

Novell DeFrame

2.01

www.novell.com

INSTALLATION AND ADMINISTRATION

103-000240-001



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 © 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.

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

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.

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.

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 Distributed Print Services is a trademark of Novell, Inc.

Novell iFolder 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 DeFrame?	13
Part II Installation	
2 Preparing for Installation	17
Verifying Software Requirements	17
Windows Terminal Server Requirements.	18
Client Workstation Requirements.	19
Directory Service Requirements	20
Novell iFolder Requirements (Optional)	20
Deciding Between DDU and DLU	21
3 Installing DeFrame	23
Setting Up eDirectory	24
Setting Up a Terminal Server	28
Running the DeFrame Terminal Server Setup Program	29
Applying Support Pack 1	33
Starting the DeFrame Services.	33
Configuring the Terminal Server to Support Contextless Login	34
Limiting Access to the Terminal Server File System	35
Setting Up Workstations	36
Installing the DeFrame Client.	36
Enabling Local Drive Support.	38
Enabling Printer Support	38
What's Next?	39
Part III Administration	
4 Configuring Users as DeFrame Users	43
5 Creating Terminal Server User Accounts Dynamically	47
DeFrame Dynamic User (DDU) versus ZENworks for Desktops Dynamic Local User (DLU)	47
DeFrame Dynamic User (DDU)	48
Activating DDU on a Terminal Server	49
Configuring DeFrame Users As Dynamic Users	49
Deactivating DDU on a Terminal Server	52

ZENworks for Desktops Dynamic Local User (DLU)	53
Configuring a Terminal Server to Support DLU	53
Creating a User Policy Package	55
Configuring the DLU Policy	58
Associating the User Package with Users	60
A Note About the ZENworks for Desktops 3.2 Windows Terminal Server Policy	60
6 Publishing Applications	63
Creating the Application Object in Novell eDirectory	63
Manually Creating the Application Object	64
Using the DeFrame Published Application Synchronization Utility to Create the Application Object	65
65	
Configuring the Application Object	66
Configuring ICA Client Settings	70
Configuring File System Access Rights	71
Enabling Usage Tracking	74
Distributing the Application through OnDemand Services.	75
Distributing the Application through ZENworks for Desktops	76
7 Configuring Load Balancing Services	77
How Load Balancing Works	77
Starting a Watchdog Service	78
Assigning Applications to a Watchdog Service	79
Adjusting the Load Calculation Weights.	81
Server Load Calculation	84
Modifying the Watchdog Service Configuration Settings	85
8 Managing Disconnected Sessions	89
DeFrame Disconnected Session Tracking Service	89
Application Usage Tracking	90
9 Controlling DeFrame Service Logging	93
10 Setting Up File Storage through Novell iFolder	95
Installing iFolder	96
Creating DeFrame iFolder Server Objects	96
Assigning iFolder Servers to Users	98
Providing Users with Information Needed to Log In to iFolder Servers	99
11 Setting Up Application Access through a Firewall	101
Opening Ports in the Firewall	101
Configuring Network Address Translation.	102
Defining Terminal Server Public Addresses in eDirectory.	103
Defining Applications as Public Applications	103

12 Removing DeFrame from a Terminal Server	105
Uninstalling the DeFrame Files	105
Deleting the DeFrame Server Object	106

Part IV Appendixes

A DeFrame User Configuration Utility	109
B Upgrading From DeFrame 2.0	115
Reviewing the DeFrame 2.01 Requirements.	115
Extending the eDirectory Schema	115
Upgrading DeFrame Terminal Servers.	116
Upgrading User Workstations	117
Removing Obsolete eDirectory Objects	118
C Documentation Updates	119
October 7, 2002 (Patch 102002).	119
December 20, 2002 (Support Pack 1)	121

About This Guide

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

- ♦ “Introduction” on page 11
- ♦ “Installation” on page 15
- ♦ “Administration” on page 41

Documentation Updates

For the most recent version of the DeFrame 2.01 *Installation and Administration Guide*, see the DeFrame documentation at 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 DeFrame?,” on page 13

1

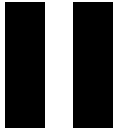
What is DeFrame?

Novell® DeFrame™ provides directory-enabled management and delivery of thin-client applications. Integrated with Novell ZENworks® OnDemand Services™ 2, it lets you provision thin-client applications as well as desktop and Web applications. Integrated with ZENworks for Desktops 3.2, it lets you deliver both thin-client and desktop applications through Novell Application Launcher™.

The following lists highlights some of the features that let you realize the benefits of using DeFrame to manage and deliver thin-client applications:

- ♦ **Directory-Based Management:** All application, user, and terminal server information is stored and managed in Novell eDirectory™. Using eDirectory makes configuration and management of application delivery quicker and easier.
- ♦ **Server Load Balancing:** You can use DeFrame Load Balancing Services to ensure that when a user launches an application it is run from the least-utilized server, maximizing both server performance and user satisfaction.
- ♦ **Dynamic User Management:** DeFrame Dynamic User (DDU) functionality can be used to dynamically manage user accounts on terminal servers. When a user launches a thin-client application, DeFrame creates the user's local account on the server. When the user exits the application, DeFrame removes the account. DeFrame also supports roaming user profiles so that user settings and documents can be persistent from session to session, regardless of which terminal server the user's current application is running from.

- ◆ **Citrix MetaFrame Support:** DeFrame supports both Microsoft* Windows* Terminal Services and Citrix* MetaFrame*.
- ◆ **Novell iFolder Integration:** You can integrate Novell iFolder™ with DeFrame to enable users to store files used with thin-client applications to network storage areas.
- ◆ **ZENworks OnDemand Services Integration:** You can include thin-client applications in OnDemand Services packages. This consolidates your thin-client, desktop, and Web applications in one location for your users and allows you to charge for thin-client applications and track application usage.
- ◆ **ZENworks for Desktops (ZfD) 3.2 Integration:** You can use the ZfD Novell Application Launcher to deliver thin-client applications to users. This consolidates your thin-client applications and desktop applications into one location. You can also use ZfD Dynamic Local User (DLU) policies to dynamically manage user accounts on terminal servers rather than using the DeFrame Dynamic User (DDU) functionality.



Installation

Chapter 2, “Preparing for Installation,” on page 17

Chapter 3, “Installing DeFrame,” on page 23

2

Preparing for Installation

This section contains information to help you implement a new installation of Novell® DeFrame™. If you are upgrading from Novell DeFrame 2.0, please see [Appendix B, “Upgrading From DeFrame 2.0,” on page 115](#).

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

- ◆ [“Verifying Software Requirements” on page 17](#)
- ◆ [“Deciding Between DDU and DLU” on page 21](#)

Verifying Software Requirements

Refer to the following sections for the Novell DeFrame software requirements:

- ◆ [“Windows Terminal Server Requirements” on page 18](#)
- ◆ [“Client Workstation Requirements” on page 19](#)
- ◆ [“Directory Service Requirements” on page 20](#)
- ◆ [“Novell iFolder Requirements \(Optional\)” on page 20](#)

Windows Terminal Server Requirements

The following are the minimum Windows terminal server requirements:

Item	Minimum Requirement
Windows Operating System	Windows 2000 Server with Service Pack 1 (latest service pack recommended).
Novell Client™	If using Novell ZENworks® for Desktops Dynamic Local User policies, Novell Client 4.81 (or later) for Windows NT*/2000. If using DeFrame Dynamic User, Novell Client 4.83 (or later) for Windows NT/2000.

Citrix MetaFrame Requirements (Optional)

Citrix MetaFrame is optional. However, if you choose to use Citrix MetaFrame, the following list gives the minimum requirements:

- ◆ Citrix MetaFrame 1.8 with Service Pack 3. The latest service pack and hotfixes are recommended. You can download them at the [Citrix Web site \(http://www.citrix.com\)](http://www.citrix.com).
- ◆ Citrix MetaFrame XP Feature Release 2 (FR2). The latest service pack and hotfixes are recommended. You can download them at the [Citrix Web site \(http://www.citrix.com\)](http://www.citrix.com).

IMPORTANT: If you install DeFrame on FR2, please refer to [Novell Technical Information Document 10073207 \(http://support.novell.com/cgi-bin/search/searchtid.cgi?/10073207.htm\)](http://support.novell.com/cgi-bin/search/searchtid.cgi?/10073207.htm) for information about additional configuration steps you need to take to ensure that contextless login works.

The following Citrix MetaFrame services are not required for use with DeFrame:

- ◆ Citrix Installation Services
- ◆ Citrix Master Browser
- ◆ Citrix Published Applications
- ◆ Citrix Program Neighborhood

If you want to continue using Citrix tools to publish while using DeFrame, you can use the DeFrame Published Application Synchronization utility to import Citrix application information into Novell eDirectory™ information. For more information, see [“Creating the Application Object in Novell eDirectory” on page 63](#).

In addition, DeFrame currently does not support the use of Citrix Load Balancing Services on the terminal servers where DeFrame is installed. If you want to load balance your DeFrame terminal servers, you must use the DeFrame Load Balancing Services.

To ensure ICA connection support, verify that you have an equivalent number of MetaFrame licenses as you do users who will use ICA sessions.

Client Workstation Requirements

The following are the minimum requirements for a workstation that will run a thin-client application from DeFrame terminal server:

Item	Minimum Requirement
Windows Operating System	Windows 98 SE. Windows NT 4.0 Workstation. Windows 2000 Professional. Windows XP.
Web Browser	Required for DeFrame to integrate with OnDemand Services to enable thin-client applications to be requested and launched. Microsoft Internet Explorer 5.5 with SP2. Microsoft Internet Explorer 6.0. Netscape browsers are not supported in this DeFrame release.

Item	Minimum Requirement
Novell Client	<p>Optional. Required only if DeFrame will integrate with ZENworks for Desktops 3.2 to enable thin-client applications to be launched from Novell Application Launcher™.</p> <p>The minimum requirement is the same as for ZFD 3.2:</p> <ul style="list-style-type: none"> ◆ Novell Client for Windows 95/98, version 3.3 with SP4. ◆ Novell Client for Windows NT/2000/XP, version 4.8 with SP4.
NetBEUI Protocol	<p>Optional. Required only if you want users to have access to their local drives. Works only if the user and the terminal server are on the same network segment or you've set up WINS (Windows Internet Name Service) to resolve NetBIOS names into IP addresses.</p>

Directory Service Requirements

DeFrame uses Novell eDirectory as its directory service. The DeFrame Installation program extends the eDirectory schema and adds several objects required by DeFrame. DeFrame supports eDirectory 8.5 or higher.

DeFrame accesses eDirectory through the Novell Client, which must be installed on each terminal server (see [“Windows Terminal Server Requirements” on page 18](#)). You do not need to have eDirectory installed on your terminal servers.

DeFrame requires ConsoleOne® 1.3.2 or higher.

Novell iFolder Requirements (Optional)

You can use Novell iFolder™ 1.x to enable users to save files from a thin-client session to one or more network storage locations. DeFrame makes the storage locations available when a user launches an application and removes the user's access when he or she exits the application. If you want the storage locations available to users even when they are not using thin-client applications configured through DeFrame, you can install the standard iFolder client to their workstations to provide continual access.

Novell iFolder runs on NetWare® 5.1, NetWare 6, Windows NT 4.0, and Windows 2000 servers. It is included in NetWare 5.1 and 6 and is also available separately. For more information about iFolder, including how to purchase it if you don't already own it, see the [Novell iFolder product information site \(http://www.novell.com/products/ifolder\)](http://www.novell.com/products/ifolder).

Deciding Between DDU and DLU

To run applications on the terminal server, users need to have local user accounts on the terminal server. You can create local accounts for each user on each terminal server, or you can have DeFrame use its DDU (DeFrame Dynamic User) functionality to dynamically manage (create and remove) user accounts.

With DDU, when a user launches an application, DeFrame creates a local user account (using the user's eDirectory username) for the user. After the user exits the application, DeFrame deletes the local user account. To preserve the user's registry settings and files from one session to another, you can configure DeFrame to support roaming profiles.

If you have ZENworks for Desktops 3.2, you can use the Dynamic Local Users (DLU) policy rather than DDU to dynamically manage local user accounts.

For information about the similarities and differences between DDU and DLU, see [Chapter 5, "Creating Terminal Server User Accounts Dynamically," on page 47](#).

3

Installing DeFrame

This section contains information to help you implement a new installation of Novell® DeFrame™. If you are upgrading from Novell DeFrame 2.0, please see [Appendix B, “Upgrading From DeFrame 2.0,” on page 115](#).

DeFrame includes two installation programs you must run and an additional two that you may need to run:

- ♦ **DeFrame eDirectory Setup:** This Setup program installs the DeFrame snap-ins to a ConsoleOne® location, extends the schema of the tree where you want DeFrame objects to reside, and adds the DeFrame objects. It must be run once.
- ♦ **DeFrame Terminal Server Setup:** This Setup program installs DeFrame files to a terminal server and defines the terminal server in Novell eDirectory™ by creating a DeFrame Server object. It must be run on each terminal server you want to use with DeFrame.
- ♦ **DeFrame Workstation Setup:** This Setup program installs the DeFrame client to a workstation. You only need to run it on a workstation if thin-client applications will be launched on the workstation through Novell Application Launcher™/Explorer (ZENworks® for Desktops).
- ♦ **DeFrame Shared Drive Setup:** This Setup program enables local drive support for users running thin-client applications in RDP client sessions. It requires the NetBEUI protocol to be enabled on the workstation and will not work unless the workstation is on the same network segment as the terminal server or WINS (Windows Internet Naming Service) is configured to resolve NetBIOS names into IP addresses.

The following sections provide instructions for running each of the Setup programs:

- ♦ “Setting Up eDirectory” on page 24
- ♦ “Setting Up a Terminal Server” on page 28
- ♦ “Setting Up Workstations” on page 36

After you’ve completed the above tasks, see “What’s Next?” on page 39 for directions on what you need to do to enable users to run thin-client applications.

Setting Up eDirectory

The DeFrame eDirectory Setup program (setup.exe) installs the DeFrame snap-ins to a ConsoleOne location, extends the schema of the tree where you want DeFrame objects to reside, and adds the DeFrame objects. You only need to run it one time.

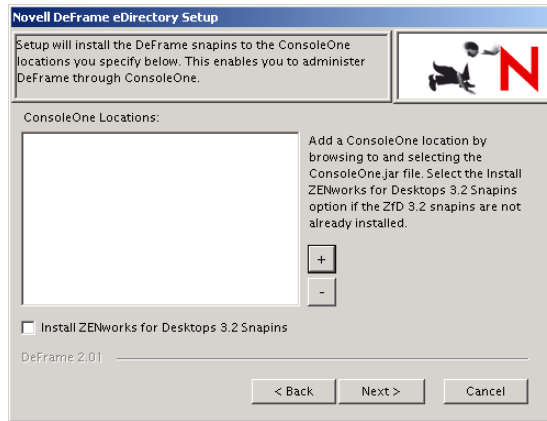
To run the Setup program:

- 1** At a Windows 2000 workstation or server, run setup.exe located in the \deframe\ndssetup directory on the *Novell ZENworks OnDemand Services 2 CD*.




- 2** At the Welcome dialog box, click Next to display the DeFrame License Agreement dialog box.

- 3 After reading the License Agreement > click Next to accept the License Agreement and display the ConsoleOne Snapins page.

