giving Users Trustee Rights to Packages

With the Package Request gadget configured to limit package availability, users will only have rights to a package if you make them trustees of the Package object and the package's container object. If you have many users who you want to make the package available to, you can add those users to a group and then make the group a trustee of a Package object and its container object. The following sections provide instructions:

- ♦ [“Making Users Trustees of a Package Object” on page 96](#)
- ♦ [“Making Users Trustees of a Container Object” on page 97](#)

Making Users Trustees of a Package Object

- 1** In ConsoleOne[®], right-click the Package object > click Trustees of This Object to display the Trustees of This Object page.
- 2** Click Add Trustee, browse for and select the user or group who you want the package to be available to, then click OK to display the Rights Assigned to Selected Objects dialog box.

To make all objects in a container trustees of the Package object, you can select the container object rather than the individual User and Group objects.

- 3** Click OK to accept the default rights and add the user to the list of trustees.
- 4** Repeat **Step 2** and **Step 3** to assign additional users as trustees.
- 5** When finished adding users, click OK to save your changes.
- 6** Repeat **Step 1** through **Step 5** to assign users rights to another package.

Making Users Trustees of a Container Object

Regardless of the trustee rights that a user has to a Package object, if he or she does not have trustee rights to the package's container object, the package will not be available.

You need to assign rights beginning with the OnDemand Services catalog root container (as specified on the OnDemandService object) down to the package's container. For example, if the catalog root container is PACKAGES and the PACKAGES\APPLICATIONS container is where the package resides that you're making available, you need to assign trustee rights to the PACKAGES container and the APPLICATIONS container.

NOTE: If desired, you can simply assign rights to the catalog root container (PACKAGES, in the above example) and mark the rights as inheritable. However, this would cause the user to see all containers under the catalog root, even if a container does not have any applications that were made available to user.

To make a user a trustee of a container:

- 1** In ConsoleOne, right-click the container object > click Trustees of This Object.
- 2** Click Add Trustee, browse for and select the user or group, then click OK to display the Rights Assigned to Selected Objects dialog box.

To make all objects in a container trustees of the Package object, you can select the container object rather than the individual User and Group objects.

- 3** In the Property list, click [Entry Rights], then deselect Browse (make sure all rights are deselected).

- 4** In the Property list, click [All Attributes Rights], then deselect Compare and Read (make sure all rights are deselected).
- 5** Click Add Property, select OU, then click OK to add it to the Property list. Leave the rights set to the default (Compare and Read).
- 6** Click OK to add the user to the list of trustees.
- 7** Repeat **Step 2** through **Step 6** to make additional users trustees of the container.
- 8** When finished adding users, click OK to save your changes.
- 9** Repeat **Step 1** through **Step 8** to make users trustees of another container.

9

Securing Web Applications With Novell iChain

You can use Novell® iChain® 2.x to secure the Web applications you are provisioning through Novell ZENworks® OnDemand Services™. Doing so ensures that a user will only have access to a Web application if he or she has purchased (or been assigned access to) the Web application's package.

To integrate OnDemand Services with iChain, you set up your Web applications as protected resources with an access level of Secure. You then create access control list (ACL) rules that allow access to the Web applications in these secured areas. When a user purchases a package that includes a Web application, OnDemand Services adds the user to the membership list of the ACL rule associated with the Web application. This gives the user access to the Web application until the purchase expires, at which time OnDemand Services removes the user from the rule's membership list.

Complete the tasks in the following sections to secure your Web applications through iChain:

- ◆ “Setting Up Protected Resources” on page 100
- ◆ “Creating ACL Rules” on page 101
- ◆ “Configuring iChain Server ACL List Refresh” on page 102
- ◆ “Making the ODSAdmin Object a Trustee of ACL Rules” on page 104
- ◆ “Associating ACL Rules with Web Applications” on page 105

For additional information about iChain, see the Novell iChain 2.0 documentation at the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

Setting Up Protected Resources

If you have not already set up your Web applications as Secure access protected resources, you need to do so. In general, you should define the Web application folders, not the application files themselves, as protected resources. For example:

- ♦ If you have a single Web application folder structure (for example, the Web Application Server's webapp folder and subfolders), you can define that single Web application folder as a protected resource. Its files and subfolders become secured and are accessible only to users included in the membership lists of ACL rules that allow access.
- ♦ If your Web applications are located in different folders and do not have a common parent folder (or don't have a common parent folder that you want to define as a protected resource with Secure access level), you can define each Web application's folder as a protected resource. This accomplishes the same purpose as setting up a single protected resource; it simply involves a little more work for you.

Protected resources are configured through the iChain Service object (ISO) in Novell eDirectory™. To define a protected resource.

- 1** In ConsoleOne®, right-click the iChain Service object > click Properties to display the property pages.
- 2** Click the Protected Resources tab.
- 3** On the Protected Resource page, click the Add button to display the Add New Protected Resource dialog box.
- 4** Fill in the following fields:

Resource Name: Enter a name to identify the protected resource. The name differentiates the resources in the Protected Resources list and must be unique within the list. You can, if desired, enter the resource's URL as its name.

URL Prefix: Enter the resource's URL. For example,

`www.novell.com/webapps/*`

would define the webapps directory and subdirectories as a protected resource. On the other hand,

`www.novell.com/webapps/index.html`

would define only the `index.html` file as a protected resource.

Access: Select Secure so that access will be controlled through ACL rules.

Pass Parameters in the Query String: Leave this option selected.

Pass Parameters as Header Variables: Leave this option unselected.

- 5** Click OK to add the resource to the Protected Resource list.
- 6** Repeat **Step 3** through **Step 5** to set up additional protected resources.
- 7** When you've finished setting up protected resources, click OK to save your changes.

Creating ACL Rules

When a Web application is part of a protected resource, it is accessible only through an ACL rule that has been configured to allow access to the Web application.

You can configure an ACL rule to provide access to multiple Web applications. Generally, however, you should create separate ACL rules for each Web application. If you inadvertently use the same ACL rule for Web applications in different packages, when a user purchases one of the packages he or she will also gain access to the Web applications in the other package. Although the user couldn't launch the unpurchased Web applications through OnDemand Services (because they won't appear in the Launch Item gadget), he or she could manipulate a purchased Web application's URL and possibly gain access to the unpurchased Web application.

To create an ACL rule:

- 1** In ConsoleOne, if you are using iChain 2.0, right-click the container where you want to create the object, click New, click Object to display the New Object dialog box, select iChain Access Control Rule, then click OK to display the New iChain Access Control Rule dialog box.

or

If you are using iChain 2.1, right-click the container where you want to create the object, click New, click iChain Object, select iChain Access

Control Rule, then click OK to display the New iChain Access Control Rule dialog box.

- 2** Enter a name for the ACL Rule object, select Define Additional Properties, then click OK.
- 3** Click the Access Control tab.
- 4** Click the Add button on the right side of the Allowed URLs list to display the Add New Resource dialog box.
- 5** Fill in the following fields:

Resource Name: Browse for and select the protected resource that the Web application is part of. The protected resource's URL forms the basis of the Web application's URL.

URL Postfix: Enter the information that, when appended to the protected resource's URL, completes the Web application's URL.

An asterisk (*) as the last character provides access to the folder content and all subfolders. A question mark (?) as the last character provides access to the folder contents but not the subfolders.

For example, if the protected resource's URL were `www.mycompany.com/webapps`, entering `/timemanager/?` would give the user access to all files in the `www.mycompany.com/webapps/timemanager` folder. An asterisk used in place of the question mark would grant access to that folder and its subfolders.

- 6** Click OK to add the URL to the Allowed URLs list.
- 7** Click OK to save the ACL rule.
- 8** Repeat **Step 1** through **Step 7** to create ACL rules that allow access to other Web applications.

Configuring iChain Server ACL List Refresh

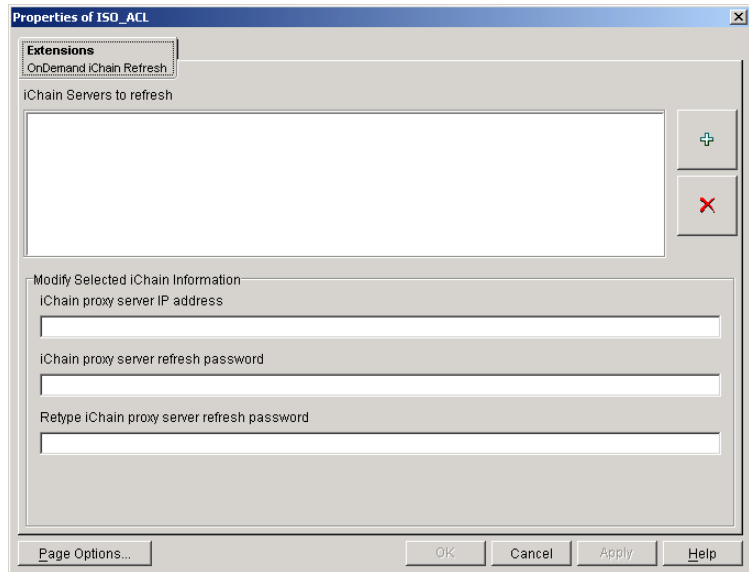
Whenever the OnDemand Services adds a user to an ACL rule's membership list or removes a user from the list, each iChain server where the ACL rule is being applied needs to be refreshed immediately so that the membership list changes will be enforced. Otherwise, a user who has just purchased a package with a Web application won't be able to run the application until the iChain server refreshes its ACL rules. Likewise, a user whose package has expired

will be able to continue to use the package's Web applications until the iChain server refreshes.

OnDemand Services can immediately refresh the ACL rules of one or more iChain servers. For each ACL rule you create, you define the iChain servers you want OnDemand Services to refresh. When a user is added or removed from the ACL rule's membership list, OnDemand Services refreshes the specified iChain servers.

To define the iChain servers that OnDemand Services will refresh when an ACL rule's membership list is modified:

- 1** Right-click the ACL Rule object, then click Extensions of This Object.
- 2** Click Add Extension, select the commerceProxySecretStore auxiliary class extension, then click OK to display the OnDemand iChain Refresh page.



- 3** Click the Add (+ sign) button to add an address to the iChain Servers to Refresh list.
- 4** In the list, select the address that was just added to display it in the iChain Proxy Server Address field.

5 Modify the following fields:

iChain Proxy Server Address: Change the sample address to the iChain server's address. Press Enter after you've modified the address so that the list will reflect the change. If you don't press Enter, the change will not be made.

iChain Proxy Server Refresh Password: Type the existing password required to refresh ACL rules. This is the same password used by the iChain Config user.

Retype iChain Proxy Server Refresh Password: Confirm the password by retyping it.

6 Click Apply to save the changes.

7 Repeat **Step 3** through **Step 6** to define additional iChain server's whose ACL rules list should be refreshed when OnDemand Services updates the ACL rule.

8 When finished defining iChain servers, click OK, then click Close to close the extensions dialog box.

Making the ODSAdmin Object a Trustee of ACL Rules


The OnDemand Administrator (ODSAdmin) object must be a trustee of the ACL rules so that OnDemand Services can read and update the rules.

- 1** Right-click the ACL Rule object, then click Trustees of This Object.
- 2** Click Add Trustee, browse for and select the ODSAdmin object, then click OK to display the Rights Assigned to Selected Objects dialog box.
- 3** Leave [Entry Rights] set to the defaults.
- 4** Select [All Attributes Rights], then click Write. Compare, Read, and Write should all be selected.
- 5** Click OK to add the ODSAdmin user to the list of trustees.
- 6** Click OK to close the Trustees of This Object page.

Associating ACL Rules with Web Applications

For OnDemand Services to add a user to the correct ACL rule's membership list when the user purchases a package that contains a Web application, you need to add the ACL rule to the Snapin Command field in the Web Application's Commerce Item object.

To associate an ACL rule with a Web application:

- 1** Right-click the Web application's Commerce Item object > click properties.
- 2** In the Snapin Command field, click the Define Command button () to display the Snapin Command dialog box.
- 3** Browse for and select the ACL rule that allows access to the Web application, then click OK to close the Snapin Command dialog box.
- 4** Click OK to save your changes.

10

Organizing Packages and Applications

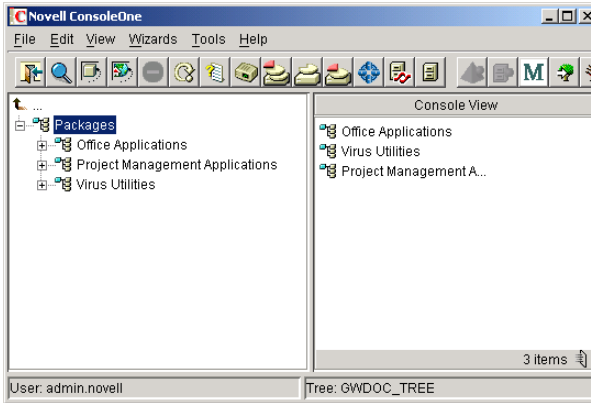
You can organize how packages are displayed in the Package Request gadget and how applications are displayed in the Launch Item gadget.

- ◆ [“Organizing Packages in the Package Request Gadget” on page 107](#)
- ◆ [“Organizing Applications in the Launch Item Gadget” on page 108](#)

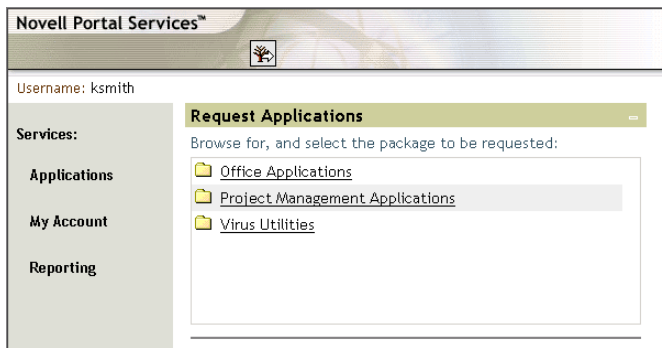
Organizing Packages in the Package Request Gadget

The Package Request gadget displays packages exactly as they are organized in the package catalog in Novell® eDirectory™. The package catalog is the container structure where packages need to reside in order to be available to users. The location of the package catalog is determined by the catalog root container, which is specified in the OnDemandService object.