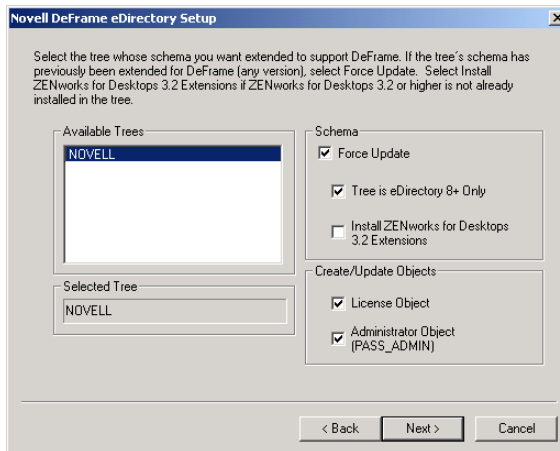


You can install the DeFrame snap-ins to one or more ConsoleOne locations. For example, you could install them to the ConsoleOne location on your local workstation and to the ConsoleOne location on a NetWare® server.

- 4 Click the  button > browse for and select the ConsoleOne.jar file that resides in the location where you want to copy the snap-ins. Repeat this step to specify additional ConsoleOne locations.

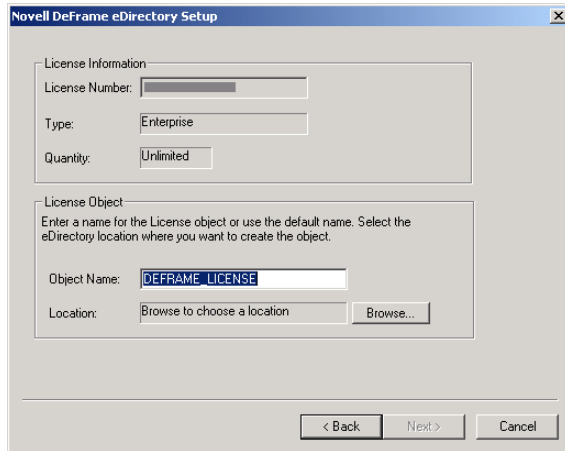
Applications are represented in eDirectory as Application objects. For Application objects to be managed through ConsoleOne, the ZENworks for Desktops snap-ins must be installed also.

- 5 Select Install ZENworks for Desktops 3.2 Snapins if they do not already reside in the ConsoleOne locations you've selected. The ZfD 3.2 snap-ins will only be in the selected ConsoleOne locations if you installed them during a previous installation of ZfD 3.2.
- 6 Click Next to install the ConsoleOne snap-in files and display the following dialog box:



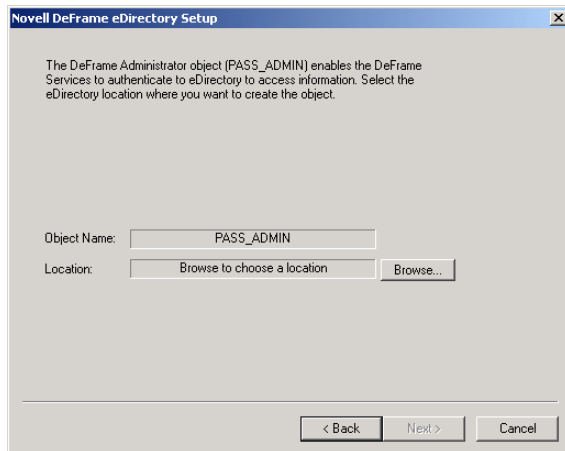
- 7** Select the tree whose schema you want to extend to support DeFrame objects.
- 8** Make sure the Force Update check box is selected (default setting).
- 9** If the selected tree is eDirectory 8 (or higher) only (meaning that none of the servers in the tree have a pre-eDirectory 8 replica), leave the Tree is eDirectory 8+ Only check box checked. If the tree includes pre-eDirectory 8 versions, uncheck this box to enable older object classes.
- 10** Select Install ZENworks for Desktops 3.2 Extensions if you do not have ZENworks for Desktops 3.2 installed in the tree.

This will cause the schema to be extended to support Application objects. Application objects are native to ZENworks for Desktops but are also required by DeFrame.
- 11** Make sure the License Object and Administrator Object (PASS_ADMIN) check boxes are selected.
- 12** Click Next to display the following dialog box:



You use this dialog box to specify the eDirectory name and context for the DeFrame License object that will be created. It also specifies the license information, which is predefined.

- 13** Click Browse to select a context for the DeFrame License object. If necessary, you can use the Create Container option to create a new container structure for the object.
- 14** Click Next to display the following dialog box:



You use this dialog box to specify the location for the DeFrame Administrator user (PASS_ADMIN). The DeFrame services use this user to authenticate to eDirectory.

- 15** Click Browse to select a context for the object. If necessary, you can use the Create Container option to create a new container structure for the object.
- 16** Click Finish to create the objects, then click Finish again to exit the Setup program.
- 17** Apply the OnDemand Services 2 Support Pack 1 to update the DeFrame snap-ins for ConsoleOne.

The OnDemand Services 2 Support Pack 1 provides updates and defect fixes designed to increase the reliability and stability of OnDemand Services 2 and its interoperability with ZENworks for Desktops 3.2 Support Pack 1. We recommend that you apply this support pack to update the ConsoleOne snap-ins you just installed. For instructions, refer to section 3.2 Updating the ConsoleOne Snap-Ins in the support pack's Readme file (Readme_ODS2SP1.txt).

If you have not yet done so, you can download the support pack from the Novell Support site (<http://support.novell.com/servlet/tidfinder/2964553>).

Setting Up a Terminal Server

To set up a terminal server for use with DeFrame, complete the following tasks. You must perform these tasks on all terminal servers you plan to use with DeFrame:

- ◆ “Running the DeFrame Terminal Server Setup Program” on page 29
- ◆ “Applying Support Pack 1” on page 33
- ◆ “Starting the DeFrame Services” on page 33
- ◆ “Configuring the Terminal Server to Support Contextless Login” on page 34
- ◆ “Limiting Access to the Terminal Server File System” on page 35

Running the DeFrame Terminal Server Setup Program

The DeFrame Terminal Server Setup program (setup.exe) installs DeFrame files to the terminal server and creates a DeFrame Terminal Server object in eDirectory.

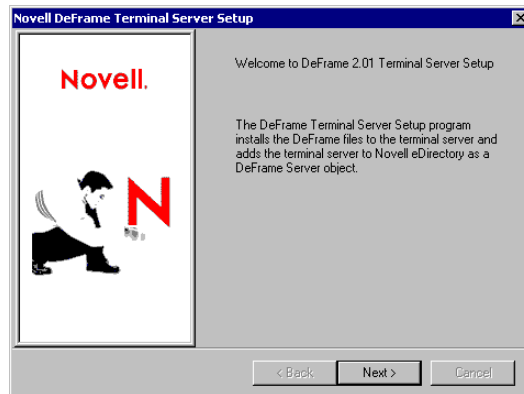
- 1** At the terminal server, make sure you are logged in to the eDirectory tree where you installed DeFrame and that your primary connection is to that tree.

To ensure that your primary tree connection is correct, right-click the N icon in the status area of the taskbar > click NetWare Connections, select the correct tree, then click Set Primary.

- 2** Use the Add/Remove Programs option to run the Setup program.

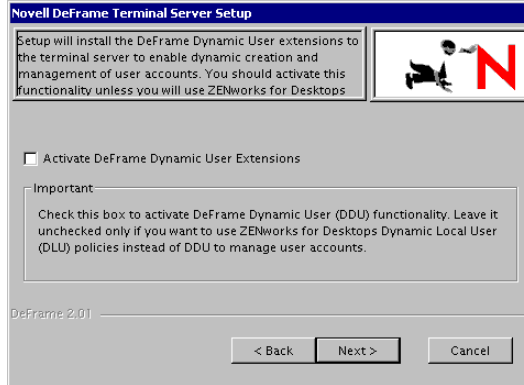
You must use Add/Remove Programs to launch the Setup program unless you use the CHANGE USER /INSTALL command at a command prompt to enable the Setup program to write changes to the system directory. To launch the Setup program using Add/Remove Programs:

- 2a** Click the Start menu > Settings > Control Panel, then click Add/Remove Programs.
- 2b** Click Add New Programs, click CD or Floppy, click Next, browse for and select the setup.exe program located in the \deframe\tssetup directory on the *Novell ZENworks OnDemand Services 2 CD*, then click Finish to start the Setup program.



- 3** Click Next to display the license agreement.

- 4 Read the license agreement, then click Next to accept the license and display the following page.

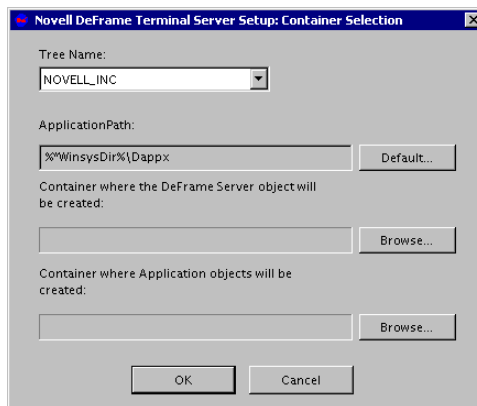


- 5 Select the Activate DeFrame Dynamic User Extensions check box.

or

If you have ZENworks for Desktops 3.2 and want to use Dynamic Local User policies to create and manage user profiles instead of the DeFrame Dynamic User functionality, make sure the option is not selected.

- 6 Click Next to install the DeFrame files and display the following dialog box:



7 Fill in the following fields:

Tree Name: Verify that the selected tree is the one where you want a DeFrame Server object created for the terminal server. This should be the same tree whose schema you extended using the DeFrame eDirectory Setup program.

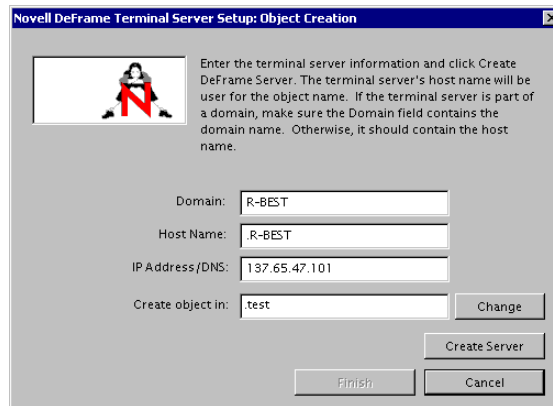
ApplicationPath: Leave this field as is. It represents an ApplicationPath variable that specifies where the DeFrame client, used to launch thin-client applications, resides on a user's workstation.

DeFrame Server Object Container: Browse for and select the container where you want the DeFrame Server object created. You can select any container in the tree. If the container does not exist, you can create the container as you browse the tree.

Application Objects Container: This container is used only if you use the DeFrame Published Application Sync service to automatically create Application objects for applications published through the Citrix Published Application Manager utility. Setup of the DeFrame Published Application Sync service is covered in [“Using the DeFrame Published Application Synchronization Utility to Create the Application Object” on page 65](#).

You can specify any container. If you already have a container for Application objects, you can specify that container.

8 Click OK, then click Next to display the following dialog box:



You use this dialog box to define the terminal server's information and create its DeFrame Server object.

9 Fill in the following fields:

Domain: If the terminal server is a Primary Domain Controller or a Backup Domain Controller, make sure the domain name is entered in this field. This enables DeFrame, when using DeFrame Dynamic User (DDU) functionality, to dynamically create user accounts in the domain database when the user logs into the terminal server.

IMPORTANT: DeFrame will also remove the name from the domain database when the user is logged out of the terminal server. If you have domain users who should not be removed from the domain database, make sure that their usernames and passwords do not match any users who will log into the terminal server. If a match occurs, DeFrame will use the existing domain user account and then remove it when the user is logged out of the terminal server.

If the terminal server is not a Primary Domain Controller or Backup Domain Controller, make sure the server's host name is entered in this field. This enables DeFrame, when using DDU functionality, to dynamically create (and remove) user accounts in the terminal server's local user database.

If you don't plan to use DDU functionality to dynamically create user accounts, you can leave this field blank. However, you will need to ensure that users have permanent local or domain user accounts on the terminal server, or you will need to use ZENworks for Desktops 3.2 Dynamic Local User (DLU) policies to create the user accounts. If you create permanent user accounts, the usernames and passwords for those accounts must match the users' eDirectory usernames and passwords.

Host Name: Displays the terminal server's host name. Verify that the name is correct. The host name will be used as the DeFrame Server object name. The dot preceding the host name is not necessary and will not be included in the object name.

IP Address/DNS: Displays the terminal server's IP address or DNS host name. Verify that the information is correct.

Create Object In: Displays the container where the terminal server's DeFrame Server object will be created. This is the container you specified previously. Click Change if you want to change it at this point.

10 Click Create Server to create the object, then click Finish.

11 Click Finish again to exit the Setup program.

12 Exit the Add/Remove Programs dialog box.

Applying Support Pack 1

You should update the DeFrame server files at this point by applying the OnDemand Services 2 Support Pack . For instructions, For instructions, refer to section 3.6 Updating DeFrame Terminal Server Files in the support pack's Readme file (Readme_ODS2SP1.txt)

If you have not yet done so, you can download the patch from the [Novell Support site \(http://support.novell.com/servlet/tidfinder/2963904\)](http://support.novell.com/servlet/tidfinder/2963904).

Starting the DeFrame Services

The DeFrame Terminal Server Setup program installs three services to the terminal server:

- ♦ **DeFrame Access Control Service:** Provides users with file system rights required to run applications, creates and manages dynamic user accounts, and keeps track of mapped drives when using Novell iFolder™ as your file storage solution. It must be run on each terminal server.
- ♦ **DeFrame Watchdog Service:** Provides server load balancing. It must be run on each terminal server that will participate in load balancing. The Watchdog service is discussed in detail in [Chapter 7, “Configuring Load Balancing Services,” on page 77](#). If you plan to use DeFrame Load Balancing Services, you should review that information.
- ♦ **DeFrame Disconnected Session Tracking Service:** Tracks disconnected sessions in order to reactivate a disconnected session whose session information (user, application, and so forth) matches the information for a newly-opened session. The newly-opened session is then removed. You should run it on each terminal server in order to minimize the proliferation of disconnected sessions.

By default, the Access Control, Watchdog, and Disconnected Session Tracking services are configured to start automatically on server startup.

To start a service immediately rather than waiting for a server restart:

- 1** Click the Start menu > Settings > Control Panel > Administrative Tools > Services.
- 2** Double-click a service. The display names for the services are:

- ◆ Novell DeACL Service
- ◆ Novell Dewatch Balancing Service
- ◆ Novell Disc Session Tracking Service

3 Click Start.

4 Click OK.

5 Repeat **Step 2** through **Step 4** for each service you want to start.

Configuring the Terminal Server to Support Contextless Login

With contextless login, when a user launches a thin-client application from his or her workstation, OnDemand Services or Novell Application Launcher passes the user's eDirectory distinguished name and password to the DeFrame client running on the terminal server. The DeFrame client then uses the name to log the user into eDirectory (which is where the dynamic user information and other required information is stored). In order for contextless login to work without the user being prompted for his or her eDirectory username and password, you need to perform the following steps:

- 1** Turn on the server's Use Client Provided Logon Information setting and turn off the Always Prompt for Password setting. To do so:
 - 1a** At the terminal server, click Start > Programs > Administrative Tools > Terminal Services Configuration
 - 1b** Highlight a connection type (the default is RDP-Tcp) and double-click to enter the properties.
 - 1c** In the Logon settings tab, check the Use Client Provided Logon Information setting and uncheck the Always Prompt for Password setting.
 - 1d** Repeat for each connection type.
- 2** Configure the default profile for the Novell Client™. To do so:
 - 2a** Right-click the Novell icon (N icon) in the status area of the taskbar > click Novell Client Properties.
 - 2b** Click the Location Profiles tab.
 - 2c** In the Location Profiles list, select Default, then click Properties to display the Location Profiles Properties dialog box.

- 2d** Select Login Service in the Service list, select Default in the Service Instance List, then click Properties to display the Novell Login dialog box.
- 2e** Deselect (turn off) the Save Profile After Successful Login option.
- 2f** Click the NDS tab.
- 2g** In the Tree field, select the eDirectory tree where the thin-client applications are configured as Application objects.
- 2h** Delete any information from the Context and Server fields.
- 2i** To save the configuration settings, click OK until you've closed all dialog boxes.

Limiting Access to the Terminal Server File System

When you install a Windows 2000 server with Terminal Server Components selected, you are given the option of defining default user and group object permissions as Windows 2000 compatible only or pre-Windows 2000 compatible.

By default, pre-Windows 2000 compatible is selected, which means users receive full access to the registry and file system on the server. This option is required in order to run many legacy (16-bit) applications.

With pre-Windows 2000 compatible permissions, the Everyone group receives Full Control rights to the drive, which is inherited down the file system. With Windows 2000 compatible permissions, the Everyone group receives Full Control right to the root of the drive and user profile directory only (permissions are not inherited) and Read and Execute rights to the rest of the file system.

You should ensure that you restrict user access to the server as much as possible. The easiest way to globally control access to system resources is to modify the Everyone group's default permissions. You can also remove the Everyone group and add a group meeting your security requirements. For detailed information, see your Windows 2000 server documentation.

Setting Up Workstations

Complete the tasks in the following sections to set up user workstations:

- ♦ “Installing the DeFrame Client” on page 36
- ♦ “Enabling Local Drive Support” on page 38
- ♦ “Enabling Printer Support” on page 38

Installing the DeFrame Client

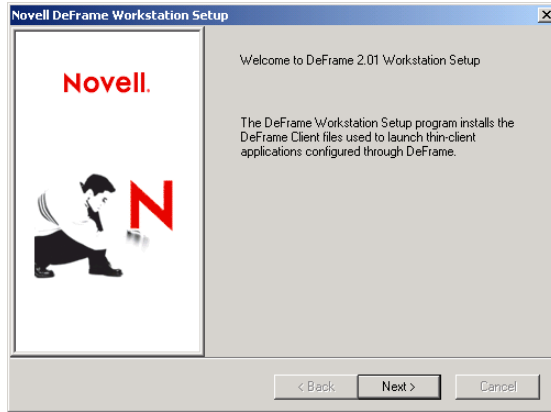
Thin-client applications can be delivered to users through two methods:

- ♦ The OnDemand Services Launch Item gadget
- ♦ Novell Application Launcher (ZENworks for Desktops)

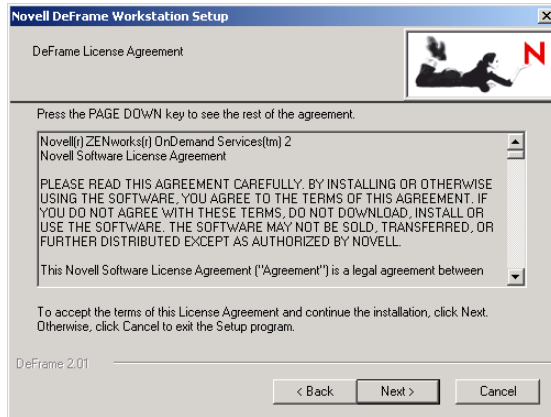
If you will be delivering applications through OnDemand Services and Novell Application Launcher, you do not need to complete the instructions in this section. The required DeFrame workstation files (DeFrame client, ICA client, and RDP client) will be installed on users’ workstations the first time they launch a thin-client application from OnDemand Services.

If you will be delivering thin-client applications through Novell Application Launcher only, then you need to install the DeFrame client on each workstation. To install the client:

- 1** Make sure the user has the rights required to install software to his or her workstation. For Windows 2000 workstations, MS Terminal Server Client needs to be installed, or the user needs to belong to the Administrator group.
- 2** Run setup.exe, located in the \deframe\wksetup directory on the *Novell ZENworks OnDemand Services 2* CD.



- 3** After reading the welcome, click Next to display the License Agreement dialog box.



- 4** Read the License Agreement, then click Next to accept the license and install the files.
- 5** When installation is complete, click Finish to exit the Setup program.
- 6** Log off and log in again.
- 7** Apply the OnDemand Services 2 Support Pack 1 to update the DeFrame files.

We recommend that you apply this support pack to update the DeFrame files you just installed. This is required if you've updated your DeFrame terminal servers to Support Pack 1. For instructions, refer to section 3.7 Updating the DeFrame Workstation Files in the support pack's Readme file (Readme_ODS2SP1.txt).

If you have not yet done so, you can download the support pack from the [Novell Support site \(http://support.novell.com/servlet/tidfinder/2964553\)](http://support.novell.com/servlet/tidfinder/2964553).

Enabling Local Drive Support

Saving files to a network location is supported with both the RDP client and the ICA client. For example, any drives that are mapped through the user's Novell Client login script or the local user account will be available to the user.

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 the *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, users must be logged in as a member of the Administrators group to successfully run the Shared Drive Setup program.

Enabling Printer Support

Local printing is supported with both the ICA and RDP client. The workstation must either have a local printer installed or a network printer captured to a local port.

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.

What's Next?

After running the DeFrame setup programs, the DeFrame files should be installed to the appropriate locations. To begin using DeFrame, you need to:

- ◆ Configure eDirectory users as DeFrame users. See [Chapter 4, “Configuring Users as DeFrame Users,”](#) on page 43.
- ◆ If you want terminal server user accounts created (and removed) dynamically, configure DeFrame users as dynamic users through DeFrame Dynamic User (DDU) functionality or Zfd 3.2 Dynamic Local User (DLU) policies. See [Chapter 5, “Creating Terminal Server User Accounts Dynamically,”](#) on page 47.
- ◆ Configure the terminal server applications you want available to users. See [Chapter 6, “Publishing Applications,”](#) on page 63.



Administration

- Chapter 4, “Configuring Users as DeFrame Users,” on page 43
- Chapter 5, “Creating Terminal Server User Accounts Dynamically,” on page 47
- Chapter 6, “Publishing Applications,” on page 63
- Chapter 7, “Configuring Load Balancing Services,” on page 77
- Chapter 8, “Managing Disconnected Sessions,” on page 89
- Chapter 9, “Controlling DeFrame Service Logging,” on page 93
- Chapter 10, “Setting Up File Storage through Novell iFolder,” on page 95
- Chapter 11, “Setting Up Application Access through a Firewall,” on page 101
- Chapter 12, “Removing DeFrame from a Terminal Server,” on page 105

4

Configuring Users as DeFrame Users

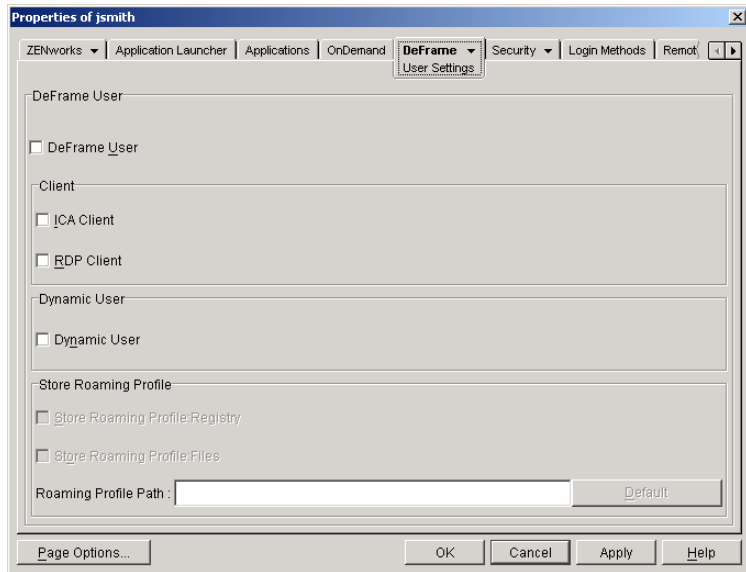
Before a user can launch a thin-client application through Novell® ZENworks® OnDemand Services™ or Novell Application Launcher™/Explorer (ZENworks for Desktops), the user must be configured as a Novell DeFrame™ user. There are several ways this can be done:

- ◆ You can use ConsoleOne® to manually configure users in Novell eDirectory™ as DeFrame users. Each user must be configured individually. See the steps below.
- ◆ You can use the DeFrame User Configuration utility to configure multiple users at the same time. See [Appendix A, “DeFrame User Configuration Utility,” on page 109](#).
- ◆ You can configure the OnDemand Services User Administration gadget to define a user as a DeFrame user and populate the DeFrame user settings when it creates the user in eDirectory. For information, see [User Administration Gadget \(OD_UserAdminGadget\) in Gadget Configuration Settings](#) in the OnDemand Services 2 *Installation and Administration* guide.
- ◆ You can configure the OnDemand Services Launch Item gadget to define a non-DeFrame user as a DeFrame user and populate the DeFrame user settings the first time the user launches a thin-client application. For information, see [Launch Item Gadget \(OD_LaunchItemGadget\) in Gadget Configuration Settings](#) in the OnDemand Services 2 *Installation and Administration* guide.

To use ConsoleOne to configure a user as a DeFrame user:

- 1 In ConsoleOne, right-click the User object > click Properties to display the User object's property pages.

2 Click the DeFrame tab to display the User Settings page.



3 Configure the following options:

DeFrame User: Select this option to enable the user as a DeFrame user.

ICA Client: Select this option to force applications that are configured to run in either an ICA or RDP client session to run in an ICA client session on the user’s workstation. The application’s terminal server must support the ICA protocol (in other words, the Windows server must have Citrix MetaFrame installed). Otherwise, an RDP client session will be used.

RDP Client: Select this option to force applications that are configured to run in either an ICA or RDP client session to run in an RDP client session on the user’s workstation.

If you select both ICA Client and RDP Client, and an application is configured to run in either type of session, an ICA client session will be used. For a summary showing the protocol (RDP or ICA) depending on the user’s Client setting, an application’s protocol setting, and a server’s supported protocols, see [“Configuring the Application Object” on page 66](#).

Dynamic User: Select this option to have DeFrame use its DeFrame Dynamic User (DDU) functionality to dynamically create the user's local account each time the user launches an application from the server. DeFrame creates 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. If the terminal server is a member of an NT domain and the Domain Controller server name is included in the terminal server's Domain field (DeFrame Server object > DeFrame Server Environment tab > Environment page) and the user's eDirectory username and password matches the user's NT domain username and password, the local user account will be created using the NT domain account information.

IMPORTANT: If a user account already exists that matches the user's eDirectory username, DeFrame will use the existing account rather than create a new account. When the user exits the application, the account will be deleted, even though DeFrame did not create the account. If you have persistent accounts (local or domain) that you want to remain persistent, you should not use DDU.

If you don't enable this option, you need to ensure that the user has a permanent local or domain user account, or you need to use ZENworks for Desktops 3.2 Dynamic Local User (DLU) policies to create the user's account. If you create permanent user accounts for the user, the username and password for those accounts must match the user's eDirectory username and password.

Store Roaming Profile: These settings are available only if Dynamic User is selected. When the user exits an application, DeFrame deletes the user's local account. You can use the following settings to save the user's profile (registry settings and files):

- ◆ **Store Roaming Profile -- Registry:** Select this option to have the user's Registry settings saved with the user's roaming profile.
- ◆ **Store Roaming Profile -- Files:** Select this option to have the user's files, including any that he or she created using the application and saved to the terminal server's local drive, saved with the user's roaming profile.
- ◆ **Roaming Profile Path:** Specify the location where you want the user's roaming profile stored. You can specify a location on the terminal server (for example, c:\profiles or \profiles) or you can specify a network location (for example, \\server1\vol1\profiles).

Click Default if you want to use the default path, as defined in the DEFAULT_REG object located at the root of the eDirectory tree.

You can also include Windows environment variables in the path as well as a special DeFrame variable (%USERNAME%). 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%.

- 4** Click OK to save your changes.

5

Creating Terminal Server User Accounts Dynamically

For a user to run an application from a terminal server, he or she needs to have a user account on the terminal server. Novell® DeFrame™ can create local user accounts dynamically using DeFrame Dynamic User (DDU) functionality or Novell ZENworks® for Desktops Dynamic Local User (DLU) policies. Refer to the following sections for information about dynamically creating local user accounts.

- ♦ “DeFrame Dynamic User (DDU) versus ZENworks for Desktops Dynamic Local User (DLU)” on page 47
- ♦ “DeFrame Dynamic User (DDU)” on page 48
- ♦ “ZENworks for Desktops Dynamic Local User (DLU)” on page 53

DeFrame Dynamic User (DDU) versus ZENworks for Desktops Dynamic Local User (DLU)

If you have ZENworks for Desktops (ZfD) 3.2, you have the option of using DeFrame Dynamic User (DDU) functionality or ZfD Dynamic Local User (DLU) policies. You cannot use both at the same time. Doing so can cause problems when users log in to a terminal server that has been configured to use DDU.

The following table compares basic feature support by DDU and DLU.

Feature	DDU	DLU
Create/Remove User Accounts and Profiles	Yes	Yes
Maintain User Profiles from Session to Session (Roaming Profiles)	Yes	Yes
Abandoned User Recovery	Yes - automatically synchronizes abandoned user profiles if possible	No - abandoned user profiles must be manually deleted
Local Group Assignments	No - users given Everyone group access	Yes - users added to groups specified by DLU policy (by default, Power Users group)
Application-Specific Windows File System Rights	Yes - for individual applications, DDU can grant file and folder access rights	No - users receive rights associated with their assigned local group
Policy Support	No	Yes

DDU is specific to terminal servers and is applied to a user only when the user logs in to a terminal server that is configured for DDU. Although you cannot use both DDU and DLU at the same time, you can use DDU in combination with other Zfd 3.2 policies.

DeFrame Dynamic User (DDU)

To configure DeFrame to manage a terminal server's local user accounts, complete the tasks in the following sections:

- ◆ [“Activating DDU on a Terminal Server” on page 49](#)
- ◆ [“Configuring DeFrame Users As Dynamic Users” on page 49](#)

To turn off DeFrame's ability to manage user accounts, complete the task in the following section:

- ◆ [“Deactivating DDU on a Terminal Server” on page 52](#)

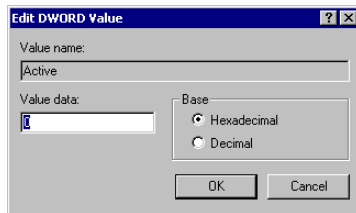
Activating DDU on a Terminal Server

For DDU to be used on a terminal server, it needs to be activated. During installation of DeFrame on the terminal server, the DeFrame Terminal Server Setup program gave the choice of activating or deactivating DDU. If DDU was activated at that time, you can skip the following steps. Otherwise, complete the steps to activate DDU.