All packages can reside in the catalog root container, or you can organize your package catalog by creating containers under the catalog root. In the following example, Packages is the catalog root container and Office Applications, Virus Utilities, and Project Management Applications are three subcontainers.



The following example shows the resulting folder structure for the Package Request gadget. Notice that the name of the catalog root container (Packages, in the above example) is not displayed to users.



Because the Package Request gadget displays the catalog exactly as it is organized in eDirectory, you will want to organize the catalog in a manner that makes sense to both you and your users.

Organizing Applications in the Launch Item Gadget

The Launch Item gadget displays desktop and thin-client applications in the locations designated in the Folders list of their Application objects (Application objects > Identification tab > Folders page). For information about using the Folders list, see Application Management in the ZENworks

for Desktops 3.2 *Administration Guide* at the [Novell Documentation Web site](http://www.novell.com/documentation) (<http://www.novell.com/documentation>).

The Launch Item gadget displays Web applications in the location specified in the Catalog Position field on a Web Applications Commerce Item object (Commerce Item object > General tab > General page). For information about using the Catalog Position field, see “**Web Applications**” on page 75.

11

Customizing Notification Messages

Novell® ZENworks® OnDemand Services™ creates the following e-mail notifications:

- ♦ **Package Request:** Sent to a cost center's budget holder to inform him or her of a user's request to purchase a package.
- ♦ **Request Response:** Sent to a user in response to his or her package request. Indicates whether the budget holder approved or denied the request.
- ♦ **Purchase Information:** Sent to a user to inform him or her that he or she has been assigned a package.

These e-mail notifications include text in the message subject and the message body that you can customize to meet your own needs.

- ♦ [“Modifying the Message Subject” on page 112](#)
- ♦ [“Modifying the Message Body” on page 112](#)

IMPORTANT: You must enable users to receive notification messages. This is done through the Receive Email Notifications setting on a User object (User object > OnDemand tab > Purchases page). For more detailed instructions, see [“Assigning Users to the Cost Center” on page 123](#). In addition, the E-Mail Address setting for each user and budget holder needs to be defined on their User object (User object > General tab > Identification page).

Modifying the Message Subject

The Subject line for all three notification messages must be the same. The default Subject line text is ApprovalFlow Notification.

To modify the Subject line:

- 1** In ConsoleOne[®], right-click the OnDemandService object > click Properties.
- 2** In the Mail Subject field, type the text you want to use for the Subject line.
- 3** Click OK to save your changes.

Modifying the Message Body

The CommerceMailMsg.properties file contains the body text for each notification message. The file is included in the CommerceLdapCmd.jar, located in the Web Application Server's webapps\nps\web-inf\lib directory.

To modify the message text:

- 1** Extract the CommerceMailMsg.properties file from the CommerceLdapCmd.jar file.

or

If you are using a zip utility such as WinZip that lets you edit a file without first extracting the file, open the CommerceLdapCmd.jar file in that utility.

- 2** Open the CommerceMailMsg.properties file in a text editor.

The text for the notification messages is found in the following three sections:

```
## E-mail: Purchase Message  
## E-mail: Request Approval Message  
## E-mail: Response to Request Message
```

You can include any text you want by entering it. You can add comments to the file by entering text preceded by a pound sign (#).

The predefined arguments you can use are listed at the beginning of the file. For example, you can use argument 0 to include the current date and time in the text.

- 3** Edit the message text and save the changes.
- 4** If you extracted the CommerceMailMsg.properties file from the .jar file to edit it, add the edited file back into the .jar file.

12 Supporting Desktop Application Users

For a Novell® ZENworks® OnDemand Services™ user to run desktop applications (Novell ZENworks for Desktops applications), the user must:

- ♦ Have the Novell Client™ installed on his or her computer. The Novell Client can be downloaded from the [Novell Downloads site \(http://download.novell.com\)](http://download.novell.com).
- ♦ Be logged in to Windows if using Windows 95/98.
- ♦ Install the Web browser plug-in that supports launching of desktop applications from the OnDemand Services Launch Item gadget. OnDemand Services automatically installs the plug-in, called `nalexec.cab`, the first time the user launches a desktop application. After the plug-in is installed, it uses Novell Application Launcher™ to launch the application. If Application Launcher does not reside on the user's workstation, the plug-in attempts to install it from the ZENworks server defined in the Launch Item gadget's configuration settings (see **“Launch Item Gadget (OD_LaunchItemGadget)” on page 160**). If the plug-in installs Application Launcher, users might be prompted to restart the computer. They should do this before launching an application.

On Windows NT/2000/XP, the user must be logged in as a member of the Administrators group for OnDemand Services to be able to install the plug-in (and Application Launcher if necessary). Logging in as an Administrator is only required the first time; users can log in through a non-Administrator Windows account after the plug-in has been successfully installed.

On Windows 98, if the ZENworks server from which the plug-in will install Application Launcher is a Windows 2000 server, the Windows 98 user must have a user account on the Windows 2000 server. Otherwise, the plug-in will not be able to install the Application Launcher files.

13

Supporting Thin-Client Application Users

The following sections provide information to help you support Novell® ZENworks® OnDemand Services™ users who are running thin-client applications (Novell DeFrame™ applications).

- ♦ “Installing the DeFrame, ICA, and RDP Web Clients” on page 117
- ♦ “Updating the ICA Web Client” on page 118
- ♦ “Installing the DeFrame Client” on page 119
- ♦ “Issues for Users to Be Aware Of” on page 120

Installing the DeFrame, ICA, and RDP Web Clients

In order for a user to launch thin-client applications from the Launch Item gadget, the DeFrame, ICA, and RDP Web clients must be installed on the user’s workstation.

The Web clients, packaged as .cab files, are installed the first time the user launches a thin-client application through the Launch Item gadget. The following .cab files are used:

- ♦ **dappx.cab:** Contains the DeFrame Web client. The Launch Item gadget will install this client only if a user launches a thin-client application and the client is not already installed on the user’s workstation.
- ♦ **wficat.cab:** Contains the ICA Web client. The .cab file contains the 6.30.1050.0 version of the ICA Web client that replaces all versions prior

to May 2002. The Launch Item gadget installs the ICA Web client the first time a user launches an application configured for an ICA session.

- ♦ **nmstsc.cab:** Contains the RDP Web client. The .cab file contains the same traditional RDP Web client that shipped with the initial OnDemand Services release. The RDP Terminal Services Advanced Client (TSAC) is not yet supported. The Launch Item gadget installs the RDP Web client the first time a user launches an application configured for an RDP session.

For the Web clients to be installed on a Windows 95/98 workstation, the user must be logged into Windows on the workstation.

For the Web clients to be installed on a Windows NT/2000/XP workstation, the user must be logged in as a member of the Administrators or Power Users group. Logging in as Administrator or a Power User is only required the first time; users can log in through a non-Administrator account thereafter.

Updating the ICA Web Client

You can update the ICA Web client whenever Citrix* releases a new version. This is not supported for the RDP Web client.

- 1** Download the new ICA Web client .cab file from the [Citrix ICA client download site \(http://www.citrix.com/download/ica_clients.asp\)](http://www.citrix.com/download/ica_clients.asp).
- 2** Copy the file to the Launch Item gadget's bin directory. The full path is:

```
tomcat\webapps\nps\portal\gadgets\com.novell.ondemand.gadgets.LaunchItemGadget\bin
```
- 3** Change the ICAVersion property in main.xml to the new version number:
 - 3a** Open the file in an editor. The file is located in the following directory:

```
tomcat\webapps\nps\portal\gadgets\com.novell.ondemand.gadgets.LaunchItemGadget
```

- 3b** Locate the following line:

```
<!ENTITY ICAVersion "6,30,1050,0">
```

3c Change the version to the new version number. If you are not sure of the new version number, you can open the .cab file, extract one of the .DLL files, then check the version information on the file properties.

3d Save the changes to the file.

4 Refresh the portal style sheets by restarting the Tomcat servlet engine.

or

If you are using the NPS 1.5 full version (rather than the runtime version), you can refresh the style sheets by logging in to NPS as a portal admin, and clicking Portal Administration > Administer the Portal > Portal > Refresh Portal. On the Refresh Portal page, select Cached Stylesheets for Backend Rendering, then click Refresh Only.

The next time a user launches an ICA thin-client application through the Launch Item gadget, the new ICA Web client is installed to the user's workstation.

Installing the DeFrame Client

In order for a user to launch thin-client applications from Novell Application Launcher (ZENworks for Desktops), the full DeFrame client must be installed on the user's workstation. Novell Application Launcher must also be installed on the workstation.

1 On a Windows 95/98 workstation, log in as the user.

or

On a Windows NT/2000/XP workstation, log in as a member of the Administrators group.

2 Run the DeFrame Workstation Setup program (setup.exe located in the \deframe\wksetup directory on the CD).

Issues for Users to Be Aware Of

You might want to make users aware of the following issues they might encounter while using a thin-client application:

- ◆ Printing to a local printer (a printer connected to the user's workstation) or a network printer captured to a local port is supported with both the RDP client and the ICA client.
- ◆ Printing to a non-captured network printer is not supported with the RDP client. The ICA client supports printing to a Novell Distributed Print Services™ (NDPS®) printer defined on the user's workstation.
- ◆ Saving files to a network location is supported with both the RDP client and the ICA client. The user's terminal server login script (Windows or Novell Client™) must provide the network location access and mappings.
- ◆ Saving files to a local drive is supported with the ICA client. However, when using the RDP client, users must run the DeFrame Shared Drive Setup program, *setup.exe*, located in the `\deframe\sharedrive` directory on *Novell ZENworks OnDemand Services 2* CD. They must also install the NetBEUI protocol on their workstations. This solution will work only if the user's workstation is on the same network segment as the application's terminal server (in other words, the user does not need to go through any routers to reach the terminal server) or WINS (Windows Internet Naming Service) is configured to resolve NetBIOS names into IP addresses.

On Windows NT/2000/XP, users must be logged in as a member of the Administrators or Power Users group to successfully run the Shared Drive Setup program.

- ◆ With the RDP client or Citrix MetaFrame* 1.8, the application will use the terminal server's date and time, not the workstation's date and time. If the terminal server and the user's workstation are not in the same time zone, this could cause inaccuracies or confusion in applications that have time/date reliant functions (such as a Novell GroupWise® appointment or a Microsoft Outlook* calendar item).

14

Creating Cost Centers

Cost centers enable Novell® ZENworks® OnDemand Services™ to correctly record users' purchase charges against the users' organizations. OnDemand Services requires at least one cost center. By default, a cost center was created during the OnDemand Services installation. You can, however, create additional cost centers if your organization requires it.

Each cost center must have at least one budget holder, and all OnDemand Services users need to be assigned to a cost center. When a user requests a package, the budget holder for the user's cost center can approve or deny the request. If the request is approved, charges associated with the package's purchase or use are recorded against the cost center.

Complete the tasks in the following sections to set up your cost centers:

- ♦ [“Creating a Cost Center and Adding Budget Holders” on page 121](#)
- ♦ [“Assigning Users to the Cost Center” on page 123](#)

Creating a Cost Center and Adding Budget Holders

You create cost centers in Novell eDirectory™ by using Organizational Role objects. Each user you add as an occupant to the organizational role becomes a budget holder for the cost center.

To create a cost center and assign its budget holders:

- 1 In ConsoleOne®, right-click the container where you want to create the cost center > click New > click Object.

- 2** Select Organizational Role in the Class list, then click OK to display the New Organizational Role dialog box.
- 3** Enter a name for the cost center (for example, DEPT12345), then click OK to create the object.
IMPORTANT: Do not use an & character in the name.
- 4** Right-click the newly-created object, then click Extensions of this Object.
- 5** In the Extensions of this Object dialog box, click Add Extension.
- 6** In the list of auxiliary class extensions, select commerceBudgetHolderRole, then click OK to display the BudgetHolder page.
- 7** Fill in the following fields:
Current Budget Holder: The cost center can have more than one budget holder. This allows for reassignment of the budget holder responsibilities from one person to another (for example, if a budget holder goes on vacation). The current budget holder determines which budget holder is active.
User Containers: Budget holders can create user accounts in eDirectory. By default, the user accounts will be created in the same container as the cost center object. If you add one container to this list, the users will be created in that container. If you add two or more, the budget holder will be able to select which container a user is created in.
- 8** Click OK to close the page, then click Close to close the Extensions of this Object dialog box.
- 9** Right-click the object > click Properties.
- 10** On the Identification page, use the Occupant list to add each user you want to assign as a budget holder for the cost center. Include the user you defined as the current budget holder.
IMPORTANT: Adding a user to the Occupant list not only makes the user a budget holder for the cost center but also gives him or her security equivalence to the cost center object, which is required for budget holders. If you do not define the current budget holder as an occupant, you need to manually assign the user to the cost center's Security Equal to Me list.
- 11** Click OK.

Assigning Users to the Cost Center

After you've created a cost center and assigned its budget holders, you need to assign users to the cost center.