- 1** At the terminal server, run `regedit.exe` to open the registry.
- 2** Select the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Novell\Graphical  
Login\NWLGE\LoginXDdu
```

- 3** Double-click the Active value to display the Edit DWORD Value dialog box.



- 4** Change the Value Data field to 1 (1 activates DDU, 0 deactivates it).
- 5** Click OK to save the change
- 6** Exit `regedit.exe`.

Configuring DeFrame Users As Dynamic Users

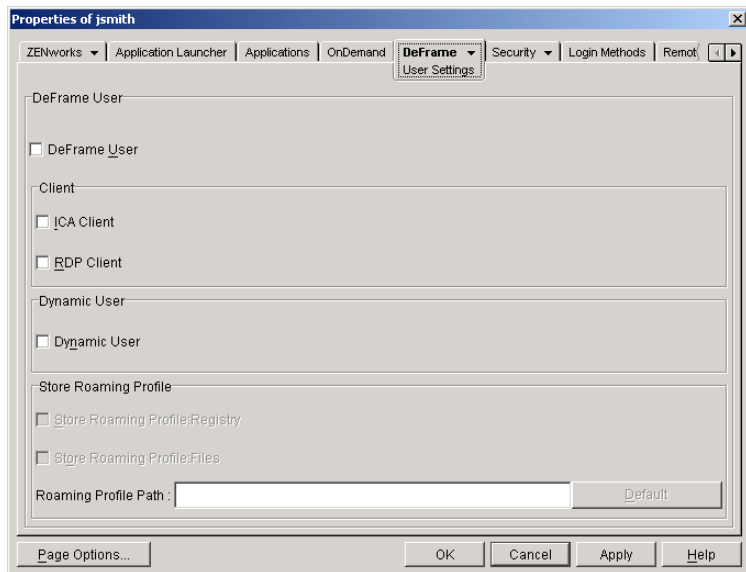
You need to enable each user whose local user account you want managed by DeFrame. At the same time, you can configure such dynamic user settings as whether or not to use roaming profiles to store registry settings and files. There are several ways to enable users and configure their settings:

- ◆ You can use ConsoleOne[®] to manually configure users in Novell eDirectory[™] as DeFrame users. Each user must be configured individually. See the steps below.

- ◆ You can use the DeFrame User Configuration utility to configure multiple users at the same time. See [Appendix A, “DeFrame User Configuration Utility,”](#) on page 109.
- ◆ You can configure the OnDemand Services User Administration gadget to define a user as a DeFrame user and populate the DeFrame user settings when it creates the user in eDirectory. For information, see [User Administration Gadget \(OD_UserAdminGadget\)](#) in [Gadget Configuration Settings](#) in the OnDemand Services 2 *Installation and Administration* guide.
- ◆ You can configure the OnDemand Services Launch Item gadget to define a non-DeFrame user as a DeFrame user and populate the DeFrame user settings the first time the user launches a thin-client application. For information, see [Launch Item Gadget \(OD_LaunchItemGadget\)](#) in [Gadget Configuration Settings](#) in the OnDemand Services 2 *Installation and Administration* guide.

To enable a user as a DeFrame user and configure his or her dynamic user settings:

- 1** In ConsoleOne, right-click the User object > click Properties to display the User object’s property pages.
- 2** Click the DeFrame tab to display the User Settings page.



3 Configure the following options:

DeFrame User: Select this option to enable the user as a DeFrame user.

ICA Client: Select this option to force applications that are configured to run in either an ICA or RDP client session to run in an ICA client session on the user's workstation. The application's terminal server must support the ICA protocol (in other words, the Windows server must have Citrix MetaFrame installed). Otherwise, an RDP client session will be used.

RDP Client: Select this option to force applications that are configured to run in either an ICA or RDP client session to run in an RDP client session on the user's workstation.

If you select both ICA Client and RDP Client, and an application is configured to run in either type of session, an ICA client session will be used. For a summary showing the protocol (RDP or ICA) depending on the user's Client setting, an application's protocol setting, and a server's supported protocols, see [“Configuring the Application Object” on page 66](#).

Dynamic User: Select this option to have DeFrame use its DeFrame Dynamic User (DDU) functionality to dynamically create the user's account each time the user launches an application from the server. DeFrame creates 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. If the terminal server is a Primary Domain Controller (PDC) or Backup Domain Controller (BDC) and the domain name is defined in the terminal server's Domain field (DeFrame Server object > DeFrame Server Environment tab > Environment page), DeFrame creates the user account in the domain database. If the terminal server is not a PDC or BDC and the terminal server host name is defined in the terminal server's Domain field (DeFrame Server object > DeFrame Server Environment tab > Environment page), DeFrame creates the user account in the terminal server's local database.

IMPORTANT: If a user account already exists that matches the user's eDirectory username, DeFrame will use the existing account rather than create a new account. When the user exits the application, the account will be deleted, even though DeFrame did not create the account. If you have persistent accounts (local or domain) that you want to remain persistent, you should not use DDU.

If you don't enable this option, you need to ensure that the user has a permanent local or domain user account, or you need to use ZENworks for Desktops 3.2 Dynamic Local User (DLU) policies to create the user's account. If you create permanent user accounts for the user, the username and password for those accounts must match the user's eDirectory username and password.

Store Roaming Profile: These settings are available only if Dynamic User is selected. When the user exits an application, DeFrame deletes the user's local account. You can use the following settings to save the user's profile (registry settings and files):

- ◆ **Store Roaming Profile -- Registry:** Select this option to have the user's Registry settings saved with the user's roaming profile.
- ◆ **Store Roaming Profile -- Files:** Select this option to have the user's files, including any that he or she created using the application and saved to the terminal server's local drive, saved with the user's roaming profile.
- ◆ **Roaming Profile Path:** Specify the location where you want the user's roaming profile stored. You can specify a location on the terminal server (for example, c:\profiles or \profiles) or you can specify a network location (for example, \\server1\vol1\profiles). Click Default if you want to use the default path, as defined in the DEFAULT_REG object located at the root of the eDirectory tree.

You can also include Windows environment variables in the path as well as a special DeFrame variable (%USERNAME%). 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%.

- 4 Click OK to save your changes.

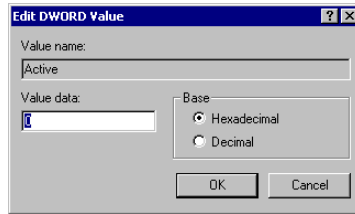
Deactivating DDU on a Terminal Server

If you decide not to use DDU on a terminal server, you need to make sure that it is deactivated. To do so:

- 1 At the terminal server, run regedit.exe to open the registry.
- 2 Select the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Novell\Graphical  
Login\NWLGE\LoginXDdu
```

- 3 Double-click the Active value to display the Edit DWORD Value dialog box.



- 4 Make sure the Value Data field is set to 0 (0 deactivates DDU, 1 activates it).
- 5 Click OK to save the change
- 6 Exit regedit.exe.

ZENworks for Desktops Dynamic Local User (DLU)

If you have ZENworks for Desktops 3.2 installed, you can use a Dynamic Local User policy to manage terminal server user accounts. The following sections provide instructions:

- ◆ “Configuring a Terminal Server to Support DLU” on page 53
- ◆ “Creating a User Policy Package” on page 55
- ◆ “Configuring the DLU Policy” on page 58
- ◆ “Associating the User Package with Users” on page 60

Configuring a Terminal Server to Support DLU

Complete the tasks in the following sections to make sure that a terminal server is configured correctly to support DLU:

- ◆ “Setting Up Workstation Manager” on page 54
- ◆ “Setting the eDirectory Tree Read by Workstation Manager” on page 54
- ◆ “Deactivating DeFrame Dynamic User (DDU)” on page 55

Setting Up Workstation Manager

The Dynamic Local User policy is applied by the Zfd Workstation Manager installed on the terminal server. Workstation Manager is installed with the Novell Client™ and the Workstation Manager service is started. You can verify that Workstation Manager is installed and running on the terminal server by checking for the Novell Workstation Manager services in the Services window.

With DeFrame installed on the terminal server, you cannot use Workstation Manager to apply workstation policies to the terminal server. To apply workstation policies to the terminal server, the Workstation Manager service must authenticate to eDirectory as the Workstation object that was imported for the terminal server. However, the DeFrame services (Access Control, Disconnected Session Tracking, Load Balancing, and so forth) authenticate to eDirectory as the DeFrame PASS_ADMIN user and establishes this connection for the Windows system services. Because Windows system services share eDirectory connections, when Workstation Manager authenticates to eDirectory, it is authenticated as the PASS_ADMIN user rather than the terminal server's Workstation object so the workstation policies configured through the terminal server's Workstation object are never applied.

To ensure that no conflicts occur, we recommend that you not import the terminal server into eDirectory as a Workstation object.

Setting the eDirectory Tree Read by Workstation Manager

Make sure Workstation Manager is configured to read the Directory tree where the DLU policy will reside.

- 1** Click the Start menu > Settings > Network and Dial-up Connections.
- 2** In the Network and Dial-up Connections dialog box, right-click the Local Area Connection icon, then click Properties.
- 3** In the list of available components, select Novell Workstation Manager, then click Properties.
- 4** Verify that the Enable Workstation Manager box is checked and that the tree is set correctly.

- 5 (Optional) Check the Tree value in the Windows registry, underneath the HKEY_LOCAL_MACHINE/SOFTWARE/NOVELL/Workstation Manager/Identification key.

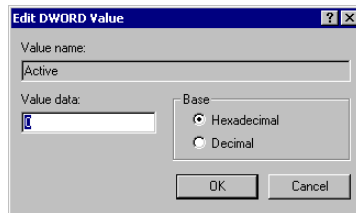
Deactivating DeFrame Dynamic User (DDU)

Make sure that DeFrame Dynamic User (DDU) is deactivated on the terminal server. To do so:

- 1 At the terminal server, run regedit.exe to open the registry.
- 2 Select the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Novell\Graphical  
Login\NWLGE>LoginXDdu
```

- 3 Double-click the Active value to display the Edit DWORD Value dialog box.



- 4 Make sure the Value Data field is set to 0 (0 deactivates DDU, 1 activates it).
- 5 Click OK to save the change
- 6 Exit regedit.exe.

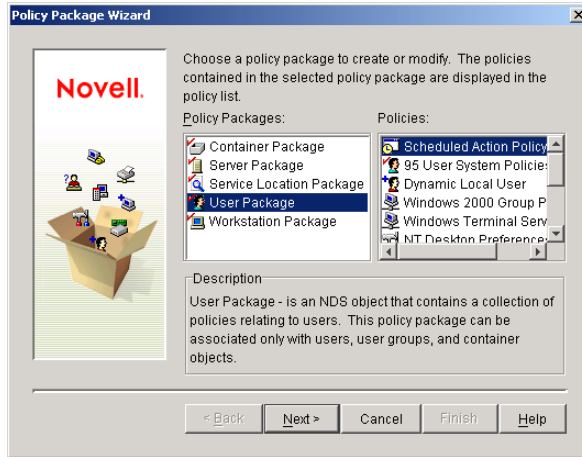
Creating a User Policy Package

The DLU policy is part of a User Policy package. You can enable the DLU policy on existing User Policy packages, or you can create new User Policy packages specifically for the DLU policy. If you already have a User Policy package that you want to use for the DLU policy, skip to [“Configuring the DLU Policy” on page 58](#).

The following steps provide brief instructions for creating a User Policy package. For more detailed information, see the ZENworks for Desktops 3.2 documentation at the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

To create a User Policy package:

- 1 In ConsoleOne, right-click the container where you want to create the package's object > click New > click Policy Package to display the Policy Package Wizard.



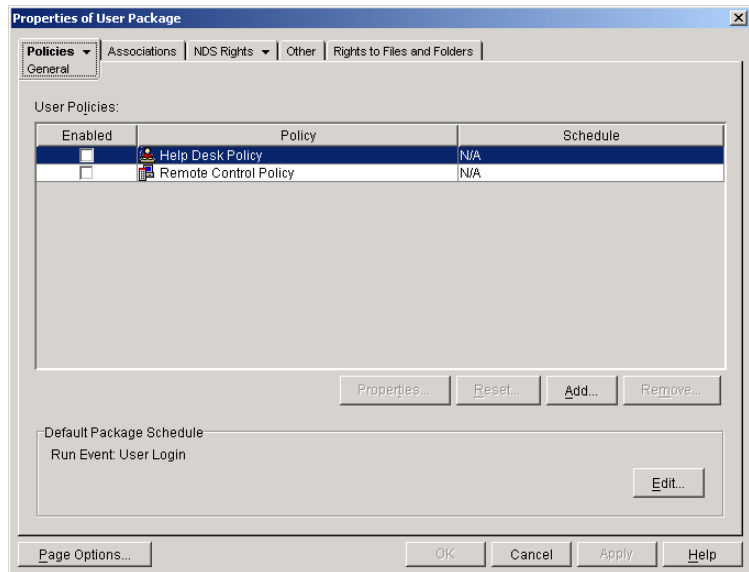
- 2 In the Policy Packages list, select User Package, then click Next.



- 3** If necessary, change the package's object name and the container where it will be created, then click Next.



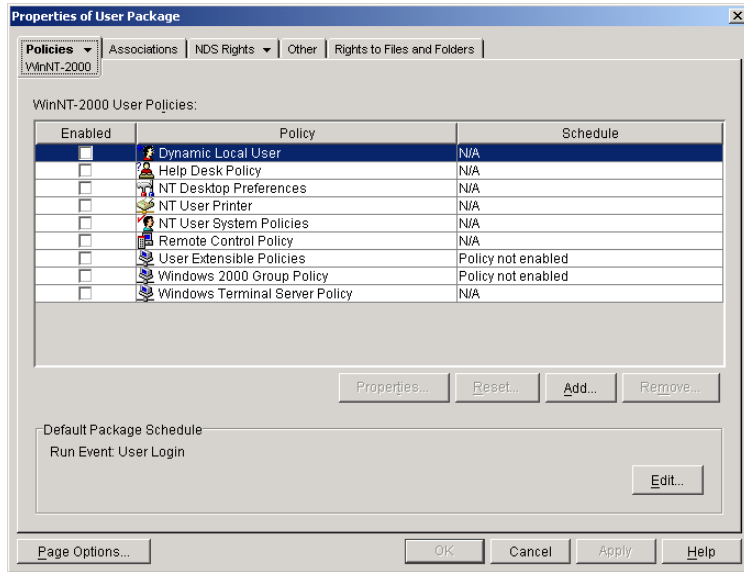
- 4** In the Summary page, select Define Additional Properties, then click Finish to create the User Package object and display the object's property pages.



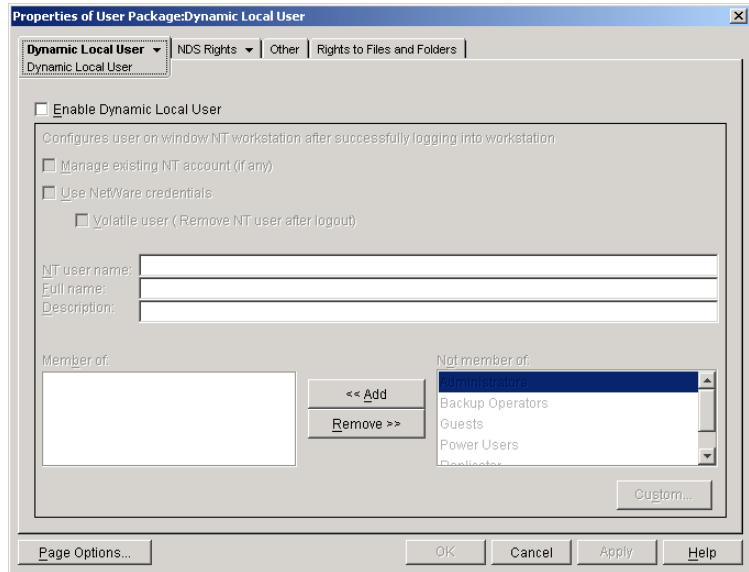
- 5** Continue with the next section, [Configuring the DLU Policy](#).

Configuring the DLU Policy

- 1 If the User Package object's property page is not open, right-click the User Package > click Properties.
- 2 Click the Policies tab > click WinNT-2000 to display the WinNT-2000 policies page.



- 3 Click the check box to the left of the Dynamic Local User Policy to enable the policy, then click Properties to display the Dynamic Local Users property page.



4 Configure the following fields:

Enable Dynamic Local User: Select this option to enable dynamic creation of local users.

Manage Existing NT Account (if any): If you want the DLU policy applied to any existing local user accounts (so that they become dynamic accounts), select this option.

Use NetWare Credentials: Select this option so that eDirectory usernames and passwords are used when creating the local user accounts.

Volatile Users: Select this option so that local user accounts will be removed after the user is logged out of the terminal server.

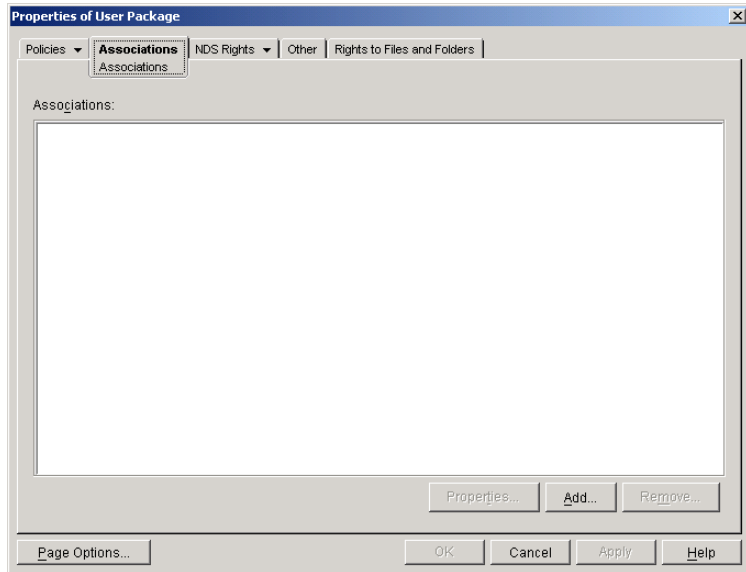
Member of: Select the groups that you want users made members of.

5 Click OK to save your changes and close the Dynamic Local Users property page.

6 Continue with the next section, [Associating the User Package with Users](#).

Associating the User Package with Users

- 1 If the User Package object's property page is not open, right-click the User Package > click Properties.
- 2 Click the Associations tab to display the Associations page.



- 3 Click Add, then browse to and select the users who you want the DLU policy applied to.
- 4 When you've finished adding users, click OK to save your information.
The DLU policy is now configured and will be applied to any users associated with the User Package.

A Note About the ZENworks for Desktops 3.2 Windows Terminal Server Policy

The Users Policy Package also includes a Windows Terminal Server policy. If you are using this policy to control users' terminal server connection or login settings, you need to ensure that the Inherit Client Configuration option is turned on. This is on the policy's Login page (User Package object > Windows

Terminal Server Policy properties > Terminal Configuration tab > Login page). Otherwise, the policy will cause the session to open to the terminal server's desktop rather than the launched application.

6

Publishing Applications

To enable an application to be published through Novell® DeFrame™, complete the tasks in the following sections:

- ♦ “Creating the Application Object in Novell eDirectory” on page 63
- ♦ “Configuring the Application Object” on page 66
- ♦ “Distributing the Application through OnDemand Services” on page 75
- ♦ “Distributing the Application through ZENworks for Desktops” on page 76

Creating the Application Object in Novell eDirectory

To publish a thin-client application to users, you need to create an Application object in Novell eDirectory™. You can either create the Application object manually, or, if you are using Citrix MetaFrame terminal servers, you can use the DeFrame Published Application Synchronization utility to import the application information from the Citrix Published Application Manager utility into eDirectory. The following sections provide instructions for both methods:

- ♦ “Manually Creating the Application Object” on page 64
- ♦ “Using the DeFrame Published Application Synchronization Utility to Create the Application Object” on page 65

Manually Creating the Application Object

- 1** In ConsoleOne[®], right-click the container where you want to create the Application object > click New > click Object to display the New Object dialog box.
- 2** In the New Object list, select App:Application, then click OK to display the New Application dialog box.
- 3** Select Create a Simple Application Object (No .AOT/.AXT or .MSI File), then click Next.
- 4** Fill in the following fields:

Object Name: Enter a name for the Application object. This name must be unique within the container where you are creating it. Make sure the name is 30 or fewer characters. 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)	/ (forward slash)
, (comma)	# (pound sign)

Path to Executable File: Leave this field empty. You will provide this information in a different location.

Define Additional Properties: Make sure this option is selected.

- 5** Click Finish to create the Application object and display the Application object's property pages.
- 6** Leave the Application object's property pages open and continue with [“Configuring the Application Object” on page 66.](#)

Using the DeFrame Published Application Synchronization Utility to Create the Application Object

If you are using the Citrix Published Application Manager utility to publish applications from Citrix MetaFrame servers, you can use the DeFrame Published Application Synchronization utility to import the applications into eDirectory as Application objects. For each application imported, the Published Application Synchronization utility will create an Application object and configure it with the ICA client settings (colors, window size, sound, etc.) established in the Citrix Published Application Manager utility.

You must run the Published Application Synchronization utility on any Citrix MetaFrame server whose applications you want created as Application objects in eDirectory. If an application resides on two different terminal servers, only one Application object will be created; you will need to configure the single Application object to recognize that the application is hosted on multiple servers (see [“Configuring the Application Object” on page 66](#)).

To run the utility:

- 1** Make sure the utility’s context.ini file contains the eDirectory context in which you want to create the Application objects.

The context.ini file is located in the terminal server’s c:\winnt\system32\deframe directory. Under the [Applications] heading, enter the context where you want the Application objects created. For example:

```
.farm1.deframe.services.novell
```

The Published Application Synchronization utility cannot distinguish between different Citrix server farms. Because of this, you need to ensure that Application objects created for one farm are in a different eDirectory context than Application objects created for another farm. Otherwise, the service will overwrite one farm’s Application objects with another farm’s Application objects. For example, the following contexts could be used for three different farms:

```
.farm1.deframe.services.novell  
.farm2.deframe.services.novell  
.farm3.deframe.services.novell
```

- 2** Make sure you are authenticated to eDirectory as user who has rights to create objects in the context listed in the context.ini file.

- 3** Launch the utility by running PassSrv.exe from the c:\winnt\system32\deframe directory.

The utility does not have an interface. When it is finished, it creates a log file, PassSrvLog.txt, that records whether or not Application object creation was successful or failed. The log file is created in the same directory as PassSrv.exe.

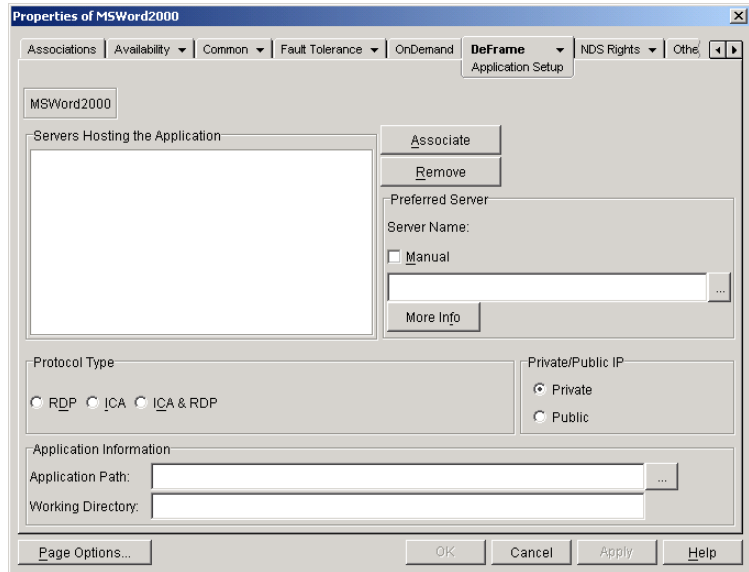
IMPORTANT: If you make changes to an application's information in the Citrix Published Applications Manager utility after the application has been imported into eDirectory, you will need to either 1) delete the Application object in eDirectory and reimport the application or 2) make the same changes to settings in the Application object.

- 4** Repeat **Step 1** through **Step 3** on each Citrix MetaFrame terminal server whose applications you want to import into eDirectory.
- 5** Continue with the next section, **Configuring the Application Object**.

Configuring the Application Object

After you've created the Application object, you need to make sure it is configured correctly.

- 1** If the Application object's property pages are not currently displayed, right-click the object > click Properties.
- 2** Click the DeFrame tab > click Application Setup to display the Application Setup page.



3 Fill in the following fields:

Servers Hosting the Application: Use this list to define the terminal servers from which the application can be run. When a user launches the application, the application is run from the least utilized terminal server. To add a terminal server to the list, click Associate, then browse for and select a DeFrame Server object.

IMPORTANT: In order for DeFrame to determine the least utilized terminal server, the terminal server and application must be configured to participate in load balancing. For information, see [“Configuring Load Balancing Services” on page 77](#).

Preferred Server: If the application is hosted on multiple terminal servers and the servers are participating in load balancing, the DeFrame Watchdog Service creates a preferred server list. The list, which is prioritized according to server utilization (with the least-utilized server receiving the highest priority), is used to determine the server that will run the application the next time it is launched.

The preferred server list is not shown on this page, but is displayed under the `deframeAppPSList` attribute on the Other tab. Each server is listed by its distinguished name and priority. For example:

```
ts1.deframe.server.novell01
ts2.deframe.server.novell00
```

00 is the highest priority (least-utilized server), 01 is the second highest, and so forth.

You need to use this Preferred Server field only if the application will be run from one terminal server, or if you want to force the application to run from one terminal server. In that case, click Manual to indicate a manual assignment > add the preferred terminal server to the Server Name field by browsing for and selecting the appropriate DeFrame Server object.

To see a manually-assigned preferred server's address information, click More Info.

Protocol Type: Select the protocol (RDP, ICA, or ICA & RDP) for the client session in which the application will run. The application's protocol setting overrides the user's protocol setting (User object > DeFrame tab > User Settings page) and is used as long as the server supports the protocol. The following table provides a summary.

Server's Protocols	Application's Protocol Setting	User's Protocol Setting	Protocol Used for Client Session
RDP	RDP	RDP	RDP
RDP	RDP	ICA	RDP
RDP	RDP	ICA & RDP	RDP
RDP	ICA	RDP	RDP
RDP	ICA	ICA	RDP
RDP	ICA	ICA & RDP	RDP
RDP	ICA & RDP	RDP	RDP
RDP	ICA & RDP	ICA	RDP
RDP	ICA & RDP	ICA & RDP	RDP
ICA & RDP	RDP	RDP	RDP
ICA & RDP	RDP	ICA	RDP

Server's Protocols	Application's Protocol Setting	User's Protocol Setting	Protocol Used for Client Session
ICA & RDP	RDP	ICA & RDP	RDP
ICA & RDP	ICA	RDP	ICA
ICA & RDP	ICA	ICA	ICA
ICA & RDP	ICA	ICA & RDP	ICA
ICA & RDP	ICA & RDP	RDP	RDP
ICA & RDP	ICA & RDP	ICA	ICA
ICA& RDP	ICA & RDP	ICA & RDP	ICA

Private/Public IP: Select whether the terminal server's private or public IP address (DeFrame Server object > DeFrame Server Environment tab > Environment page) will be used when launching the application. If you select Private, the application will only be available to users inside your firewall. If you select Public, the application will only be available to users outside your firewall.

If you need the application to be available both inside your firewall and outside, you will need to create two Application objects, one for private access and one for public access.

Application Path: Enter the path to the application's executable file. If you are running ConsoleOne on the terminal server, you can browse for and select the file. If you've added multiple terminal servers to the Servers Hosting the Application list, the application's executable must be in the same location on each of the terminal servers.

Working Directory: The working directory defaults to the directory where the application resides. If necessary, you can change the directory.

- 4 Leave the Application object's property pages open and continue with the next section, [Configuring ICA Client Settings](#).

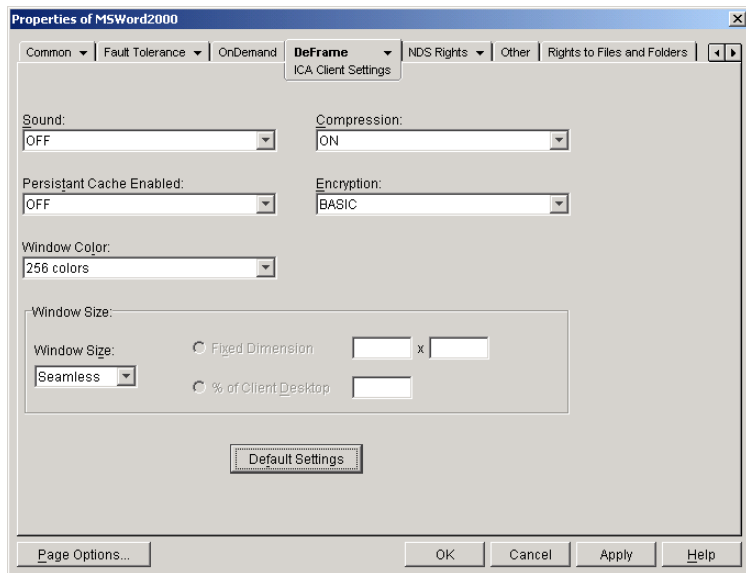
Configuring ICA Client Settings

This section applies only if you've set the application's protocol type to ICA or ICA & RDP (DeFrame tab > Application Setup page).

You can use the ICA Client Settings page to determine how the ICA client will handle the application. This includes window color, window size, encryption, sound, cache, and compression.

To configure the ICA client settings:

- 1 Click the DeFrame tab > click ICA Client Settings.



- 2 Configure the following settings, or click Default Settings to reset the settings to their predefined values:

Sound: Select ON to enable sound for the session, or select OFF to disable sound. Enabling sound increases the terminal server load and requires increased bandwidth for transmitting the sound data. The default is OFF.

Compression: Select ON to enable data compression, or select OFF to disable compression. Enabling data compression can reduce the bandwidth required to transmit the data. The default is ON.

Persistent Cache: Select ON to enable caching of frequently-used screen elements to the user's local drive, or select OFF to disable caching. Creating a persistent cache can reduce the bandwidth required to transmit reusable data. The default is OFF.

Encryption: Select the minimum encryption level for data transmissions. The default is BASIC. If your Citrix MetaFrame servers support it, you can use RC5 encryption with 40-bit, 56-bit, or 128-bit keys.

IMPORTANT: If you use an encryption level other than BASIC, contextless login to the terminal server will be disabled. Users will be prompted to enter their username and password whenever they launch the application.

Window Color: Select the window color display setting for the session. You can select 16 colors, 256 colors, high color (16 bits), or true color (32 bits). The default is 256 colors.

Window Size: Select the window size for the session. You can select a predefined window size (640x480, 800x600, 1024x768, 1280x1024, 1600x1200), specify a custom size based on a fixed dimension or percentage of the user's desktop, or specify a seamless window. A seamless window makes the application look like a local application to the user rather than a remote application and enables the user to use local function keys, switch between applications on the task bar, and tile local and remote windows. The default is seamless window.

- 3 When finished, continue with the next section, [Configuring File System Access Rights](#).

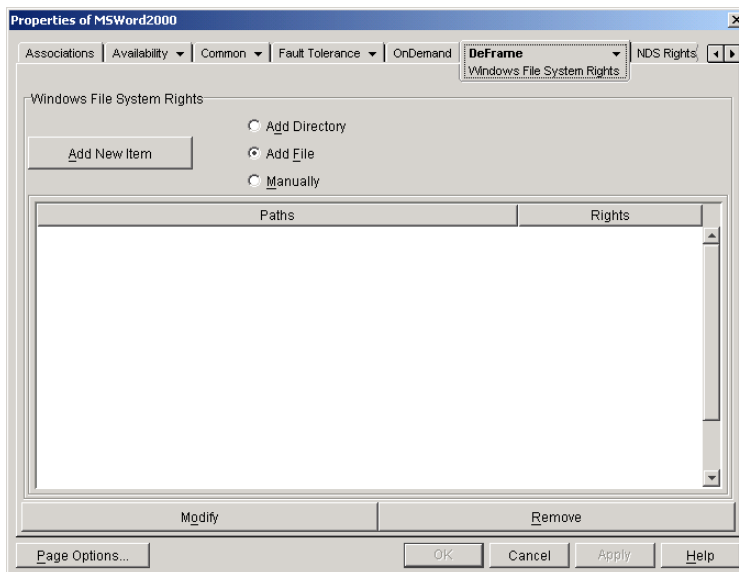
Configuring File System Access Rights

When a user launches the application, DeFrame creates a local user profile on the terminal server. As part of this process, the user is added to a local group and receives the file system rights assigned to that local group. If you are using the DeFrame Dynamic User (DDU) functionality, the user is added to the Everyone group. If you are using ZENworks for Desktops Dynamic Local User (DLU) policy, the user is added to the group defined in his or her policy.

If, when running the application, users will require file system rights in addition to the rights granted to the local group, you need to manually configure these rights. If you don't need to assign rights, skip to the next section, [“Enabling Usage Tracking” on page 74](#).

To configure file system rights:

- 1 Click the DeFrame tab > click Windows File System Rights.



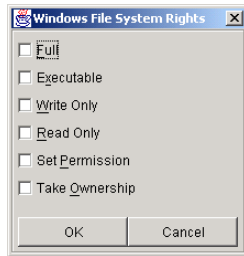
- 2 Select one of the following options:
 - ◆ **Add Directory:** Lets you select a directory and assign rights.
 - ◆ **Add File:** Lets you select a file and assign rights.
 - ◆ **Add Manually:** Lets you enter a file or directory and assign rights. Use this option if the file or directory is not on the local system or you do not have a drive mapped to it. Make sure you enter the path from the perspective of the terminal server (for example, C:\MSWORD2000).
- 3 If you selected Add Directory, click Add New Item, use the Add Directory dialog box to select the directory, then click OK to display the Windows File System Rights dialog box.

or

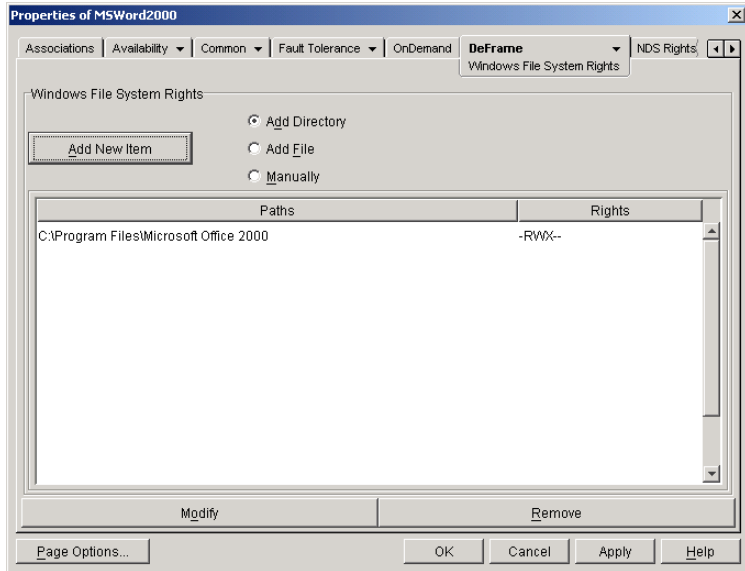
If you selected Add File, click Add New Item, use the Open dialog box to select the file, then click Open to display the Windows File System Rights dialog box.

or

If you selected Add Manually, click Add New Item, enter the file or folder path, then click OK to display the Windows File System Rights dialog box.



- 4** Select the rights for the folder or file, then click OK to add the file or folder to the list.



- 5 If necessary, repeat **Step 2** through **Step 4** to assign any additional rights users will need to run the application.
- 6 After you've finished assigning rights, click OK to save your information.
- 7 Make sure the DeFrame Access Control service (Novell DeACL Service) is running on all terminal servers where this application resides. For information about starting the service, see "**Starting the DeFrame Services**" on page 33.
- 8 Continue with the next section, **Enabling Usage Tracking**.

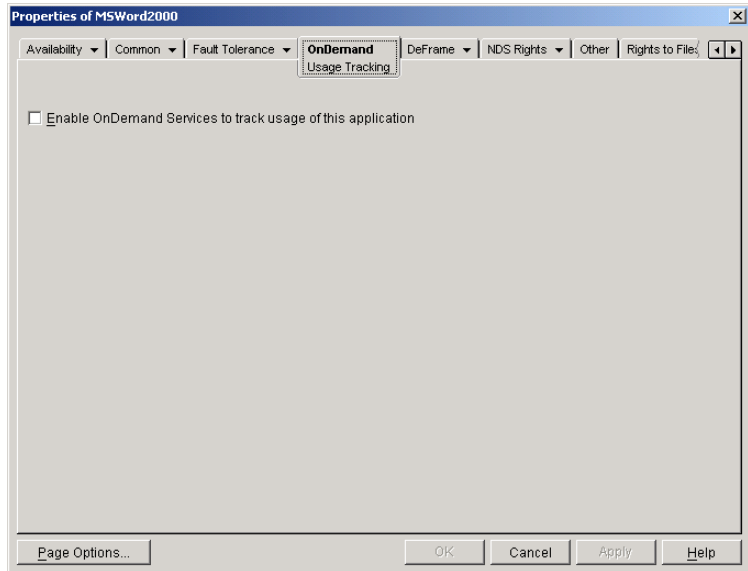
Enabling Usage Tracking

If the application will be delivered through an OnDemand Services package that includes usage charges, you need to enable the application's usage to be tracked.

If the application will be delivered through Novell Application Launcher™ in ZENworks® for Desktops, do not enable usage tracking. Usage tracking of thin-client applications is not supported in Novell Application Launcher and will result in the user being unable to launch the application.

To enable the application's usage to be tracked:

- 1 Click the OnDemand tab to display the Usage Tracking page.



- 2 Turn on the Enable OnDemand Services to Track Usage of This Application option.
- 3 Click OK to save your changes.
- 4 Continue with then next section, [Distributing the Application through OnDemand Services](#), or with “[Distributing the Application through ZENworks for Desktops](#)” on page 76.

Distributing the Application through OnDemand Services

To make the application available to users through OnDemand Services, you need to include the application in an OnDemand Services package. For information, see [Creating Application Packages](#) in the *OnDemand Services 2 Installation and Administration* guide.

Distributing the Application through ZENworks for Desktops

To make the application available to users through ZENworks for Desktops, you can associate the Application object with User, Group, or container objects. Users will then be able to launch the application from Novell Application Launcher. For information, see the ZENworks for Desktops 3.2 documentation at the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

Because associating an Application object with users eliminates the OnDemand Services purchase and usage tracking capabilities for that application, you might want to first review the information in [Creating Application Packages](#) in the OnDemand Services 2 *Installation and Administration* guide.

7

Configuring Load Balancing Services

If you have more than one terminal server, you can use Novell® DeFrame™ Load Balancing Services to more evenly distribute the workload among the servers. When a user launches an application that resides on multiple servers, Load Balancing Services evaluates the current load of each server and runs the application from the least-utilized server. This provides users with the best application performance while maximizing the efficiency of your terminal servers.

To configure Load Balancing Services, complete the tasks in the following sections:

- ♦ [“How Load Balancing Works” on page 77](#)
- ♦ [“Starting a Watchdog Service” on page 78](#)
- ♦ [“Assigning Applications to a Watchdog Service” on page 79](#)
- ♦ [“Adjusting the Load Calculation Weights” on page 81](#)
- ♦ [“Modifying the Watchdog Service Configuration Settings” on page 85](#)

How Load Balancing Works

DeFrame load balancing is performed by the Watchdog service (Novell Dewatch Balancing Service), a Windows service that you must run on each terminal server that will participate in load balancing. The Watchdog service performs two main tasks:

- ♦ It calculates the current workload of the terminal server. The calculation is based on the utilization of the server’s resources (memory, CPU usage,

and so forth) and the weight, or importance, assigned to each resource. You can adjust the resource weights on a per-server basis to account for different performance capabilities (see [“Adjusting the Load Calculation Weights” on page 81](#)).

- ◆ If it is designated as the primary Watchdog service for an application, it collects the workload statistics for each terminal server hosting the application and creates a preferred server list. The preferred server list is prioritized with the least-utilized server having the highest priority and the most-utilized server having the lowest priority. The application will be launched from the least-utilized server, if it is available.

You can have one Watchdog service designated as the primary service for all applications, in which case it will determine the preferred servers for all of the applications. Or you can designate multiple primary Watchdog services, each with responsibility to monitor a different group of applications.

The number of applications you assign to a primary Watchdog service depends on several factors, including the workload that the Watchdog service’s server can handle and the speed of the Watchdog service’s access to Novell eDirectory™.

Each primary Watchdog service can also have a backup Watchdog service. The backup Watchdog service can be any other Watchdog service, including primary or backup services for other applications. The backup Watchdog service becomes the primary Watchdog service only if the primary service becomes unavailable. At that point, the two services switch roles and remain that way until the new primary service becomes unavailable or you manually switch them using ConsoleOne®.

For instructions on designating a Watchdog service as the primary service or backup service for an application, see [“Assigning Applications to a Watchdog Service” on page 79](#)

Starting a Watchdog Service

The Watchdog service must be running on each terminal server that will participate in load balancing.

- 1** Click the Start menu > Settings > Control Panel > Administrative Tools > Services.
- 2** Double-click Novell DeWatch Balancing Service.

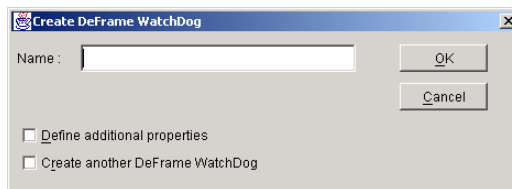
- 3** In the Startup Type field, select Automatic.
- 4** Click Start to immediately start the service.
- 5** Click OK.
- 6** Continue with the next section, “[Assigning Applications to a Watchdog Service](#)” on page 79, to select the applications that the Watchdog service will monitor.
- 7** If you want to change the default weight settings used by the Load Balancing service to calculate server load, see “[Adjusting the Load Calculation Weights](#)” on page 81.

Assigning Applications to a Watchdog Service

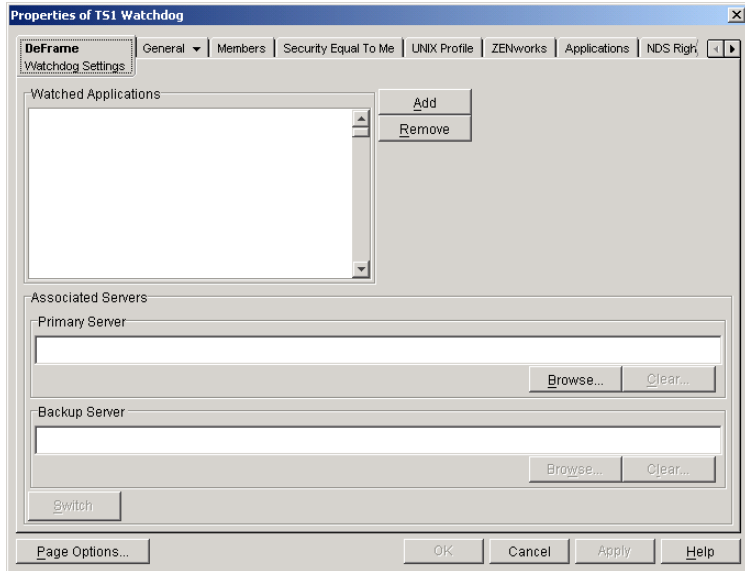
In order for a Watchdog service to generate the primary server list for an application, you must assign the application to the Watchdog service. You do this by creating a Watchdog object in eDirectory and associating the application with the Watchdog object.

To create a Watchdog object and associate applications with it:

- 1** In ConsoleOne, right-click the container where you want to create the Watchdog object > click New > click Object to display the New Object dialog box.
- 2** In the Class list, select DeFrameWatchDog, then click OK to display the Create DeFrame WatchDog dialog box.



- 3** In the Name field, enter a name for the Watchdog object.
- 4** Select Define Additional Properties, then click OK to create the Watchdog object and display the object’s property pages.



5 In the Watched Applications list, add the applications you want monitored by the Watchdog service. To do so, click Add, select the appropriate Application object, then click OK.

6 In the Primary Server box, click Browse, then select the terminal server (DeFrame Server object) where the Watchdog service resides.

This Watchdog service becomes the primary service for the applications displayed in the Watched Applications list.

7 (Optional) In the Backup Server box, select the terminal server (DeFrame Server object) where the backup Watchdog service resides.

The backup Watchdog service will only be used if the primary Watchdog service becomes unavailable or if you click the Switch button to make the backup service the primary service.

8 Click OK to save your information.

Adjusting the Load Calculation Weights

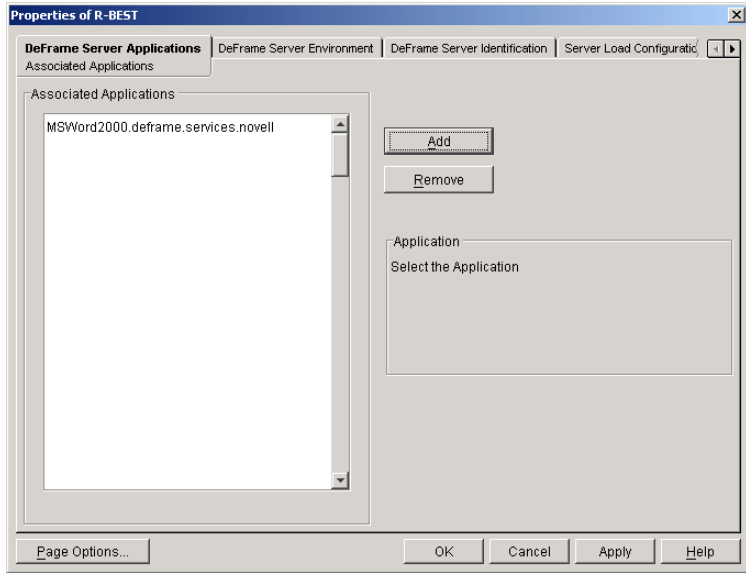
There are five server resources that are used to calculate a server's load: memory, CPU usage, users, page file, and virtual memory. The Watchdog service automatically calculates the load for each of these resources from information obtained from the server. The weights assigned to these individual resource loads are then factored into the calculation to determine the overall server load.

You can use the default weight settings for each resource, or you can modify the weights on a per-server basis to account for different performance capabilities. The weight of a resource determines how much influence its load has on the server load score relative to the other resource loads. For example, if you set the memory weight to 100, the CPU usage weight to 100, and the remaining resource weights to 0, the memory and CPU usage loads exert equal influence (50%) on the server load score and the other resource loads are ignored. If you set all five resource weights to 100, their loads each have equal influence (20%) on the server load score.

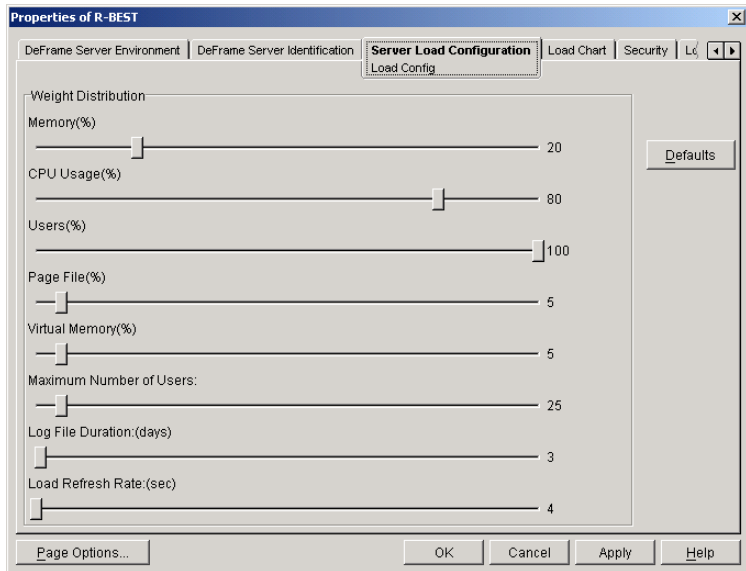
For a detailed explanation of how the server load score is calculated, see [“Server Load Calculation” on page 84](#).

To adjust the weight assigned to a resource:

- 1 In ConsoleOne, right-click the terminal server object (DeFrame Server object) > click Properties to display the object's property pages.



2 Click the Server Load Configuration tab to display the Load Config page.



3 Modify the following settings:

Memory, CPU Usage, Users, Page File, Virtual Memory: Select the weight you want applied to each resource load. A higher weight in relationship to the other resources places more importance on that resource's load when calculating the server load score. For example, if the server has lots of memory, you might want to set a high weight for memory. Doing so would result in a lower server load score when much of the server's memory is available (its memory load is low). A low server load score increases the likelihood that the server would become the preferred server for an application.

Maximum Number of Users: All resource loads are calculated from information obtained from the server, with the exception of the Users load. The Users load is calculated by dividing the number of currently logged-in users (obtained from the server) by the maximum number of users specified by this setting. Set this number equal to the maximum number of concurrent sessions you expect the terminal server to handle.

Log File Duration: Select the number of days after which the DeFrame Watchdog Service will purge information from its log file. The log file, named DEWATCHLOG.TXT, is located in the server's C:\WINNT\SYSTEM32\DEFRAME directory. The default is 3 days, which means the DeFrame Watchdog Service purges the information from the log file every 3 days.

Load Refresh Rate: This setting applies only if the server is functioning as a primary Watchdog service. Select how often the Watchdog service requests load information from other Watchdog services. The default is every 4 seconds.

- 4** If you want to reset all settings to the default configuration, click Defaults.

The default configuration is stored in the DEFAULT_REG object at the root of the eDirectory tree. You can modify the default configuration. The modifications you make will be applied to all terminal servers that are using the default configuration.

- 5** Click OK to save your changes.

The Watchdog service will begin using the new resource weights in 60 seconds. You can restart the Watchdog service to force it to immediately use the new weights.

Server Load Calculation

To calculate the server load score, the Watchdog service uses the following formula:

$$\text{server load score} = (\text{resource 1 load score}) + \dots + (\text{resource 5 load score})$$

where *resource 1 load score* through *resource 5 load score* are the load scores calculated for memory, CPU usage, users, page file, and virtual memory. The following formula is used for that calculation:

$$\text{resource load score} = (100)(\text{adjusted weight})(\text{resource used}/\text{resource total})$$

The (*resource used/resource total*) ratio is calculated using values obtained from the server, with the exception of the users resource ratio (*users logged in/maximum users allowed*). For this ratio, the maximum users allowed value is determined by the Maximum Number of Users setting on the Load Configuration page.

The resource's adjusted weight is calculated from the following formula:

$$\text{adjusted weight} = (\text{resource weight}) / (\text{total of all resource weights})$$

The adjusted weight is required because each resource can be assigned a weight between 0 and 100 percent. However, because the total weight for all five resources cannot exceed 100%, the weight value for each resource is internally adjusted so that the resources' total weight equals 100%. For example, substituting the default weights into the formula gives the following adjusted weights (totaling 100%):

Memory: $20 / (20+80+20+5+5) = .154$ (15.4%)
CPU Usage: $80 / (20+80+20+5+5) = .614$ (61.4%)
Users: $20 / (20+80+20+5+5) = .154$ (15.4%)
Page File: $5 / (20+80+20+5+5) = .038$ (3.8%)
Virtual Memory: $5 / (20+80+20+5+5) = .038$ (3.8%)

Using these adjusted weights and sample values for the (*resources used/resources available*) ratios, the resource load scores are:

Memory load score: $(100)(.154)(.75) = 11.55$
CPU Usage load score: $(100)(.614)(.27) = 16.58$
Users load score: $(100)(.154)(.4) = 6.16$
Page File load score: $(100)(.038)(.72) = 2.74$
Virtual Memory load score: $(100)(.038)(.2) = .76$

Plugging these resource load scores into the server load score formula gives the following result:

$$11.55 + 16.58 + 6.16 + 2.74 + .76 = 37.79$$

If the server load score is less than 100, the server is included in load balancing and can become an application's preferred server, provided its load score is less than any other server hosting the application. If the load score is 100 or greater, the server is excluded from load balancing until its score drops below 100.

Modifying the Watchdog Service Configuration Settings

You can use the Watchdog service's configuration settings to provide advanced control of the service, including:

- ◆ The amount of detail that is output to the service's log file.
- ◆ How often the service, when functioning as a backup for another Watchdog service, checks to see if the primary Watchdog service is alive.
- ◆ How often the service, when functioning as a primary service, requests load information from other Watchdog services.
- ◆ How often the service, when functioning as a primary service, waits for load information to be returned from other Watchdog services before timing out.

The configuration settings are stored in the terminal server's Windows registry under the following key:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\DeWatch\Config
```

The following table describes the configuration settings (values).

Value	Type	Default	Description
SettingsRecacheDelaySecs	DWORD	20	Determines how often the Watchdog service rereads these registry settings.

Value	Type	Default	Description
LogFileOutputFilter	DWORD	0	<p>Determines the amount of information that is output to the logfile. The valid settings are:</p> <p>0 = errors 1 = errors and warnings 2 = all information</p> <p>The logfile, dewatchlog.txt, is located in the c:\winnt\system32\deframe directory.</p>
LogFileConvertUni	DWORD	0	<p>The default setting (0) causes the Watchdog service to generate a Unicode logfile. If you want the Watchdog service to generate an ansi text file, change the setting to 1.</p> <p>0 = no 1 = yes</p>
LogGetLastError	DWORD	0	<p>The default setting (0) instructs the Watchdog service to output its error code and message only. If you want the Watchdog service to output additional Windows error information, change the setting to 1.</p> <p>0 = no 1 = yes</p>
AliveCheckDelaySecs	DWORD	30	<p>Determines how often the Watchdog service, when functioning as a backup service, checks to see if the primary Watchdog service is still available.</p>

Value	Type	Default	Description
AliveCheckTimeoutSecs	DWORD	10	Determines how long the Watchdog service, when functioning as a backup service, waits for the primary Watchdog service to respond to its alive check. If the primary service does not respond within the specified amount of time, the backup service switches roles with the primary service.
DSRecacheDelaySecs	DWORD	30	Determines how often the Watchdog services reads eDirectory for updated information. If the Watchdog service is a primary service, it reads eDirectory to refresh the list of applications assigned to it. If the Watchdog service is not a primary service, it reads eDirectory to see whether it has been assigned primary or backup status.

Value	Type	Default	Description
LoadRequestTimeoutSecs	DWORD	10	<p>Determines how long the Watchdog service, when functioning as a primary service, waits for other Watchdog services to return server load information. If a service's response is not received within the specified amount of time, the service's previous load information is used.</p> <p>NOTE: By default, a primary service requests server load information every 4 seconds. You can change this setting in eDirectory (DeFrame server object > Server Load Configuration tab > Load Configuration tab > Load Config page > Load Refresh Rate field).</p>
SocketSendTimeoutMillis	DWORD	100	Determines how long the Watchdog service will retry sending information through a socket before timing out.
PacketProcessTimeoutMillis	DWORD	100	Determines how long the Watchdog service will hold a socket or IPC waiting for incoming data.

8

Managing Disconnected Sessions

When a user disconnects from a terminal server client session instead of exiting the session, the session's status is changed from Active to Disconnected. The user can then reconnect to the existing session at a later time.

The following sections contain information to help you best manage disconnected sessions for Novell® DeFrame™ thin-client applications:

- ♦ “DeFrame Disconnected Session Tracking Service” on page 89
- ♦ “Application Usage Tracking” on page 90

DeFrame Disconnected Session Tracking Service

The DeFrame Disconnected Session Tracking service tracks disconnected sessions in order to reactivate a disconnected session whose session information (user, application, and so forth) matches the information for a newly-opened session. The newly-opened session is then removed. By tracking and reactivating disconnected sessions whenever possible, the service keeps new sessions from being opened when they are not necessary and also keeps disconnected sessions from proliferating and wasting session licenses.

The Disconnected Session Tracking service should be run on all terminal servers where DeFrame is installed. The DeFrame Terminal Server Setup program configures it to start when the server starts up. If necessary, you can check the service's current running status by looking for the Novell Disc Session Tracking Service in the Services window.

How it Works

When a user disconnects from a session, the DeFrame session monitor (deshell.exe) records session information in the User object in Novell eDirectory™. At the same time, the Disconnected Session Tracking service (dcstk.exe) records additional session information under the terminal server's DeFrame Server object.

Approximately every 10 seconds, the Disconnected Session Tracking service checks the session information recorded in the two objects and compares it against the server's current sessions to see if the disconnected session still exists or if a new session has been opened that matches the session information in eDirectory. If the disconnected session has been removed (either reset or timed-out), the service removes the session information from the User and DeFrame Service objects. If a new session has been opened that matches the disconnected session information (user, application, and so forth), it removes the new connection and reactivates the disconnected session for the user. The service then removes the disconnected session information from the User and DeFrame Server objects.

Application Usage Tracking

If a user disconnects from an application whose usage is being tracked through OnDemand Services, the usage tracking continues and the user might be unaware that he or she is being charged for the disconnected time.

You can work around this issue by placing a time limit on disconnected sessions so that disconnected sessions are removed after a period of time. However, if some of your thin-client applications are designed to be launched by the user and run in a disconnected session to process information or perform some other unattended tasks, you might not want to enforce a disconnected session time limit.

To set a disconnected session time limit, you can configure the terminal server's connections. This is done by modifying the Session information on a connection's properties in the Terminal Services Configuration utility. For detailed instructions, see your terminal server documentation.

If you are using Novell ZENworks® for Desktops, you can enable a Terminal Server policy (on a User package) and make sure the User package is applied to the users who will run applications from terminal servers. 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 detailed information, see the [ZENworks for Desktops 3.2 documentation \(http://www.novell.com/documentation\)](http://www.novell.com/documentation).

9

Controlling DeFrame Service Logging

You can control the amount of information that the Access Control, Watchdog, and Disconnected Tracking Session services output to log files. The logfiles are written to the `c:\winnt\system32\deframe` directory.

Each service's logging level is configured through a Windows registry value on the service's terminal server.

Access Control Service

- ◆ Logfile name: `aclog.txt`
- ◆ Key: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\DeACL Service\Parameters\Configuration`
- ◆ Value name: `LogTransaction`
- ◆ Value data: `0 | 1 | 2`

0 provides the least logging information and 2 the most. 2 should be used only when troubleshooting problems.

Watchdog Service

- ◆ Logfile name: `dewatchlog.txt`
- ◆ Key: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\DeWatch\Config`
- ◆ Value name: `LogFileOutputFilter`
- ◆ Value data: `0 | 1 | 2`

0 provides the least logging information and 2 the most. 2 should be used only when troubleshooting problems.

Disconnected Session Tracking Service

- ◆ Logfile name: dstk.txt
- ◆ Key: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\DSCTK
- ◆ Value name: LogLevel
- ◆ Value data: 0 | 1 | 2

0 provides the least logging information and 2 the most. 2 should be used only when troubleshooting problems.

10

Setting Up File Storage through Novell iFolder

By integrating Novell® DeFrame™ with Novell iFolder™ 1.x or 2.x, you can enable users to save files from a thin-client session to one or more network storage locations. DeFrame makes the storage locations available when a user launches an application and removes the user's access when he or she exits the application. If you want the storage locations available to users even when they are not using thin-client applications configured through DeFrame, you can install the standard iFolder client to their workstations to provide continual access.

Novell iFolder runs on NetWare® 5.1, NetWare 6, Windows NT 4.0, and Windows 2000 servers. It is included in NetWare 5.1 and 6 and is also available separately. For more information about iFolder, including how to purchase it if you don't already own it, see the [Novell iFolder product information site \(http://www.novell.com/products/ifolder\)](http://www.novell.com/products/ifolder).

The following tasks explain how to set up DeFrame to work with iFolder:

- ◆ “Installing iFolder” on page 96
- ◆ “Creating DeFrame iFolder Server Objects” on page 96
- ◆ “Assigning iFolder Servers to Users” on page 98
- ◆ “Providing Users with Information Needed to Log In to iFolder Servers” on page 99

Installing iFolder

If iFolder is not already installed, install it using the documentation provided with the software or on the [Novell Documentation Web site \(http://www.novell.com/documentation\)](http://www.novell.com/documentation). Keep the following in mind as you install it:

- ◆ If you want users to have access to iFolder servers when they are not running thin-client applications, follow the instructions to have users install the iFolder client. If you only want iFolder servers available when they are using a thin-client application, don't have users install the client.
- ◆ Make sure that all containers where users reside are defined as iFolder search contexts.
- ◆ If users will access iFolder servers located in different Novell eDirectory™ trees, they must have user accounts in each tree and each iFolder server's search contexts must be configured to include the user contexts.

Creating DeFrame iFolder Server Objects

To supply DeFrame with the information it needs to access iFolder servers, you define the iFolder servers as DeFrame iFolder Server objects in eDirectory. You must do this with each iFolder server you want users to have access to.

To create a DeFrame iFolder Server object:

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

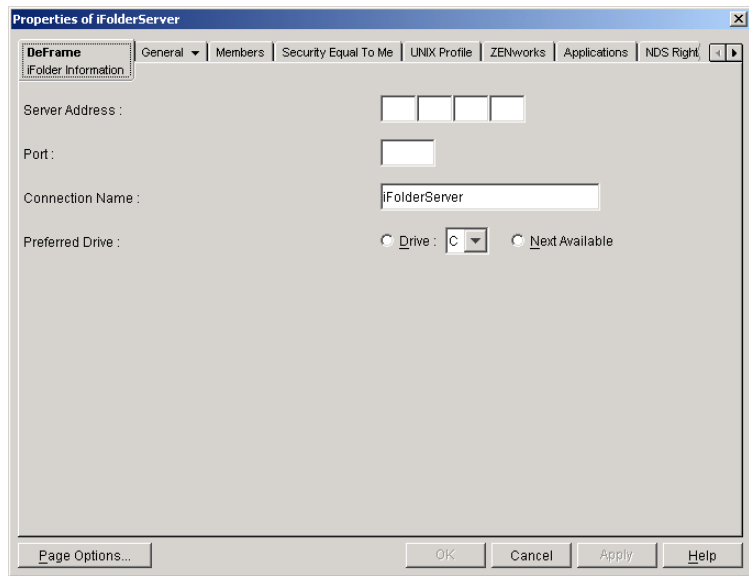
You can create the object in any container in the tree. For example, you could create it in the same container where you created your DeFrame Server objects.

- 2** In the Class list, select deframeIFolder, then click OK to display the Create iFolder Server Object dialog box.

- 3** Enter a name for the object.

The name can be any name that is unique within the container.

- 4 Select Define Additional Properties, then click OK to create the DeFrame iFolder Server object and display its property pages.



- 5 Fill in the following fields:

Server Address: Enter the IP address of the iFolder server.

Port: Enter the port number assigned to the iFolder server. The iFolder Installation program defaults to port 80. If you are not sure what port the iFolder server is using, use the iFolder Server Management Console to verify the port number.

Connection Name: Enter the name you want to use for the iFolder server connection. This name is used when creating a user's drive mapping to the iFolder server. If you don't enter a connection name, the DeFrame iFolder Server object's name is used.

Preferred Drive: When a user connects to the iFolder server, iFolder maps a drive on the user's workstation to the iFolder server (using the name in the Connection Name field). Select the drive that you want mapped to the iFolder server, or select Next Available if you want iFolder to map the next available drive. If you select a specific drive and it is

already in use, iFolder will map the next available drive. The drive is mapped using the following syntax:

```
connection_name on 'NetDrive-username' (drive_letter:)
```

- 6** Click OK to save your changes.
- 7** Repeat the above steps to create DeFrame iFolder Server objects for any additional iFolder servers.