If a user already exists in eDirectory, you must use ConsoleOne to assign him or her to the cost center. If the user doesn't exist in eDirectory, the cost center's budget holder can create the user's account through the OnDemand Services User Administration gadget, at which time the user will automatically be assigned to the cost center.

To assign an existing user to a cost center:

- 1** In ConsoleOne, right-click the User object > click Properties.
- 2** Click the OnDemand tab to display the Purchases page.
- 3** Fill in the following fields:

Account ID: The user's account ID is recorded on each Purchase and Usage object that he or she generates. If you have a third-party billing system or reporting system, you can base your billing or reporting on the account ID.

The account ID can be a unique identifier, such as the user's employee workforce number, or it can be a common identifier, such as the user's cost center ID as it is defined in the billing or reporting system.

Cost Center: Browse to and select the cost center (represented by an Organizational Role object) that you want the user assigned to.

Allow Users to Unsubscribe to Packages: If you want the user to be able to cancel his or her subscription to a package, select this option. If you do, the user can use the Workflow Tracking gadget to view an approved subscription purchase and cancel the subscription.

Receive E-Mail Notifications: Select this option to enable the user to receive e-mail messages notifying him or her of approved or denied purchases. The user's E-Mail Address setting (User object > General tab > Identification page) must also be configured with the user's correct e-mail address .

- 4** Click OK to save the information.

15

Billing for Purchases

To be able to bill for purchases, you need to extract the purchase information from Novell® eDirectory™ to an existing billing system. If you already have an interface (for example, a DirXML® or JDBC* driver) for accessing and managing information from eDirectory, you can modify that interface to also extract and manage the purchase information. If you don't have an interface, you will need to develop one or have one developed for you.

Developing Your Own Interface

The [Novell Developer Kit site \(http://developer.novell.com/ndk\)](http://developer.novell.com/ndk) contains developer kits to help you access and manage information in eDirectory. The *eDirectory SDK for Java* kit contains the libraries, tools, sample code, and documentation necessary for you to fully integrate a Java application with eDirectory and the *eDirectory SDK for C* kit contains similar items for you to integrate a C application.

As you build your interface, keep in mind the following:

- ♦ The Purchase objects and Usage objects contain the information (cost center, user name, user account ID, and so forth) you will want to extract for billing purposes.
- ♦ To decide which Purchase objects to bill for, you might want to key off the Expired status or date in the Purchase object.
- ♦ After you have extracted a purchase's information from eDirectory into your billing system, you'll want to remove the Purchase object and its Usage objects. If you mark the Purchase object as Expired and Billed, the Maintenance process will remove it and its Usage objects. Because a

Purchase object cannot be deleted until after its Usage objects have been deleted, it might take the Maintenance process two cycles to remove the objects: the first to remove the Usage objects and the second to remove the Purchase object.

- ◆ You can develop Java classes that perform tasks on Purchase and Usage objects, and then have Maintenance process run the classes. For information, see the [purchaseClassx](#) and [usageClassx](#) entries under “Modifying the Maintenance and UsageServer Configuration File (Commerceserver.cfg)” on page 140.

Getting Development Help

If you want help developing an interface, Novell provides several options:

- ◆ **Developer Labs:** Developer Labs will work side-by-side with you to develop the interface between your billing system and eDirectory. For information, see the [Novell Developer Support site \(http://developer.novell.com/support\)](http://developer.novell.com/support).
- ◆ **Novell Consulting:** Novell ConsultingSM will work with you to design, develop, and implement a solution that best fits your environment and needs. For information, see the [Novell Consulting site \(http://www.novell.com/consulting\)](http://www.novell.com/consulting)

16

Changing the Status of a Purchase

When a user purchases a package, Novell® ZENworks® OnDemand Services™ creates a Purchase object under the Package object in Novell eDirectory™. In addition, if the purchase is usage-based, when the user runs any of the package's applications the UsageServer process creates a Usage object under the Purchase object.

Throughout the lifetime of a user's purchase, the following statuses are recorded on the Purchase object to track and manage the purchase:

- ♦ **Billed:** The user's cost center has been billed for the purchase. This status is not marked automatically. To change this status, you must manually mark it or have another application (such as a Java application that provides an interface between your billing system and eDirectory) mark it.
- ♦ **Expired:** The purchase has expired. OnDemand Services automatically marks this status whenever the purchased time has been used or the ending date for the purchase has passed.
- ♦ **Deletable:** The purchase is 30 days past its expiration date and has been billed. OnDemand Services automatically marks this status. After the purchase is marked deletable, the Maintenance process will delete the Purchase object and its Usage objects. Because a Purchase object cannot be deleted until after its Usage objects have been deleted, it might take the Maintenance process two cycles to remove the objects: the first to remove the Usage objects and the second to remove the Purchase object.

Thirty (30) is the default number of days. You can change this number in the `commerceserver.cfg` file, located in the `sys:\ondemand` directory on a NetWare® server and the `\ondemand` directory on a Windows server. You

can also determine how often the Maintenance process runs; the default is every 10 minutes. For additional information, see [“Starting the Maintenance Process” on page 64](#).

- ♦ **Subscription:** The purchase is a subscription purchase. When this subscription period expires, OnDemand Services will automatically renew the subscription and create a new Purchase object for that subscription period.

To modify the status of a purchase:

- 1** In ConsoleOne[®], right-click the Purchase object > click Properties.
A Purchase object is located under its Package object.
- 2** On the General page, select or deselect the status you want to change.
- 3** Click OK.

17

Managing the Maintenance and UsageServer Processes

Novell® ZENworks® OnDemand Services™ includes two Java processes, Maintenance and UsageServer, that provide Novell eDirectory™ maintenance and application usage-tracking. The following sections provide information to help you understand and manage the two processes:

- ♦ [“Understanding the Maintenance Process” on page 129](#)
- ♦ [“Understanding the UsageServer Process” on page 130](#)
- ♦ [“Explicitly Defining the UsageServer’s Address” on page 134](#)
- ♦ [“Installing Additional Maintenance and UsageServer Processes” on page 137](#)
- ♦ [“Modifying the Maintenance and UsageServer Configuration File \(Commerceserver.cfg\)” on page 140](#)
- ♦ [“Modifying Logging Settings” on page 143](#)

Understanding the Maintenance Process

The Maintenance process monitors the Purchase and Usage objects associated with packages stored in eDirectory. When a purchase expires, the Maintenance process marks it as Expired. Thirty days later (by default), the Maintenance process deletes any expired Purchase objects (and the associated Usage objects) provided that you’ve marked them as Billed. The Maintenance process will not mark a Purchase object as billed; you must do it manually (see [Chapter 16, “Changing the Status of a Purchase,” on page 127](#)) or create a utility to do so (see [Chapter 15, “Billing for Purchases,” on page 125](#)).

You can run multiple Maintenance processes if necessary. However, each process must be responsible for a specific portion of the eDirectory tree. In other words, you cannot have two Maintenance processes maintaining the same Purchase and Usage objects. You control this by using the `baseContext` setting in the `commerceserver.cfg` file to specify the base container for the Maintenance process. The Maintenance process will only maintain Purchase and Usage objects located within the base container and its subcontainers. For information about the `commerceserver.cfg` file, see [“Modifying the Maintenance and UsageServer Configuration File \(Commerceserver.cfg\)” on page 140](#)

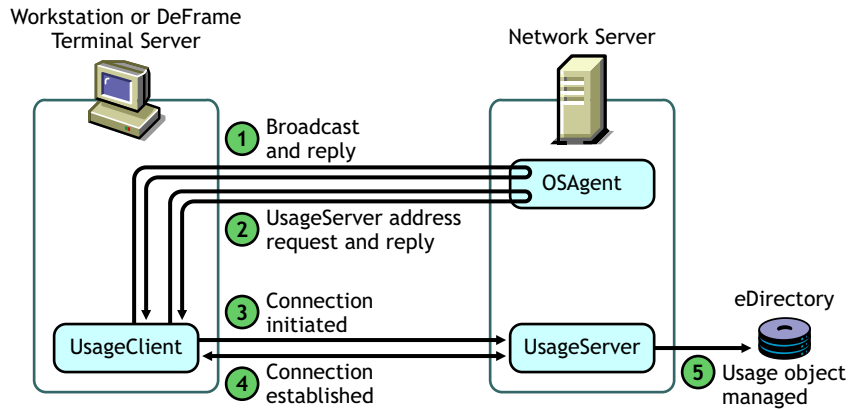
Understanding the UsageServer Process

The UsageServer combines with the UsageClient to track application usage. You must run the UsageServer if any packages use a pricing scheme based on time or use (see [“Establishing the Package’s Cost” on page 81](#)). The following sections provide information about the UsageServer process:

- ◆ [“How the UsageServer and UsageClient Work” on page 130](#)
- ◆ [“What to Do If the UsageServer and the UsageClient Are in Different Subnets” on page 131](#)
- ◆ [“Multiple UsageServers and Load Balancing” on page 134](#)

How the UsageServer and UsageClient Work

The UsageServer consists of two components, the OSAgent and the UsageServer process, both of which run on the same server. The OSAgent is responsible for facilitating a connection between the UsageClient (running on the user’s workstation or the DeFrame terminal server) and the UsageServer process. The UsageServer is responsible for maintaining usage information in eDirectory. The following diagram illustrates this process.



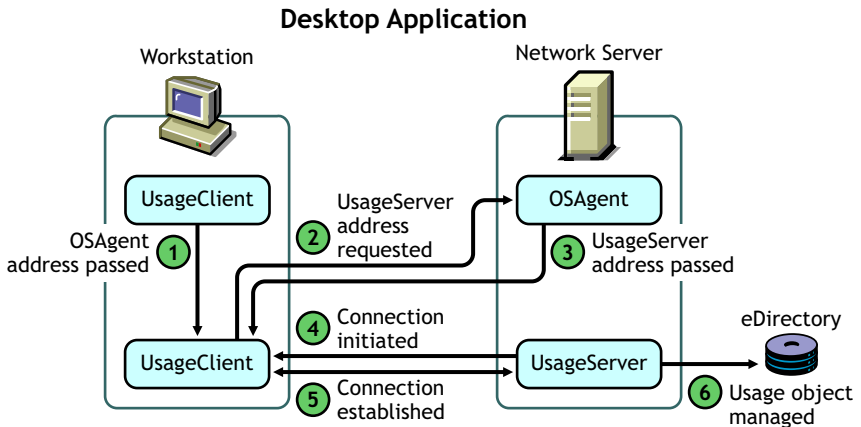
1. When a user launches a usage-based application, the UsageClient broadcasts a request looking for available OSAgents. The OSAgent replies with its IP address.
2. The UsageClient requests the UsageServer's IP address from the OSAgent. The OSAgent passes the UsageServer's IP address to the UsageClient.
3. The UsageClient initiates a connection with the UsageServer.
4. The UsageClient and UsageServer establish a connection.
5. Based on information passed to it by the UsageClient, the UsageServer then accesses eDirectory to create the Usage object associated with the application.

What to Do If the UsageServer and the UsageClient Are in Different Subnets

For the UsageClient broadcast to find an OSAgent, as shown in the diagram in the previous section, **How the UsageServer and UsageClient Work**, the UsageClient and OSAgent must be in the same subnet. If you have user workstations or DeFrame servers that will not be in the same subnet as the OSAgent, you can explicitly provide the UsageClient with the OSAgent's IP address.

Usage-Based Desktop Applications

With desktop (ZENworks for Desktops) applications, you use the Launch Item gadget's UsageServer setting to explicitly define the IP address of the OSAgent you want the UsageClient to connect to. The following diagram illustrates the desktop application usage process when the UsageServer's address is explicitly defined in the Launch Item gadget.

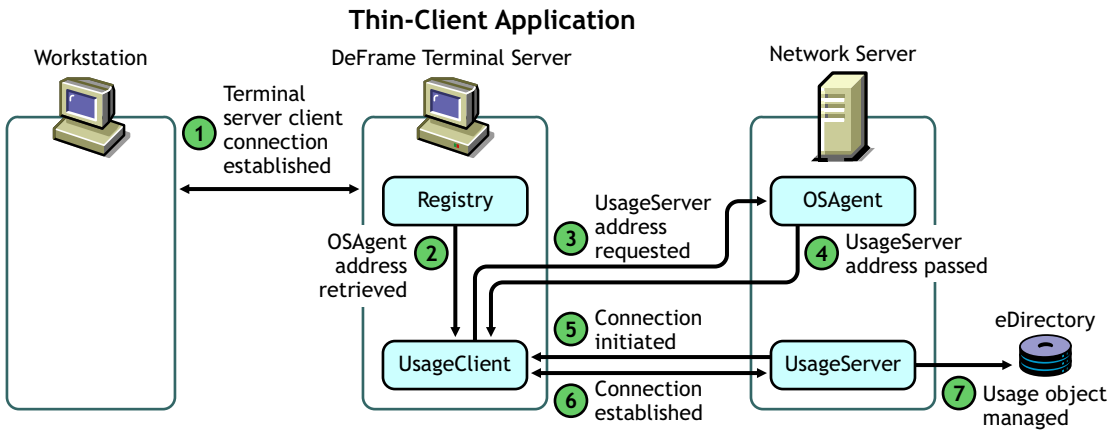


1. When a user launches a usage-based desktop application, the Launch Item gadget passes the OSAgent's IP address to the UsageClient.
2. The UsageClient requests the UsageServer's IP address from the OSAgent.
3. The OSAgent passes the UsageServer's IP address to the UsageClient.
4. The UsageClient initiates a connection with the UsageServer.
5. The UsageClient and UsageServer establish a connection.
6. Based on information passed to it by the UsageClient, the UsageServer accesses eDirectory to create a Usage object associated with the application.

For information about configuring the Launch Item gadget to provide the UsageServer address, see [“Adding the UsageServer's Address to the Launch Item Gadget” on page 135](#).

Usage-Based Thin-Client Applications

With thin-client (DeFrame™) applications, the DeFrame server's UsageClient, not the workstation's UsageClient, tracks application usage. The DeFrame server's UsageClient does not receive the OSAgent's IP address from the Launch Item gadget. Instead, you need to add the OSAgent's IP address to the DeFrame server's Windows registry. The UsageClient then pulls the IP address from the registry. The following diagram illustrates the thin-client application usage process when the UsageServer's address is explicitly defined in the DeFrame server's registry.



1. When a user launches a usage-based thin-client application, a client session is established with the DeFrame server.
2. The UsageClient, running on the DeFrame server, retrieves the OSAgent's IP address from the Windows registry.
3. The UsageClient requests the UsageServer's IP address from the OSAgent.
4. The OSAgent passes the UsageServer's IP address to the UsageClient.
5. The UsageClient initiates a connection with the UsageServer.
6. The UsageClient and UsageServer establish a connection.
7. Based on information passed to it by the UsageClient, the UsageServer then accesses eDirectory to create the Usage object associated with the application.

For information about adding the UsageServer's address to a DeFrame server's Windows registry, see [“Adding the UsageServer's Address to a DeFrame Server's Registry”](#) on page 136.

Multiple UsageServers and Load Balancing

If your usage tracking requirements are placing a heavy load on a single UsageServer, you can set up multiple UsageServers to lessen the server load. For information about installing additional UsageServers, see [“Installing Additional Maintenance and UsageServer Processes”](#) on page 137.

The UsageServer load balancing capabilities are limited to round-robinning the usage sessions. For example, if you have three UsageServers (US1, US2, and US3), the first session will be assigned to US1, the second to US2, and the third to US3. The fourth session will then be assigned to US1, the fifth to US2, and so forth. Because the usage sessions handled by one UsageServer might end before another UsageServer's sessions and new sessions are added in this round-robin manner, true load balancing might not be achieved. Overall, however, the workload will be lessened on any one server.

In a load-balanced UsageServer environment, it doesn't matter which UsageServer makes the initial usage session connection. The UsageClient will be given the address of the UsageServer that is supposed to handle the next session.

For UsageServer load balancing to work, the following conditions must be met:

- ◆ All UsageServers must be located in the same eDirectory tree.
- ◆ All UsageServers must be located in the same network subnet. If they are not, the UsageServer's OSAgents will not know about each other.

Explicitly Defining the UsageServer's Address

If you have user workstations or DeFrame servers that are not in the same subnet as the UsageServer's OSAgent, the OSAgent will not receive the UsageClient's broadcast. This will result the usage-based application failing to launch because the usage session could not be opened. To resolve this problem, you need to explicitly provide the UsageClient with the OSAgent's address.

The UsageClient receives the OSAgent's address from the Launch Item gadget for usage-based desktop (ZENworks for Desktops) applications or from the DeFrame server's Windows registry for thin-client (DeFrame) applications. The following sections provide instructions for explicitly defining the UsageServer's OSAgent's address in the Launch Item gadget and a DeFrame server's registry. If you have both usage-based desktop applications and usage-based thin-client applications, you will need to complete the tasks in both sections.

- ◆ [“Adding the UsageServer’s Address to the Launch Item Gadget” on page 135](#)
- ◆ [“Adding the UsageServer’s Address to a DeFrame Server’s Registry” on page 136](#)

Adding the UsageServer’s Address to the Launch Item Gadget

With desktop (ZENworks for Desktops) applications, you use the Launch Item gadget's UsageServer setting to explicitly define the IP address or host name of the UsageServer you want the UsageClient to connect to. When the Launch Item gadget launches a usage-based desktop application, it passes the IP address or host name is passed to the UsageClient on the user's workstation. The UsageClient then connects directly to the UsageServer's OSAgent.

The Launch Item gadget's UsageServer setting is used for directing only usage-based desktop applications to a specific UsageServer. If you also have usage-based thin-client (DeFrame) applications, see [“Adding the UsageServer’s Address to a DeFrame Server’s Registry” on page 136](#).

To add the UsageServer's address to the Launch Item gadget:

- 1** Log in to Novell Portal Services (NPS) as a portal administrator.
- 2** Click Administer the Portal.
- 3** Click Pages.
- 4** Select the page that contains the Launch Item gadget, then click Edit.

The gadget list includes all the gadgets that are included on the page. The Launch Item gadget should be in this list. However, it will be listed by its display name (for example, Launch Applications), not its gadget name (OD_LaunchItmeGadget).

- 5** In the list of gadgets, select the gadget, then click Edit.
- 6** In the UsageServer field, enter the IP address or DNS host name of the network server where the UsageServer process is installed.
- 7** Click Continue, click Save, then click OK.
- 8** Restart the Tomcat servlet engine.

Adding the UsageServer's Address to a DeFrame Server's Registry

With thin-client (DeFrame) applications, the UsageClient on the DeFrame server, not the UsageClient on the user's workstation, is used to track usage. The DeFrame server's UsageClient does not receive the IP address from the Launch Item gadget. Instead, if you want to explicitly define the UsageServer's address, you need to add the IP address or hostname to the DeFrame server's Windows registry.

The UsageServer address in the DeFrame server's registry is used for directing only usage-based thin-client applications to a specific UsageServer. If you also have usage-based desktop (ZENworks for Desktops) applications, see [“Adding the UsageServer's Address to the Launch Item Gadget” on page 135](#).

To add the UsageServer's address to the DeFrame server's registry:

- 1** At the DeFrame server, use regedit.exe to open the registry.
- 2** Add the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Novell\on-demand\Objects
```
- 3** Add a String value to the Objects key, using the following information:
Value Name: -ORBagentaddr
Value Data: Enter the IP address or DNS host name of the network server where the UsageServer process is installed.
- 4** Save the registry changes.

Installing Additional Maintenance and UsageServer Processes

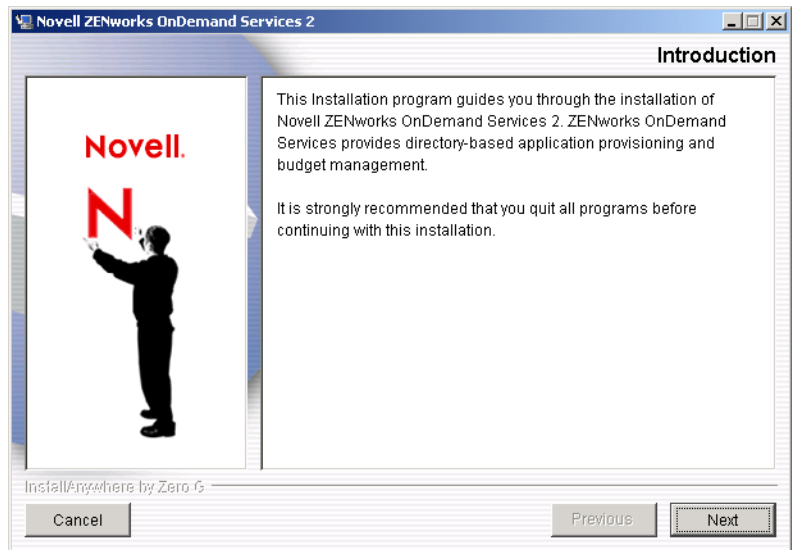
The Maintenance and UsageServer processes can be run from multiple locations at the same time, enabling you to balance the workload among several servers.

To install and configure another instance of the Maintenance and UsageServer processes:

- 1 From a Windows workstation, log in to the NetWare[®] server or Windows 2000 server where you want to install the processes.

If you are installing to a Windows 2000 server, we recommend that you run the Installation program at the server.

- 2 Insert the *Novell ZENworks OnDemand Services 2* CD.
- 3 Run `odsetup.exe` from the `\ondemand` directory.



- 4 At the Introduction page, click Next to display the License Agreement page.

- 5** After reviewing the license information, select I Accept the Terms of the License Agreement, then click Next to display the Installation Type page.
- 6** Click Custom, then click Next to display the Product Components to Install page.
- 7** Deselect all components except Maintenance and UsageServer Process, then click Next to display the Maintenance and UsageServer Platform page.
- 8** Select the type of server where you will install the processes, then click Next.
- 9** Select the server location where you want the files installed, then click Next.

The Installation program will create an \ondemand directory structure at the location you choose.

- 10** Supply the LDAP information:

LDAP Server Name: Enter the IP address or host name of the LDAP Server through which the Maintenance and Usage Server processes can authenticate to eDirectory.

Admin Name: Enter the distinguished name of the Admin user or an Admin-equivalent user. Use LDAP syntax to specify the name (for example, cn=admin,o=novell).

Container: Enter the context of the container that will become the root container for the Maintenance process. The Maintenance process will maintain purchase and usage records for any packages located within this container (and its subcontainers). Use LDAP syntax to specify the container (for example, ou=apps,ou=services,o=novell).

IMPORTANT: You need to ensure that no two Maintenance processes are configured to process the same eDirectory containers. To ensure this, it might be necessary to reconfigure your existing Maintenance processes, as explained in [Step 14](#).

- 11** Click Next to display the Maintenance and UsageServer LDAP Authentication page.
- 12** Select whether or not you want the Maintenance and UsageServer processes to use the Secure Socket Layer (SSL) protocol for their

connections to the LDAP Server, then click Next to display the Review page

- 13** Review the summary, then click Install.
- 14** When the installation is complete, before you start this instance of the Maintenance process you need to make sure that no other previously-installed Maintenance process is configured to maintain the same part of the eDirectory tree. To do so:
 - 14a** Open the `commerceserver.cfg` file for the previously-installed Maintenance process. The file is located in the `\ondemand` directory on the Maintenance process server.
 - 14b** Locate the `baseContext` property.
 - 14c** Change the context as needed.

For more information about the `baseContext` property, see [“Modifying the Maintenance and UsageServer Configuration File \(Commerceserver.cfg\)” on page 140.](#)

- 15** To start the Maintenance process: At the NetWare server console, enter `commercemaintenance.ncf`.

or

At the Windows server command prompt, change to the directory where the Maintenance process was installed (by default, `\ondemand\bin`), then enter `commercemaintenance.bat`.

- 16** To start the Usage Server process: At the NetWare server console, enter `usageserver.ncf`.

or

At the Windows server command prompt, change to the directory where the Usage Server was installed (by default, `\ondemand\bin`), then enter `usageserver.bat`.

Modifying the Maintenance and UsageServer Configuration File (Commerceserver.cfg)

The Maintenance and UsageServer processes share the same configuration file. The configuration file, named `commerceserver.cfg`, is located in the `\ondemand` directory of the server where the two processes are installed.

You can use any text editor to edit the file. The configuration settings are described below.

ldap.host: Specifies the IP address or DNS host name of the LDAP server through which the Maintenance and UsageServer processes access eDirectory.

ldap.loginName: Specifies the distinguished name of the user through which the Maintenance and UsageServer processes will authenticate to eDirectory. Use LDAP syntax to specify the name (for example, `cn=admin,o=novell`).

If the user name includes extended (multibyte) characters and the Maintenance and UsageServer processes are running on a NetWare server, you need to run `commerceserver.cfg` through the Java utility `native2ascii`. At the server console, enter the following:

```
native2ascii \ondemand\commerceserver.cfg
```

This needs to be done any time you change the user name.

ldap.useSSL: Specifies whether or not SSL will be used for the connection to the LDAP server. Values are `TRUE` and `FALSE`.

ldap.port.SSL: Specifies the SSL LDAP port number being used by the LDAP server. This port will be used if `ldap.useSSL` is set to `TRUE`. The standard SSL LDAP port is 636.

ldap.port.clearText: Specifies the non-SSL LDAP port number being used by the LDAP server. This port will be used if `ldap.useSSL` is set to `FALSE`. The standard non-SSL LDAP port is 389.

UsageHeartbeatMinutes: This setting is used by both the Maintenance and UsageServer processes to ensure accurate purchase records for packages with usage-based pricing schemes.

When a user launches a package's application, a Usage object is created under the Purchase object associated with the package. The Usage object contains a

beginning time stamp created by the UsageServer. After each 10 minutes of use (the default), the UsageServer adds an interim time stamp called the heartbeat time stamp.

When the user exits the application, the UsageServer removes the heartbeat time stamp and creates the ending time stamp. However, if something happens during the time the user is using the application so that the UsageServer is unable to create the ending time stamp, the last heartbeat time stamp remains.

The Maintenance process, during its maintenance task, looks for any Usage objects with heartbeat time stamps that are older than 10 minutes. If this situation exists, it uses the heartbeat time stamp as the ending time stamp. This ensures that the user is not billed for more time than the UsageServer was actually able to record

MaintainEveryMinutes: Specifies how often the Maintenance process will read eDirectory to process any Purchase and Usage records. The default is every 10 minutes.

Take care when configuring this setting. One of the tasks the Maintenance process performs is to revoke a user's package rights when the user's purchase expires. If you set a long maintenance interval (for example, 600 minutes or 10 hours) and the purchase pricing scheme is Clock Time, a user will continue to have access to the purchased package for up to 10 hours after the user's purchased time ended (provided the user does not log out of NPS or refresh the Launch Item gadget's list). The extra time will depend on where the Maintenance process is in its time interval when the user's purchased time runs out. For example, if it performed maintenance 4 hours prior to the user's time running out, the user could continue to use the package for another 6 hours if he or she doesn't log out of NPS or update the Launch Item gadget.

DeletePurchasesPostExpirationDays: Specifies how long the Maintenance process will wait to delete a Purchase object after the purchase has expired and been billed for (as indicated by the Expired and Billed status boxes checked on the Purchase object). The default is 1 day. Enter 0 to have Purchase objects deleted immediately upon expiration and billing. Enter -1 to never delete expired Purchase objects.

DeleteUsagePostExpirationDays: Specifies how long the Maintenance process will wait to delete a Usage object after it has expired. The default is 1 day. Enter 0 to have Usage objects deleted immediately upon expiration. Enter -1 to never delete expired Usage objects.

DeletePendingRequestExpirationDays: Specifies how long the Maintenance process will wait to delete a Purchase object created for a purchase that is still pending approval or denial. The default is 1 day. Enter 0 to have pending Purchase objects deleted immediately. Enter -1 to never delete pending Purchase objects.

DeleteRequestInfoExpirationDays: This setting applies to subscription-based purchases only. It specifies how long the Maintenance process will wait to delete the purchase request information stored in the Purchase object. The default is 1 day. Enter 0 to have purchase request information deleted immediately upon approval or denial. Enter -1 to have purchase request information deleted only when the Purchase object is deleted.

baseContext: Specifies the context of the container that the Maintenance process will use as its root container. The Maintenance process will only process purchases located within the root container and its subcontainers. Use LDAP syntax to specify the root container (for example, ou=apps,ou=services,o=novell).

purchaseClassx: The Maintenance process is multi-threaded. One thread is used to process purchases (Purchase objects) and another thread is used to process usages (Usage objects). You can use the purchaseClassx setting to specify the Java classes that you want run during the purchase thread. Currently, there are two pre-defined classes that are used:

- ◆ RequestMaintain: Used to process request information generated by the ApprovalFlow™ process. If you don't use ApprovalFlow, you don't need to have the Maintenance process run this class.
- ◆ PurchaseMaintain: Used to process Purchase objects.

The order in which you specify the classes in the `commerceserver.cfg` file is the order in which they will be processed. In the following example, the `purchaseClass1` entry is processed before the `purchaseClass2` entry:

```
purchaseClass1 = RequestMaintain
purchaseClass2 = PurchaseMaintain
```

You can also develop your own Java classes and tie them in through the `purchaseClassx` property. For example, if you have a third-party billing or reporting system, you could write a Java class to read the Purchase object attributes and export them to the billing or reporting database. By entering the

following property definition, the Maintenance process would run the Java class after the first two classes:

```
purchaseClass3 = PurchaseInfoExport
```

usageClassx: Specifies the Java classes that you want run during the Maintenance processes usage thread. Currently, there is one predefined class that is used:

- ◆ UsageMaintain: Used to process Usage objects.

The syntax is:

```
usageClass1 = UsageMaintain
```

As with the purchaseClassx property, you can tie in additional classes to perform tasks on Usage objects.

Modifying Logging Settings

The Maintenance and Usage Server processes use Jakarta Log4j by the Apache Software Foundation for logging. Each process requires its own log configuration file, although each configuration file contains the same settings.

The files, log4j_maintenance.cfg and log4j_usage.cfg, are located in the \ondemand directory on the same server volume as the Maintenance and UsageServer processes. Using the configuration settings in the files, you can set the logging information level (INFO, WARN, ERROR, FATAL, and DEBUG) and determine the output method (screen, rolling log file, and so forth).

The files contain brief descriptions of the configuration settings. For more information about the configuration settings, see the [Jakarta Log4j Web site \(http://jakarta.apache.org/log4j/docs/documentation\)](http://jakarta.apache.org/log4j/docs/documentation).

IV

Appendixes

Appendix A, “Installation Worksheets,” on page 147

Appendix B, “Gadget Configuration Settings,” on page 157

Appendix C, “Tabbed Portal Pages,” on page 171

Appendix D, “Upgrading from OnDemand Services 1.5,” on page 175

Appendix E, “Documentation Updates,” on page 183

A

Installation Worksheets

The following worksheets list the information you will be asked to provide during the Novell® ZENworks® OnDemand Services™ installation process. You might want to print the worksheets and record your information for use during installation.

- ♦ “OnDemand Services Installation Worksheet” on page 147
- ♦ “Novell Portal Services Installation Worksheet” on page 153

OnDemand Services Installation Worksheet

Item	Description
1) Novell Portal Services 1.5 Requirement	OnDemand Services requires version 1.5 of Novell Portal Services (NPS). If NPS 1.5 is not installed, the Installation program will install a runtime version of NPS 1.5.
Continue:	
♦ Yes	You need to make sure to not install OnDemand Services to a pre-1.5 version of NPS. Doing so will cause the Installation program to overwrite the pre-1.5 version with the NPS 1.5 runtime version. Because the runtime version's functionality has been limited to support only OnDemand Services, you will lose any gadgets, pages, and so forth configured in your pre-1.5 NPS installation.
♦ No	

Item	Description
2) LDAP Server Login <ul style="list-style-type: none"> ◆ Server IP address or hostname: ◆ Admin user context: ◆ Admin user password: 	<p>The Installation program uses LDAP to access Novell eDirectory™. You need to supply the hostname or IP address of an LDAP server that can provide access to the eDirectory tree where you want to create the OnDemand Services objects. In addition, you need to provide the LDAP context and password of the Admin user (or Admin-equivalent user).</p> <p>Example:</p> <ul style="list-style-type: none"> ◆ LDAP Server IP address or hostname: 123.456.78.90 ◆ Admin user context: cn=admin,o=novell ◆ Admin user password: bigredn
3) Company Root <ul style="list-style-type: none"> ◆ Container: 	<p>Specify the container you want to designate as the root container for your company. The OnDemand Services components begin user searches in this container, which means that the container must branch down to all OnDemand Services users.</p> <p>IMPORTANT: We recommend that, if possible, all objects associated with OnDemand Services (User objects, Application objects, Cost Center object, OnDemand Administrator object, OnDemandService object, and so forth) reside within the Company Root container or its subcontainers. This ensures that the OnDemand Administrator (ODSAdmin) object has the required rights to each of these objects. If you place any of these objects outside of the Company Root, you will need to manually assign rights to the ODSAdmin object. For instructions, see “ODSAdmin Trustee Rights” on page 71.</p> <p>If the container you want to use does not exist, the Installation program will let you create it.</p> <p>Example: o=novell</p>

Item	Description
4) Packages ♦ Container:	<p>Applications are made available to users through OnDemand Services packages. Specify the container where you want your Package objects to reside. This container becomes the OnDemand Services catalog root; any packages created (and enabled) within this container are made available to users. The container you specify must reside under the Company Root (see item 3).</p> <p>If the container you want to use does not exist, the Installation program will let you create it.</p> <p>Example: ou=packages,ou=services,o=novell</p>
5) Cost Center ♦ Object name: ♦ Container:	<p>When a user purchases a package, the purchase and usage charges are recorded against the user's cost center. Each user must belong to a cost center and you can have one or more cost centers.</p> <p>Because OnDemand Services requires at least one cost center, the Installation program lets you create one. Specify the name you want to use for the cost center object and the container where you want the object created. Do not use an & character in the name.</p> <p>If the container you want to use does not exist, the Installation program will let you create it.</p> <p>Example:</p> <ul style="list-style-type: none"> ♦ Object name: CostCenter1178 ♦ Container: ou=services,o=novell
6) OnDemand Administrator ♦ Container: ♦ Password:	<p>The OnDemand Services components authenticate and receive rights to eDirectory through the OnDemand Administrator object. This object is named ODSAdmin and cannot be changed. Specify the container where you want to create the ODSAdmin object and then supply the OnDemand Administrator password.</p> <p>If the container you want to use does not exist, the Installation program will let you create it.</p> <p>Example: ou=services,o=novell</p>

Item	Description
7) OnDemandService Object <ul style="list-style-type: none"> ◆ Object name: ◆ Container: ◆ Mail server: ◆ Mail from: ◆ Mail subject: 	<p>The OnDemandService object contains e-mail and currency information required by other OnDemand Services components.</p> <p>Specify a name for the object (the default is OnDemandService) and and the container where you want the object created. If the container you want to use does not exist, the Installation program will let you create it.</p> <p>A mail server is required to send e-mail notifications for requested, approved, and denied packages. Specify the fully qualified host name of an SMTP server.</p> <p>Enter a valid e-mail address to include in the From line of e-mail notifications sent to users.</p> <p>Enter the text to include in the Subject line of the e-mail notifications.</p> <p>Example:</p> <ul style="list-style-type: none"> ◆ Object name: OnDemandService ◆ Container: ou=services,o=novell ◆ Mail server: mailserver.novell.com ◆ Mail from: jsmith@novell.com ◆ Mail subject: ApprovalFlow Notification
8) Currency Code	<p>Specify the currency code (for example, USD for United States Dollar or GBP for United Kingdom Pound Sterling) that you want associated with any monetary values displayed in OnDemand Services. The Installation program lets you select currency codes from a list. If you don't know your currency code, wait until you run the Installation program to select one from the list.</p> <p>Example: USD</p>
9) Maintenance and Usage Server Platform <ul style="list-style-type: none"> ◆ NetWare® ◆ Windows 2000 	<p>Select the operating system of the server where you will install the OnDemand Services Maintenance and UsageServer processes. The Maintenance process cleans up purchase and usage records (objects) in eDirectory. The UsageServer tracks application usage.</p> <p>Example: NetWare</p>

Item	Description
10) Maintenance and UsageServer Installation Directory	<p>Specify the server location where you want to install the Maintenance and Usage Server files. For a NetWare server, you need to use a drive mapped to the server's sys: volume. For a Windows server, you can select any location. The Installation program will create an ondemand directory structure in the location you specify.</p> <p>Example: z:\ondemand</p>
11) Maintenance and UsageServer LDAP Configuration <ul style="list-style-type: none"> ◆ LDAP Server: ◆ Admin context: ◆ Container: 	<p>The Maintenance and UsageServer processes authenticate to eDirectory through LDAP. Specify the LDAP Server and Admin user (or Admin-equivalent user) you want the processes to use to authenticate.</p> <p>The Maintenance and UsageServer processes must know what container to use as the base context for purchase and usage records. The two processes restrict their processing to purchase and usage records located in the container and its subcontainers. Because purchase and usage records are stored as objects under Package objects, you can generally use the Packages container as the base context.</p> <p>Example:</p> <ul style="list-style-type: none"> ◆ LDAP Server: 123.456.78.90 ◆ Admin context: cn=admin,o=novell ◆ Container: ou=packages,ou=services,o=novell
12) Maintenance and UsageServer LDAP Authentication <ul style="list-style-type: none"> ◆ Yes ◆ No 	<p>Specify whether the Maintenance and UsageServer processes should use the SSL protocol when communicating with the LDAP Server.</p> <p>Example: Yes</p>
13) ConsoleOne® Snap-Ins Installation Location	<p>Specify the ConsoleOne location where you want to install the OnDemand Services snap-ins. The path must include the consoleone.jar file. The consoleone.jar file will not be modified. It is simply used to determine the installation locations.</p> <p>Example: c:\novell\consoleone\1.3\consoleone.jar</p>

Item	Description
14) ZENworks Location	<p data-bbox="462 164 1206 185">This item applies only if you have ZENworks for Desktops 3.2 installed.</p> <p data-bbox="462 215 1206 386">Select the ZENworks for Desktops server location where you want to install the ZENworks OnDemand Services UsageClient files. The UsageClient files are required to track usage of applications distributed through Novell Application Launcher. The files must be installed to the same directory as zenapp32.dll (typically sys:\public or c:\novell\public).</p> <p data-bbox="462 410 659 435">Example: z:\public</p>

Novell Portal Services Installation Worksheet

The OnDemand Services Installation program launches the Novell Portal Services (NPS) Installation program. The NPS Installation program will either 1) install NPS (the runtime version) and the OnDemand Services gadgets or 2) install only the OnDemand Services gadgets if an existing NPS 1.5 portal already exists.

The following table lists the information the NPS Installation program will prompt you for. If the NPS 1.5 runtime version is being installed, you will be prompted for all of the information. If NPS 1.5 is already installed, you will be prompted for items 1, 5, and 10.

Item	Description
1) Web Server Address	Enter the IP address or DNS hostname of the Web server where you will install NPS. Example: services.novell.com
2) System Configuration	Specify the operating system (NetWare 6.0 or Windows 2000) where the Web server is running.
♦ Operating system:	
♦ Web server:	Specify the type of Web server being used. OnDemand Services requires Apache or IIS.
♦ Web App server:	Specify the Web Application server being used. OnDemand Services requires Tomcat. Example: <ul style="list-style-type: none">♦ Operating system: NetWare 6.0♦ Web server: Apache♦ Web App server: Tomcat

Item	Description
3) Protocol, Port, and Web Apps Location	Specify the communication protocol (HTTP or HTTPS) you want NPS to use.
<ul style="list-style-type: none"> ◆ Protocol: ◆ Port: ◆ Web Apps: 	<p>Specify the port number to be used. If you select HTTP as the protocol, the port defaults to port 80. If you select HTTPS, the port defaults to 443. These are the standard ports for these two protocols.</p> <p>Specify the directory where the Web Application server files are installed.</p> <p>Example:</p> <ul style="list-style-type: none"> ◆ Protocol: HTTPS ◆ Port: 443 ◆ Web Apps: \\npsserver\sys\webapps
4) Web App Server Information	Depending on the Web Application server you are using, you might be prompted for additional Web Application server information. For example, if you are using Tomcat, you will be prompted for the path to the Tomcat configuration file.
5) LDAP Directory Server and Credentials	The NPS Installation program requires access to eDirectory to update the directory. This access must be through LDAP.
<ul style="list-style-type: none"> ◆ Server and port: ◆ Admin distinguished name: ◆ Admin password: 	<p>Specify the IP address/hostname and port number of the LDAP server that can provide directory access. The standard non-SSL LDAP port is 389. The standard SSL LDAP port is 636.</p> <p>Specify the distinguished name and password of the Admin user.</p> <p>Example:</p> <ul style="list-style-type: none"> ◆ Server and port: 123.45.67.890:389 ◆ Admin distinguished name: cn=admin,o=novell ◆ Admin password: bigredn
6) Portal Object	NPS requires a Portal object in eDirectory. Specify the distinguished name (context) for the Portal object. and then specify a password seed.
<ul style="list-style-type: none"> ◆ Distinguished name: ◆ Password seed: 	<p>Example:</p> <ul style="list-style-type: none"> ◆ Distinguished name: cn=portal,ou=services,o=novell ◆ Password seed: novell

Item	Description
7) Public User Object: <ul style="list-style-type: none"> ◆ Distinguished name: ◆ Password: 	<p>NPS requires a public user in eDirectory. You can specify an existing user as the public user, or you can have the NPS Installation program create a new public user.</p> <p>Example:</p> <ul style="list-style-type: none"> ◆ Distinguished name: cn=publicuser,ou=services,o=novell ◆ Password: novell
8) User Contexts	<p>To configure NPS for contextless login (meaning users only need to enter a common name rather than a full distinguished name when logging in), specify the contexts where User objects reside.</p> <p>Example:</p> <pre>ou=users,o=novell ou=users,ou=services,o=novell</pre>
9) NPS System Gadgets	<p>NPS includes several gadgets required to configure and maintain the NPS system. By default, the objects for these gadgets are created in the same context as the Portal object (item 6). If desired, you can change the objects' default context and names.</p>
10) OnDemand Services Gadgets	<p>OnDemand Services includes nine gadgets. By default, the objects for these gadgets are created in the same context as the Portal object (item 6). If desired, you can change the objects' default context and names.</p>

B

Gadget Configuration Settings

Each of the Novell® ZENworks® OnDemand Services™ gadgets has configuration settings that you can modify to change the behavior of the gadget. The following sections provide descriptions of each gadget's configuration settings:

- ◆ “Approval Gadget (OD_ApprovalGadget)” on page 158
- ◆ “Configuration Wizard Gadget (OD_ConfigWizardGadget)” on page 158
- ◆ “Current Budget Holder Gadget (OD_CurrentBHGadget)” on page 159
- ◆ “Edit Account Gadget (OD_EditAccountGadget)” on page 159
- ◆ “Launch Item Gadget (OD_LaunchItemGadget)” on page 160
- ◆ “Package Request Gadget (OD_PackageRequestGadget)” on page 164
- ◆ “Report Gadget (OD_ReportGadget)” on page 165
- ◆ “User Administration Gadget (OD_UserAdminGadget)” on page 165
- ◆ “Workflow Tracking Gadget (OD_WorkflowTrackingGadget)” on page 169

To access a gadget's configuration settings:

- 1** Log in to Novell Portal Services (NPS) as a portal administrator.
- 2** Click Administer the Portal.
- 3** Click Pages.
- 4** Select the page that contains the gadget whose configuration settings you want to modify, then click Edit.

- 5 Select the gadget, then click Edit to display the configuration settings.

Approval Gadget (OD_ApprovalGadget)

The Approval Gadget is used by budget holders to approve or deny users' requests to purchase packages. It has the following configuration settings:

Display Name: Specifies the name that will be used when displaying the gadget on the portal page.

OnDemandService Object: Specifies the distinguished name of the OnDemandService object that was created in Novell eDirectory™ during installation. Use LDAP syntax to specify the name (for example, cn=ondemandservice,ou=services,o=novell).

Include URL in Email: Specifies whether or not to include the portal's URL in approval/denial messages sent to users. This allows users to launch the portal directly from the e-mail message rather than going to their Web browsers.

iChain URL: If you are using Novell iChain® to provide secure authentication and access to your portal, enter the appropriate iChain URL, including the protocol (for example, <http://www.novell.com/nps> or <https://123.456.78.910>). Instead of the portal's URL, this iChain URL will be used in approval/denial messages sent to users.

Window Height: Specifies the height of the window that lists the purchase requests. If the list of requests is too long to fit in the window, the window will include a vertical scroll bar. The default setting is 172.

Configuration Wizard Gadget (OD_ConfigWizardGadget)

The Configuration Wizard gadget stores the OnDemand Administrator (ODSAdmin) context and password in a secure place where the other gadgets can get it in order to authenticate to eDirectory. The Configuration Wizard must be run before any other ZENworks OnDemand Services gadgets will work. It has the following configuration settings:

Display Name: Specifies the name that will be used when displaying the gadget on the portal page.

Secure Sensitive Pages: Specifies whether or not the administrator's Web browser uses the Secure Socket Layer (SSL) protocol when transferring sensitive information (such as the OnDemand Administrator password) to and from the Web server. The default is FALSE. If you set it to TRUE, make sure the Web server supports SSL connections.

Current Budget Holder Gadget (OD_CurrentBHGadget)

The Current Budget Holder gadget is used by budget holders to select the currently active budget holder for the cost center and to add or remove budget holders. It uses the following settings:

Display Name: Specifies the name that will be used when displaying the gadget on the portal page.

OnDemandService Object: Specifies the distinguished name of the OnDemandService object that was created in eDirectory during installation. Use LDAP syntax to specify the name (for example, cn=ondemandservice,ou=services,o=novell).

Window Height: Specifies the height of the window that lists the budget holders. If the budget holder list is too long to fit in the window, the window will include a vertical scroll bar. The default setting is 172.

Edit Account Gadget (OD_EditAccountGadget)

The Edit Account gadget is used by users to edit their account information (for example, name, password, and e-mail address). It uses the following configuration settings:

Display Name: Specifies the name that will be used when displaying the gadget on the portal page.

Secure Sensitive Pages: Specifies whether or not the user's Web browser uses the Secure Socket Layer (SSL) protocol when transferring sensitive information to and from the Web server. The default is FALSE. If you set it to TRUE, make sure the Web server supports SSL connections.

Launch Item Gadget (OD_LaunchItemGadget)

The Launch Item gadget displays the applications from purchased packages. and launches the applications. It has the following configuration settings:

Display Name: Specifies the name that will be used when displaying the gadget on the portal page.

Window Height: Specifies the height of the window that lists the applications. If the application list is too long to fit in the window, the window will include a vertical scroll bar. The default setting is 150.

ZENworks Host Server: Specifies the IP address or DNS host name of the network server where ZENworks for Desktops is installed. This needs to be the same ZENworks server where you installed the OnDemand Services Usage Client files. It is also the server from which OnDemand Services will install the Novell Application Launcher™ files if the files do not already reside on a workstation where a desktop application is launched.

UsageServer: Specifies the IP address or DNS host name of the network server where the UsageServer process is installed.

Display Thin-Client Applications: Specifies whether or not to display thin-client applications in the list. These are applications distributed through Novell DeFrame™. The default is TRUE.

Display Web Content: Specifies whether or not to display Web applications (or Web content) in the list. The default is TRUE.

Display Workstation Applications: Specifies whether or not to display desktop applications in the list. These are applications distributed through ZENworks for Desktops. The default is FALSE.

RDP: Specifies whether or not a Windows Terminal Services RDP session can be used when launching a thin-client application from a terminal server. The default is TRUE.

This setting applies only if the user is not already enabled as a DeFrame user in eDirectory (User object > DeFrame tab > DeFrame Properties page > DeFrame User setting). If the user is already a DeFrame user, the user's RDP Client setting in eDirectory is used (User object > DeFrame tab > DeFrame Properties page). Otherwise, the user is enabled as a DeFrame user and the RDP Client setting is populated from the value set here.

ICA: Specifies whether or not a Citrix MetaFrame ICA session can be used when launching a thin-client application from a terminal server. The default is TRUE.

This setting applies only if the user is not already enabled as a DeFrame user in eDirectory (User object > DeFrame tab > DeFrame Properties page > DeFrame User setting). If the user is already a DeFrame user, the user's ICA Client setting in eDirectory is used (User object > DeFrame tab > DeFrame Properties page). Otherwise, the user is enabled as a DeFrame user and the ICA Client setting is populated from the value set here.

DDU: Specifies whether or not you want DeFrame to dynamically create the user's account on a terminal server each time he or she launches an application from the server. DeFrame will create the user's account, using his or her eDirectory username, when the user starts the application. After the user exits the application, DeFrame deletes the user account.

This setting applies only if the user is not already enabled as a DeFrame user in eDirectory (User object > DeFrame tab > DeFrame Properties page > DeFrame User setting). If the user is already a DeFrame user, the user's Dynamic User setting in eDirectory is used (User object > DeFrame tab > DeFrame Properties page). Otherwise, the user is enabled as a DeFrame user and the Dynamic User setting is populated from the value set here.

Store Roaming Profiles -- Registry: This setting requires the DDU setting to be TRUE. It specifies whether or not a roaming profile will be used to store the user's Windows registry settings such as property settings for folders and preference settings for applications. Saving the registry settings ensures that any application modifications that are stored in the registry will persist from one session to another. If a roaming profile is not used, the registry settings will be lost when the user exits the application and his or her user account is removed.

The default is FALSE. If you set this to TRUE, you need to specify the profile storage location in the Profile Path setting. Individual user folders will be created under the profile storage location and the registry settings saved as *username.dfr* files (for example, \profiles\jsmith\jsmith.dfr).

This setting applies only if the user is not already enabled as a DeFrame user in eDirectory (User object > DeFrame tab > DeFrame Properties page > DeFrame User setting). If the user is already a DeFrame user, the user's Store Roaming Profiles: Registry setting in eDirectory is used (User object >

DeFrame tab > DeFrame Properties page). Otherwise, the user is enabled as a DeFrame user and the Store Roaming Profiles: Registry setting is populated from the value set here.

Store Roaming Profiles -- Files: This setting requires the DDU setting to be TRUE. It specifies whether or not a roaming profile will be used to store the folders and files contained in the user's home directory (c:\documents and settings*username*). Saving the user's home directory ensures that any application files (such as documents) saved to the user's home directory will persist from one session to the next. If a roaming profile is not used, the folders and files will be lost when the user exits the application and his or her account is removed.

The default is FALSE. If you set this to TRUE, you need to specify the profile storage location in the Profile Path setting. Individual user folders will be created under the profile storage location and the folders and files saved to that directory (for example, \profiles\jsmith*folders and files*).

This setting applies only if the user is not already enabled as a DeFrame user in eDirectory (User object > DeFrame tab > DeFrame Properties page > DeFrame User setting). If the user is already a DeFrame user, the user's Store Roaming Profiles: Files setting in eDirectory is used (User object > DeFrame tab > DeFrame Properties page). Otherwise, the user is enabled as a DeFrame user and the Store Roaming Profiles: Files setting is populated from the value set here.

Profile Path: This setting is required if you set either of the Store Roaming Profiles settings to TRUE. It specifies the location where you want to store the roaming profiles.

You can enter a local drive path or a network drive path. If you enter a local drive path (for example, c:\profiles), the user's profile will be created on each terminal server's local drive. This means the user might not have consistent settings and files from one server to another.

If you enter a network drive path, the same profile will be used regardless of which terminal server the user logs in to. The terminal server's DeFrame Access Control Service, logged in to eDirectory through the DeFrame PASS_ADMIN user, creates the profile on the network drive. The PASS_ADMIN user has Admin-equivalent rights to the tree where it resides, so the DeFrame Access Control Service can create the profile location on any

server in the tree. Use a valid UNC path (*\\server\vol\directory*) to specify the profile location.

You can also include the Windows %USERNAME% variable in the path as well. For example, if you want the user's profile saved to his or her user directory on a network server, you could specify *\\server1\vol1\%USERNAME%*. Because DeFrame always creates a user profile directory in the specified location, this would result in the profile being saved to a *\\server1\vol1\username\username* directory.

This setting overrides the user's Roaming Profile Path setting in eDirectory (User object > DeFrame tab > DeFrame Properties page) unless the setting is left blank. If the setting is left blank, the eDirectory setting is used.

Display Large Icons: Specifies whether application icons are displayed in the Windows small icon format or in large icon format. The default is FALSE, which means small icon format is used.

Number of Columns: This setting applies only if Display Large Icons is set to TRUE. It specifies the number of columns to use when displaying the icons. The default is 4.

Secure Sensitive Pages: Specifies whether or not the user's Web browser uses the Secure Socket Layer (SSL) protocol when transferring sensitive information to and from the Web server. The default is FALSE. If you set it to TRUE, make sure the Web server supports SSL connections.

iFolder Server Object: If you are using Novell iFolder™ to enable users to save files from thin-client applications to network storage locations, this setting specifies the iFolder servers you want available to users. Use typeless, leading dot delimited notation to specify the iFolder Server object's distinguished name (for example, *.ifolderserver.services.novell*). You can click New Value to specify more than one iFolder server.

This setting applies only if iFolder servers are not already associated with the user (User object > DeFrame tab > iFolder Servers page). If the user is already associated with iFolder servers, those servers are used.

Package Request Gadget (OD_PackageRequestGadget)

The Package Request gadget displays available application packages and enables users to request the packages. It has the following configuration settings:

Display Name: Specifies the name that will be used when displaying the gadget on the portal page.

OnDemandService Object: Specifies the distinguished name of the OnDemandService object that was created in eDirectory during installation. Use LDAP syntax to specify the name (for example, cn=ondemandservice,ou=services,o=novell).

Include URL in Email: Specifies whether or not to include the portal's URL in notification messages sent to budget holders after users request packages.

iChain URL: If you are using Novell iChain to provide secure authentication and access to your portal, enter the appropriate iChain URL, including the protocol (for example, http://www.novell.com/nps or https://123.456.78.910). Instead of the portal's URL, this iChain URL will be used in budget holder notification messages.

Window Height: Determines the height of the window used to list available packages. If the package list is too long to fit in the window, the window will include a vertical scroll bar. The default setting is 150.

Limit Package Availability: The default setting, FALSE, causes all packages to be available to users. Setting this to TRUE causes users to have access only to the packages that they have specifically been given rights to (through eDirectory). If you set this to TRUE, you need to configure user rights to packages, otherwise users will not have any packages available to them. For instructions, see [Chapter 8, "Limiting Package Availability," on page 95](#).

Report Gadget (OD_ReportGadget)

The Report gadget is used to create reports detailing the charges accrued by users for package they've purchased. It has the following configuration settings:

Display Name: Specifies the name that will be used when displaying the gadget on the portal page.

ASP Admin: Specifies the distinguished names of the users who can generate reports for all companies located in the eDirectory tree. Use LDAP syntax to specify a name (for example, cn=sgreen,o=novell). You can click New Value to add more than one user. Make sure the user contexts you enter are valid. The contexts will not be checked during entry for validity.

Company Admin: Specifies the distinguished names of the users who can generate reports for all cost centers in a single company. Use LDAP syntax to specify a name (for example, cn=jsmith,ou=services,o=novell). You can click New Value to add more than one user. Make sure the user contexts you enter are valid. The contexts will not be checked during entry for validity.

OnDemandService Object: Specifies the distinguished name of the OnDemandService object that was created in eDirectory during installation. Use LDAP syntax to specify the name (for example, cn=ondemandservice,ou=services,o=novell).

Window Height: Specifies the height of the window that lists the packages included in the report. If the package list is too long to fit in the window, the window will include a vertical scroll bar. The default setting is 200.

User Administration Gadget (OD_UserAdminGadget)

The User Administration gadget is used by budget holders to create, modify, enable, and disable user accounts, as well as assign packages to users. It has the following configuration settings:

Display Name: Specifies the name that will be used when displaying the gadget on the portal page.

OnDemandService Object: Specifies the distinguished name of the OnDemandService object that was created in eDirectory during installation.

Use LDAP syntax to specify the name (for example, cn=ondemandservice,ou=services,o=novell).

RDP: This setting applies when creating new users through the User Administration gadget. It specifies whether or not a Windows Terminal Services RDP session can be used when launching a thin-client application from a terminal server. The default is TRUE.

This setting populates the user's RDP Client setting in eDirectory (User object > DeFrame tab > DeFrame Properties page > RDP Client setting).

ICA: This setting applies when creating new users through the User Administration gadget. It specifies whether or not a Citrix MetaFrame ICA session can be used when launching a thin-client application from a terminal server. The default setting is TRUE.

This setting populates the user's ICA Client setting in eDirectory (User object > DeFrame tab > DeFrame Properties page > ICA Client setting).

DDU: Specifies whether or not you want DeFrame to dynamically create the user's account on a terminal server each time he or she launches a thin-client application from the server. DeFrame will create the user's account, using his or her eDirectory username, when the user starts the application. After the user exits the application, DeFrame deletes the user account.

This setting populates the user's Dynamic User setting in eDirectory (User object > DeFrame tab > DeFrame Properties page > Dynamic User setting).

Store Roaming Profiles -- Registry: This setting applies when creating new users through the User Administration gadget and requires the DDU setting to be TRUE. It specifies whether or not a roaming profile will be used to store the user's Windows registry settings such as property settings for folders and preference settings for applications. Saving the registry settings ensures that any application modifications that are stored in the registry will persist from one session to another. If a roaming profile is not used, the registry settings will be lost when the user exits the application and his or her user account is removed.

The default is FALSE. If you set this to TRUE, you need to specify the profile storage location in the Profile Path setting. Individual user folders will be created under the profile storage location and the registry settings saved as *username.dfr* files (for example, \profiles\jsmith\jsmith.dfr).

This setting populates the user's Store Roaming Profile:Registry setting in eDirectory (User object > DeFrame tab > DeFrame Properties page > Store Roaming Profile:Registry setting).

Store Roaming Profiles -- Files: This setting applies when creating new users through the User Administration gadget and requires the DDU setting to be TRUE. It specifies whether or not a roaming profile will be used to store the folders and files contained in the user's home directory (c:\documents and settings*username*). Saving the user's home directory ensures that any application files (such as documents) saved to the user's home directory will persist from one session to the next. If a roaming profile is not used, the folders and files will be lost when the user exits the application and his or her account is removed.

The default is FALSE. If you set this to TRUE, you need to specify the profile storage location in the Profile Path setting. Individual user folders will be created under the profile storage location and the folders and files saved to that directory (for example, \profiles\jsmith*folders and files*).

This setting populates the user's Store Roaming Profiles:Files setting in eDirectory (User object > DeFrame tab > DeFrame Properties page > Store Roaming Profile:Files setting).

Profile Path: This setting applies when creating new users through the User Administration gadget and is required if you set either of the Store Roaming Profiles settings to TRUE. It specifies the location where you want to store the roaming profiles.

You can enter a local drive path or a network drive path. If you enter a local drive path (for example, c:\profiles), the user's profile will be created on each terminal server's local drive. This means the user might not have consistent settings and files from one server to another.

If you enter a network drive path, the same profile will be used regardless of which terminal server the user logs in to. The terminal server's DeFrame Access Control Service, logged in to eDirectory through the DeFrame PASS_ADMIN user, creates the profile on the network drive. The PASS_ADMIN user has Admin-equivalent rights to the tree where it resides, so the DeFrame Access Control Service can create the profile location on any server in the tree. Use a valid UNC path (\server\vol\directory) to specify the profile location.

You can also include the Windows %USERNAME% variable in the path as well. For example, if you want the user's profile saved to his or her user directory on a network server, you could specify \\server1\vol1\%USERNAME%. Because DeFrame always creates a user profile directory in the specified location, this would result in the profile being saved to a \\server1\vol1\username\username directory.

This setting populates the user's Roaming Profile Path setting in eDirectory (User object > DeFrame tab > DeFrame Properties page > Roaming Profile Path setting).

Disable Unsubscribe: This setting applies when creating new users through the User Administration gadget. It enables or disables a user's ability to cancel a subscription to a package. The default is TRUE, which means that the user cannot cancel subscriptions.

This setting populates the user's Disable User's Ability to Unsubscribe Package's setting in eDirectory (User object > OnDemand tab > Purchases page > Disable User's Ability to Unsubscribe Packages).

Include URL in Email: This setting applies when assigning packages to users. It specifies whether or not to include the portal's URL in e-mail notifications sent to users. Notification messages let users know when they have been given access to packages. Including the URL in the messages enables users to launch the portal directly from the messages rather than going to their Web browsers.

iChain URL: This setting applies when assigning packages to users. If you are using Novell iChain to provide secure authentication and access to your portal, enter the appropriate iChain URL, including the protocol (for example, http://www.novell.com/nps or https://123.456.78.910). Instead of the portal's URL, this iChain URL will be used in notification messages sent to users.

Window Height: Specifies the height of the window that lists the users. If the user list is too long to fit in the window, the window will include a vertical scroll bar. The default setting is 200.

Secure Sensitive Pages: Specifies whether or not the budget holder's Web browser uses the Secure Socket Layer (SSL) protocol when transferring sensitive information such as passwords to and from the Web server. The default is FALSE. If you set it to TRUE, make sure the Web server supports SSL connections.

iFolder Server Object: This setting applies when creating new users through the User Administration gadget. If you are using Novell iFolder to enable users to save files from thin-client applications to network storage locations, this setting specifies the iFolder servers you want available to users. You must enter the distinguished name of the iFolder Server object, in LDAP syntax:

```
cn=ifolderserver,ou=services,o=novell
```

This setting populates the user's iFolder Servers setting in eDirectory (User object > DeFrame tab > iFolder Servers page > iFolder Servers setting).

Available Templates: This setting applies when creating new users through the User Administration gadget. You can use this setting to specify User templates that budget holders can choose from when creating user accounts. You must specify the distinguished name of the Template object, in LDAP syntax:

```
cn=usertemplate,ou=users,o=novell
```

Workflow Tracking Gadget (OD_WorkflowTrackingGadget)

The Workflow Tracking gadget is used by users to check the status of their package purchase requests and to cancel requests. It has the following configuration settings:

Display Name: Specifies the name that will be used when displaying the gadget on the portal page.

OnDemandService Object: Specifies the distinguished name of the OnDemandService object that was created in eDirectory during installation. Use LDAP syntax to specify the name (for example, cn=ondemandservice,ou=services,o=novell).

Window Height: Specifies the height of the window that lists the user's current requests. If the request list is too long to fit in the window, the window will include a vertical scroll bar. The default setting is 150.

C

Tabbed Portal Pages

The Novell® ZENworks® OnDemand Services™ gadgets can be displayed as tabbed portal pages. With tabbed pages, you have one OnDemand Services entry in the left navigation bar. When you click the OnDemand Services entry, the right pane displays a tabbed page for each gadget the user has been assigned.

To set up tabbed pages:

- 1** Log in to your portal as a portal administrator.
- 2** Click Portal Administration, then click Administer the Portal.
- 3** Click Pages to display the Pages page.
- 4** The list of pages will include seven pages created by the NPS runtime installation program:
 - ◆ **OnDemand Services:** Controls which users have access to the tabbed pages. The OnDemand Services entry will display in the left navigation bar only if a user has been assigned this page.
 - ◆ **Applications:** Includes the Launch Item gadget (OD_LaunchItemGadget) and the Package Request gadget (OD_PackageRequestGadget).
 - ◆ **Cost Center Management:** Includes the User Administration Gadget (OD_UserAdminGadget) and the Current Budget Holder Gadget (OD_CurrentBHGadget).
 - ◆ **My Account:** Includes the Edit Account Gadget (OD_EditAccountGadget) and the Workflow Tracking Gadget (OD_WorkflowTrackingGadget).

- ◆ **ODS Configuration:** Includes the Configuration Wizard Gadget (OD_ConfigWizardGadget).
- ◆ **Purchase Approval:** Includes the Approval Gadget (OD_ApprovalGadget).
- ◆ **Reporting:** Includes the Report Gadget (OD_ReportGadget).

If any of these page names already existed before you installed the OnDemand Services patch dated Oct 2002, you will have two pages named the same. To discover which page is the tabbed page, select one of the pages, click Edit, click Properties, click All Settings, then locate the Object FDN setting. If the page's common name (CN) has a number appended to it (for example, Applications-734), it is the new tabbed page. Otherwise, it is the old page. After you discover which one is the tabbed page, you can rename it.

- 5** For each page (other than the OnDemand Services page), edit the gadget settings to configure the gadgets. In particular, make sure the OnDemandService Object setting is defined for all gadgets that have that setting.

For a description of the gadget configuration settings, see [Appendix B, "Gadget Configuration Settings,"](#) on page 157.

- 6** Assign each page (other than the OnDemand Services page) to the users who should have access to the page. In general, use the following guidelines:

- ◆ **Administrator:** All pages
- ◆ **Budget Holders:** All pages except the ODS Configuration page.
- ◆ **Users:** Applications, My Account, Reporting

- 7** Run the OnDemand Services Configuration Wizard to provide the gadgets with the OnDemand Administrator credentials. To do so:

- 7a** Log out and log in again as the portal administrator. This refreshes your portal and enables you to see the ODS Configuration page.
- 7b** Click ODS Configuration, then click Set OnDemand Administrator Credentials to display the Set Credentials window.
- 7c** In the Distinguished Name field, enter the full context of the OnDemand Administrator (ODSAdmin) object (for example, cn=ODSAdmin,ou=services,o=novell).

- 7d** In the Password field, enter the OnDemand Administrator password. This password must be the password that was entered during installation.
- 7e** Click Update.
- 8** When you are ready for users to have access to the tabbed pages, assign the OnDemand Services page to the users.
- 9** To ensure that the configuration changes are recognized, restart the Web server, Web Application Server, and JVM.

D

Upgrading from OnDemand Services 1.5

The following sections provide information to help you upgrade Novell® OnDemand Services 1.5 to Novell ZENworks® OnDemand Services 2. To successfully complete the upgrade, you need to complete the tasks in the order listed.

- ◆ “Differences Between OnDemand Services 1.5 and ZENworks OnDemand Services 2” on page 175
- ◆ “What Is Upgraded” on page 177
- ◆ “Preparing for the Upgrade” on page 179
- ◆ “Performing the Upgrade” on page 179
- ◆ “Removing Obsolete OnDemand Service Objects and Attributes” on page 180
- ◆ “Removing Unused Software Components” on page 181

IMPORTANT: Your OnDemand Services 1.5 system can continue to function while you are performing the upgrade. This allows you to continue services to your users while preparing your new OnDemand Services 2 system.

Differences Between OnDemand Services 1.5 and ZENworks OnDemand Services 2

OnDemand Services 2 offers significant functionality enhancements, such as multi-type application support, administrator- or user-initiated application provisioning, and automated approval workflow. To become familiar with

these enhancements and others, refer to the information in [Chapter 1, “What Is ZENworks OnDemand Services?”](#) on page 13.

The following functionality has not been retained from OnDemand Services 1.5 to OnDemand Services 2:

- ◆ Disk space, digital certificates, and directory classified packages
- ◆ Credit card billing
- ◆ Netscape browser support; OnDemand Services 2 only supports Microsoft Internet Explorer 5.5 or higher.

In addition, OnDemand Services 2 includes significant changes in architecture from OnDemand Services 1.5. You can refer to [Chapter 2, “Preparing for Installation,”](#) on page 19 to see many of the new software and hardware requirement changes that are a result of these architectural changes. Some of the major changes include:

- ◆ OnDemand Services 2 is hosted through Novell Portal Services (NPS) 1.5, which means that the OnDemand Services interface is presented through NPS gadgets rather than Java Server pages (JSPs). OnDemand Services 2 does not support JSPs and modifications you made to your JSPs will not be carried over to the OnDemand Services gadgets.
- ◆ OnDemand Services 1.5 had to be run on a NetWare[®] 5.1 server using Netscape Enterprise Web Server and WebSphere. Novell Portal Services 1.5 and the OnDemand Services 2 gadgets must be installed to a NetWare 6 or Windows 2000 server running an Apache or Microsoft IIS Web server and the Tomcat Web Application Server.
- ◆ An OnDemand Services 1.5 Package object could contain one application only, but an OnDemand Services 2 Package object can contain multiple applications. For example, you can have a package that contains a Web application, a traditional desktop application, and a thin-client application. When a user purchases the package (or is assigned the package), he or she receives access to all packages.

What Is Upgraded

An OnDemand Services installation can be divided into three basic components: the Web server environment, the Novell eDirectory™ information, and the Maintenance and UsageServer processes. The following sections describe what is upgraded with each of these components:

- ♦ [“Web Server Environment” on page 177](#)
- ♦ [“eDirectory Information” on page 177](#)
- ♦ [“Maintenance and UsageServer Processes” on page 178](#)

Web Server Environment

Because of the architectural changes to OnDemand Services 2 and the use of Novell Portal Services, the OnDemand Services 1.5 Web server components (JSPs, etc.) are not upgradeable. Modifications you made to your OnDemand Services 1.5 JSPs will not be carried over to the OnDemand Services gadgets.

eDirectory Information

If you install OnDemand Services 2 into the same eDirectory tree where you installed OnDemand Services 1.5, the OnDemand Services Installation program will prompt you to upgrade the OnDemand Services 1.5 Package objects. Only software packages can be upgraded. If desired, you can choose not to upgrade your software packages, in which case you will need to create new OnDemand Services 2 packages for your software applications.

When a software package is upgraded, the Package object retains its cost information and its association with the application. For desktop (ZENworks for Desktops) or thin-client (DeFrame) applications, the application is associated with the Package object by its Application object being added to the object’s Package Items list. To make the package a multi-application package (a new feature of OnDemand Services 2), you can add additional applications to the package after the upgrade. For Web applications (or music, video, or document files accessed through a URL), a Commerce Item object is created to define the application in eDirectory, and the Commerce Item object is added to the Package Items list in the Package object.

During installation, the Installation program will prompt you for a Packages container. Only packages that reside within the Package container (or its subcontainers) are available to users. If you upgrade your current software packages, you should specify the container where those software packages reside as your OnDemand Services 2 Packages container.

Coexistence Note

After you're packages are upgraded to OnDemand Services 2 packages, users can continue to access and launch them through your OnDemand Services 1.5 JSP interface until you have your OnDemand Services 2 portal interface ready to roll out. However, this is not a supported configuration and you should move away from it as quickly as possible.

Maintenance and UsageServer Processes

The Maintenance and UsageServer processes, while retaining their same basic purpose and functionality, have been improved. You need to install the new processes, either to the same location as the existing processes or to a new location.

Coexistence Note

You can continue to run the OnDemand Services 1.5 Maintenance and UsageServer processes during the upgrade. After you've upgraded the eDirectory schema to OnDemand Services 2, you need to do the following:

- ◆ Shut down the OnDemand Services 1.5 Maintenance process and start the OnDemand Services 2 Maintenance process. The OnDemand Services 1.5 Maintenance process cannot properly handle the OnDemand Services 2 Package, Purchase, and Usage objects.
- ◆ Keep the OnDemand Services 1.5 UsageServer process running so that it can handle the usage tracking for applications launched from the OnDemand Services 1.5 JSPs.
- ◆ Start the OnDemand Services 2 UsageServer process so that it can handle the usage tracking for applications launched from the OnDemand Services 2 gadgets.

Preparing for the Upgrade

- 1** Review [Chapter 2, “Preparing for Installation,” on page 19](#). The same preparation tasks apply for upgrading an existing system or installing a new system.
- 2** (Optional) Remove the existing Purchase and Usage objects in eDirectory.

This is optional because there are no known issues with keeping your existing Purchase and Usage objects. During installation, the Purchase and Usage object attributes are changed minimally. A commerceAccessPolicy attribute is added to both objects and a commerceBudgetHolder attribute is added to the Purchase object. After these changes, your existing Purchase and Usage objects should continue to work with OnDemand Services 2.

 - 2a** Change the status of each Purchase object to Billed and Deletable.
 - 2b** Wait until the Maintenance process deletes the objects.
- 3** If you don't plan to continue running the OnDemand Services 1.5 Maintenance and UsageServer processes while you upgrade (see [“Maintenance and UsageServer Processes” on page 178](#)) and they currently reside in the same NetWare directory where you want to install the OnDemand Services 2 processes (by default, sys:\ondemand), delete or rename the directory.

Performing the Upgrade

To perform your upgrade, follow the instructions in [Chapter 3, “Installing OnDemand Services,” on page 29](#). The upgrade process and new installation process are identical with the following exceptions:

- ◆ When the Installation program recognizes that OnDemand Services 1.5 is installed in the eDirectory tree, you will be prompted to upgrade your Package objects. You must specify a parent container that includes all Package objects you want upgraded.
- ◆ You will be prompted for a Packages container. OnDemand Services 2 will only display packages contained within the Packages container. Specify a parent container that includes all your Package objects.

- ◆ You will be prompted for an installation location for the Maintenance and UsageServer processes. If you install it to the same NetWare server where the OnDemand Services 1.5 processes are installed, and you want to keep the OnDemand Services 1.5 processes, specify a different directory than the one in which the 1.5 processes currently reside. The Installation program defaults to the `sys:\ondemand` directory, which was also the default directory for OnDemand Services 1.5.

Removing Obsolete OnDemand Service Objects and Attributes

After you've completed the upgrade, you can remove the following objects from eDirectory:

- ◆ SAS Service -- OnDemand for *server_name*
- ◆ CommerceService for *server_name*

Although the old CommerceService for *server_name* object is no longer used, the object class is. During installation, a new commerceService object, named OnDemandService, is created.

In addition, the schema changed significantly from OnDemand Services 1.5 to OnDemand Services 2. The primary reason for the schema changes was to support the ability to assign multiple applications to a single Package object. Most changes consisted of new attributes to existing classes (objects) or new classes. However, a few attributes are no longer used on the following classes:

- ◆ commercePackage class: commerceURL, commerceSnapin, commerceGraphic, commerceStartImmediate, commerceSeller, and mail
- ◆ commerceService class: commerceCurrencyDecimals

These attributes are still stored in the classes, but are not used and are not viewable on the Package object (commercePackage class) pages and the OnDemandService object (commerceService class) pages. If you want to remove these attributes from your eDirectory schema:

- 1 Remove the data from the obsolete attributes for each Package object. To do so:
 - 1a In ConsoleOne[®], right-click a Package object > click Properties to display the Package object's pages.

- 1b** Click the Other tab.
- 1c** Select one of the following attributes in the Attributes list:
 - ◆ commerceURL
 - ◆ commerceSnapin
 - ◆ commerceGraphic
 - ◆ commerceStartImmediate
 - ◆ commerceSeller
 - ◆ mail
- 1d** Expand the selected attributes tree to display its data values.
- 1e** Delete the data values.
- 1f** Repeat **Step 1c** through **Step 1e** for each attribute.
- 1g** Click OK to save the changes to the Package object.
- 1h** Repeat **Step 1a** through **Step 1g** for each Package object.
- 2** Remove the data from the commerceCurrencyDecimals attribute for the OnDemandService object by following the same process outlined in **Step 1**.
- 3** Wait a few days to ensure that synchronization of replicas throughout your system is complete.
- 4** Use the Schema Manager (ConsoleOne > Tools menu > Schema Manager) to remove the attributes listed in **Step 1c** and **Step 2**.

Removing Unused Software Components

The following OnDemand Services 1.5 software components are not used by OnDemand Services 2. You can remove them as soon as you are no longer using OnDemand Services 1.5.

- ◆ WebSphere* (unless it is being used for more than OnDemand Services 1.5) and OnDemand Services Web server files
- ◆ OnDemand Services 1.5 Maintenance and UsageServer processes

E

Documentation Updates

This section lists updates to the *Installation and Administration* guide that have been made since the initial release of Novell® ZENworks® OnDemand Services 2™. The information will help you to keep current on documentation updates and, in some cases, software updates (such as a Support Pack release).

The information is grouped according to the date when the *Installation and Administration* guide was updated and republished:

- ♦ “October 7, 2002 (ODS2Patch102002.exe)” on page 183
- ♦ “December 20, 2002 (Support Pack 1)” on page 186

October 7, 2002 (ODS2Patch102002.exe)

The following were made to the *Installation and Administration* guide for the OnDemand Services 2 patch dated Oct 2002:

Location	Update
“Novell eDirectory and ConsoleOne” on page 21	Under the second major bullet, clarified that the NPS 1.5 server only needs a Read/Write replica if it is a NetWare® server. Windows servers do not require a replica.
“Applying Support Pack 1” on page 40	Added information about applying the OnDemand Services 2 October 2002 patch to update the ConsoleOne® snap-ins, OnDemand Services gadgets, Maintenance and UsageServer processes, and UsageClient files.

Location	Update
"Increasing the Tomcat Servlet Engine's Heap Size" on page 40	Modified the recommended heap size memory requirements for the Tomcat servlet engine.
"Adding Budget Holders" on page 43	Under step 4, added information explaining that if you select a current budget holder who is not already an occupant of the cost center's organizational role, you need to manually add the user to the cost center's Security Equal to Me list.
"Creating a Budget Holder Portal Group" on page 55	Corrected the steps required to test a portal group (see step 5).
"Desktop Applications" on page 74	Added a list of characters that cannot be used in Application object names.
"Enabling Usage Tracking of an Application" on page 90	Added a step (step 4) explaining how to have OnDemand Services terminate an application if its usage expires while the user has it open.
Chapter 13, "Supporting Thin-Client Application Users," on page 117	Updated information about the DeFrame™, ICA, and RDP Web client files that are installed to support launching of thin-client applications from the Launch Item gadget.
	Added information about updating the ICA Web client whenever Citrix releases a new version.
"Creating a Cost Center and Adding Budget Holders" on page 121	Updated step 3 to indicate that the & character is not a valid character in cost center names.
	Updated step 10 to explain that budget holders must have security equivalent to cost center object. If you add a user to the Occupant list (on the cost center object's Identification page), he or she not only becomes a budget holder but is also added to the Security Equal to Me list. However, if you make a user who is not in the Occupant list the current budget holder (on the object's Extension page), you must manually assign the user to the Security Equal to Me list.
"Understanding the Maintenance Process" on page 129	Added this new section about how the Maintenance process works.
"Understanding the UsageServer Process" on page 130	Added this new section about how the UsageServer process works.

Location	Update
<p>“Explicitly Defining the UsageServer’s Address” on page 134</p>	<p>Added this new section explaining how to explicitly define the UsageServer’s address so that if the UsageClient is not in the same subnet as the UsageServer it can still find it and create a usage session.</p>
<p>“Modifying the Maintenance and UsageServer Configuration File (Commerceserver.cfg)” on page 140</p>	<p>Added information to the ldap.loginName setting explaining that you must run the commerceserver.cfg file through the Java utility native2ascii if 1) the login name contains extended (multibyte) characters and 2) the UsageServer and Maintenance processes are running on a NetWare server.</p> <p>Corrected information under the baseContext setting that said the UsageServer uses this setting. The UsageServer does not use the baseContext setting.</p>
<p>Item 5) Cost Center in “OnDemand Services Installation Worksheet” on page 147</p>	<p>Updated information to indicate that the & character is not a valid character in cost center names.</p>
<p>“Launch Item Gadget (OD_LaunchItemGadget)” on page 160</p>	<p>Added information about the new UsageServer setting for the Launch Item gadget.</p> <p>Removed information about the Display Free Applications setting and the Display Purchased Applications settings. These settings have been removed from the gadget.</p>
<p>“User Administration Gadget (OD_UserAdminGadget)” on page 165</p>	<p>Added information about the new Available Templates setting for the User Administration gadget.</p> <p>Corrected information about the syntax required when specifying an iFolder Server object for the iFolder Server Object setting.</p>
<p>Appendix C, “Tabbed Portal Pages,” on page 171</p>	<p>Added instructions for configuring your OnDemand Services portal pages as tabbed pages.</p>

December 20, 2002 (Support Pack 1)

The following were made to the *Installation and Administration* guide for OnDemand Services 2 Support Pack 1:

Location	Update
“Applying Support Pack 1” on page 40	Added information about applying the OnDemand Services 2 October 2002 patch to update the ConsoleOne® snap-ins, OnDemand Services gadgets, Maintenance and UsageServer processes, and UsageClient files.
“Enabling Extended Character Support” on page 41	Added information about configuring Novell Portal Services to support extended characters used in OnDemand Services object names.
“Desktop Applications” on page 74	Added information about how to successfully track usage of an application that is launched through a stub program (for example, write.exe launches wordpad.exe).
“Launch Item Gadget (OD_LaunchItemGadget)” on page 160	Clarified information about when the Launch Item gadget’s Profile Path setting is used versus the Roaming Profile Path setting in eDirectory.
“Report Gadget (OD_ReportGadget)” on page 165	Added information to the ASP Admin and Company Admin fields about ensuring that the user contexts you enter are valid because the contexts will not be checked during entry for validity.