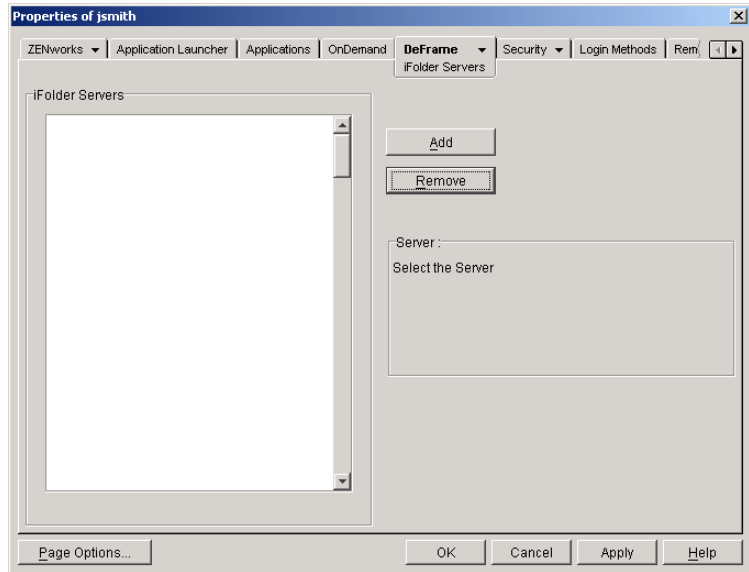
Assigning iFolder Servers to Users

Before DeFrame will enable a user to log in to an iFolder server, you must assign the iFolder server to the user. There are several ways you can do this:

- ◆ You can use ConsoleOne[®] to manually assign iFolder servers to DeFrame users. Each user must be configured individually. See the steps below.
- ◆ You can use the DeFrame User Configuration utility to configure multiple users at the same time. See [Appendix A, “DeFrame User Configuration Utility,” on page 109](#).
- ◆ You can configure the OnDemand Services User Administration gadget to define a user as a DeFrame user and populate the DeFrame user settings (including the iFolder servers setting) when it creates the user in eDirectory. For information, see [User Administration Gadget \(OD_UserAdminGadget\)](#) in [Gadget Configuration Settings](#) in the OnDemand Services 2 *Installation and Administration* guide.
- ◆ You can configure the OnDemand Services Launch Item gadget to define a non-DeFrame user as a DeFrame user and populate the DeFrame user settings (including the iFolder servers setting) the first time the user launches a thin-client application. For information, see [Launch Item Gadget \(OD_LaunchItemGadget\)](#) in [Gadget Configuration Settings](#) in the OnDemand Services 2 *Installation and Administration* guide.

To use ConsoleOne to assign iFolder servers to a DeFrame user:

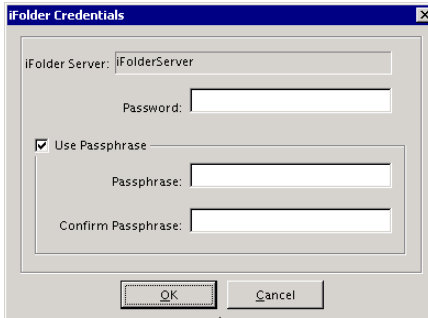
- 1** In ConsoleOne, right-click the User object > click Properties to display the User object’s property pages.
- 2** Click the DeFrame tab, then click iFolder Servers to display the iFolder Servers page.



- 3** Click Add, browse for and select a DeFrame iFolder Server object, then click OK to add it to the iFolder Servers list.
- 4** Repeat **Step 3** to add additional iFolder servers.
- 5** Click OK to save your changes.

Providing Users with Information Needed to Log In to iFolder Servers

After a user has been assigned iFolder servers, the first time the user launches a thin-client application he or she will receive the following prompt for each iFolder server.



Password: This must be the user's password for the eDirectory tree where the iFolder server is located.

Use PassPhrase: This option can be selected if the user wants to encrypt the files stored on the iFolder server. If the option is selected the user must enter and confirm a passphrase. The user's passphrase becomes part of the key that is used to encrypt and decrypt the user's files. If the user has already established a passphrase for the iFolder server (because he or she also uses the iFolder client to access the iFolder server outside of thin-client applications), that passphrase must be used here.

You might want to make sure users understand what information they must supply in these fields, and that they will be prompted for this information for each iFolder server that they log in to. They will be prompted for this information the first time they launch a thin-client application, anytime their eDirectory password changes, and anytime an iFolder server connection cannot be made.

11

Setting Up Application Access through a Firewall

If you want to allow access to thin-client applications through a network firewall, complete the tasks in the following sections:

- ♦ [“Opening Ports in the Firewall” on page 101](#)
- ♦ [“Configuring Network Address Translation” on page 102](#)
- ♦ [“Defining Terminal Server Public Addresses in eDirectory” on page 103](#)
- ♦ [“Defining Applications as Public Applications” on page 103](#)

Opening Ports in the Firewall

Open the following firewall ports for sending and receiving TCP/IP packets:

- ♦ **80 and 443:** These standard HTTP and HTTPS ports allow users to log in to the Novell® ZENworks® OnDemand Services™ Web portal to request and launch applications.
- ♦ **3389:** The RDP client and Windows Terminal Services send and receive packets via this port. This is the standard RDP port. If you’ve used a non-standard RDP port, open that port.
- ♦ **1494 (in only):** The ICA client sends packets to Citrix MetaFrame servers through this port. This is the standard ICA port. If you’ve used a non-standard ICA port, open that port. For more information about Citrix firewall requirements, see the Citrix documentation.

- ◆ **1023 and above (out only):** Citrix MetaFrame servers send packets to ICA clients through these ports. For more information about Citrix firewall requirements, see the Citrix documentation.

Configuring Network Address Translation

If the OnDemand Services Web server and Novell DeFrame™ terminal servers use private (internal) addresses, you need to make sure that your firewall's public (external) addresses map to these private addresses.

The best way to ensure that users are directed to the correct private address from a public address is to use one-to-one mappings to map one public address to one private address. If you want to use one-to-many mappings to map one public address to many private addresses, please be aware of the following:

- ◆ **Application Availability:** When you map one public address to multiple terminal server private addresses, make sure that each terminal server contains the same set of applications so that the public address will always resolve to a terminal server that contains the requested application.

To illustrate why this is important, consider the following simplified scenario: An application is hosted on one terminal server only. When a user launches the application, the user's RDP or ICA client uses the public address assigned to the application's terminal server. The public address also maps to other terminal servers that aren't hosting the application. The firewall, using its round-robin or other preconfigured method for resolving public addresses, routes the user to one of the terminal servers that doesn't have the application. The application fails to launch.

To avoid this problem, you can host all applications on all terminal servers. Or, if you haven't already done so, you can divide your terminal servers into farms and make sure each server in a farm hosts the same applications and resolves to the same public address.

- ◆ **Load Balancing Services:** Because a firewall's proxy server uses its own preconfigured method to resolve to its private addresses, if you use one-to-many mappings, DeFrame Load Balancing Services will not ultimately determine which terminal server will be used to run an application. Load Balancing Services will still determine which terminal

server is an application's preferred server and the preferred server's public address will be used. However, there is no guarantee that the proxy server will resolve the public address back to the preferred server.

- ♦ **Disconnected Session Tracking Service:** If you use one-to-many mappings, when a user disconnects from a session, there is no guarantee (for the same reason mentioned above) that the same terminal server will be used when he or she reconnects. If the user is connected to another terminal server, a new session is opened. If the user happens to be reconnected to the original terminal server and the user's disconnected session has not been reset or timed-out, the same session will be used to avoid creating another session. You can resolve the problem of disconnected sessions accumulating on a terminal server by disallowing disconnected sessions.

Defining Terminal Server Public Addresses in eDirectory

After you've set up your firewall to map a terminal server's private address and a public address, you need to add the public address to the terminal server's DeFrame Server object in Novell eDirectory™.

- 1** In ConsoleOne®, right-click the terminal server's DeFrame Server object > click Properties.
- 2** Click the DeFrame Server Environment tab to display the Environment page.
- 3** Enter the public IP address or public DNS name.
- 4** Click OK to save your changes.
- 5** Repeat the above steps for each terminal server.

Defining Applications as Public Applications

When a user launches an application using the OnDemand Services Launch Item gadget, the Launch Item gadget passes either the terminal server's private address or its public address to the DeFrame client so that it can initiate the connection to the terminal server. For the terminal server's public address to be used, you need to define the application as a public application.

IMPORTANT: An application can be accessible through both a public and private address only if you create separate Application objects and configure one as public and one as private. For example, you could create a Microsoft Word (Local) Application object and a Microsoft Word (Remote) Application object and include them in the same OnDemand Services package. The user would then need to launch the appropriate application for his or her location.

To configure an application as a public application:

- 1** In ConsoleOne, right-click the Application object > click Properties.
- 2** Click the DeFrame tab > click Application Setup.
- 3** In the Private/Public IP box, select Public.
- 4** Click OK to save your changes.

12

Removing DeFrame from a Terminal Server

If you no longer want to use Novell® DeFrame™ to manage and deliver a terminal server's applications, you can remove DeFrame from the terminal server. This requires you to uninstall files and remove the DeFrame Server object from Novell eDirectory™. Refer to the following sections for instructions:

- ♦ [“Uninstalling the DeFrame Files” on page 105](#)
- ♦ [“Deleting the DeFrame Server Object” on page 106](#)

Uninstalling the DeFrame Files

When you uninstall the DeFrame files, the DeFrame services are stopped and the files removed. After restarting the server, the DeFrame services will no longer appear in the Services window.

If you previously installed the Novell iFolder™ NetDrive to support access to iFolder servers from the terminal server, you will need to reinstall it after you uninstall the DeFrame files. DeFrame also installs a version of NetDrive. The iFolder NetDrive and the DeFrame NetDrive share common files in the system32 directory. Uninstalling the DeFrame NetDrive removes the shared files so the iFolder NetDrive will not work until it is reinstalled and its site information reconfigured.

To uninstall the DeFrame files:

- 1 At the terminal server, open the Add/Remove dialog box (Start menu > Settings > Control Panel > Add/Remove Programs).

- 2** In the Currently Installed Programs list, select DeServer 2.01, click Change/Remove, then follow the prompts.

Deleting the DeFrame Server Object

Deleting the terminal server's DeFrame Server object from eDirectory disassociates the DeFrame Server object from any Application objects. The result is that the terminal server will no longer be used to run an application launched by a user even if the application is still hosted on the server.

- 1** In ConsoleOne, right-click the terminal server's DeFrame Server object > click Delete NDS Object.
- 2** Click Yes to confirm the deletion.

IV

Appendixes

Appendix A, “DeFrame User Configuration Utility,” on page 109

Appendix B, “Upgrading From DeFrame 2.0,” on page 115

Appendix C, “Documentation Updates,” on page 119

A

DeFrame User Configuration Utility

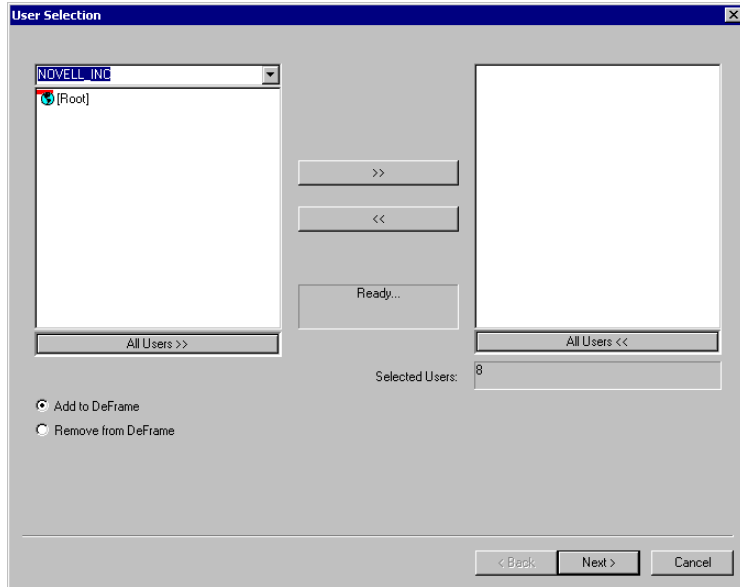
The DeFrame User Configuration utility lets you enable one or more non-DeFrame users as Novell[®] DeFrame[™] users and configure the DeFrame user settings.

You can also use this utility to reconfigure the settings for existing DeFrame users. For example, if you have already enabled users as DeFrame users, you could run the utility to configure them as dynamic users. When doing so, you need to ensure that each DeFrame user setting is configured how you want it for the users. The utility reconfigures all settings to the values you define during the current session.



To use the DeFrame User Configuration utility:

- 1** At a terminal server where you've installed DeFrame, make sure you are authenticated to Novell eDirectory[™] and have Admin-equivalent rights.
- 2** Run `glbalias.exe`, located in the `c:\winnt\system32\deframe` directory, to start the User Configuration utility and display the User Selection dialog box.

You use this dialog box to select the users you want to configure.

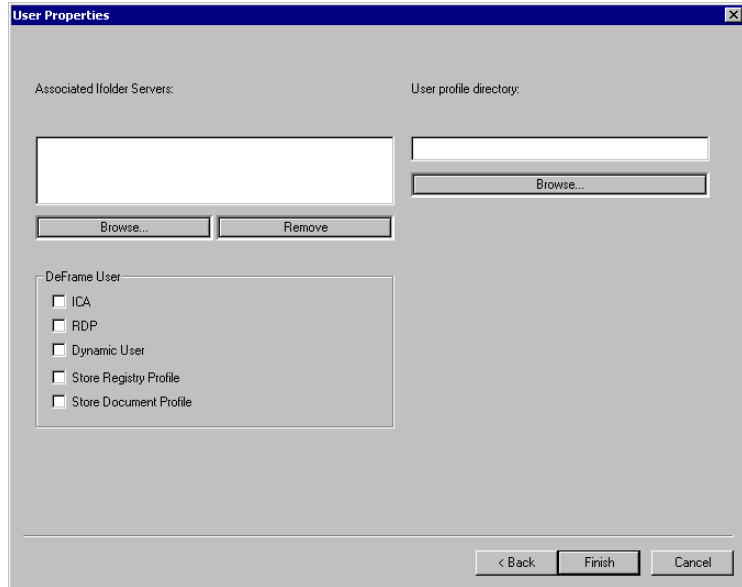


- 3** In the list of eDirectory trees, make sure the correct eDirectory tree is selected.
- 4** In the left pane, select a user you want to configure, then click the >> button to add the user to the right pane.

You can select User objects or container objects. If you select a container object, all users in the container and its subcontainers are added to the list in the right pane. A non-DeFrame user is indicated by a normal User icon  and an existing DeFrame user is indicated by a yellow-shaded User icon .

- 5** When you've added all users who you want to configure, click Next to display the User Properties dialog box.

You use this dialog box to configure the DeFrame user settings. The settings you choose will be applied to all the users you selected in the previous dialog box.



6 Configure the following settings:

Associated iFolder Servers: If you've installed Novell iFolder™ 1.x to enable network file storage, click Browse to select the iFolder Server objects to associate with the users.

ICA Client: Select this option to force applications that are configured to run in either an ICA or RDP client session to run in an ICA client session on the user's workstation. The application's terminal server must support the ICA protocol (in other words, the Windows server must have Citrix MetaFrame installed). Otherwise, an RDP client session will be used.

RDP Client: Select this option to force applications that are configured to run in either an ICA or RDP client session to run in an RDP client session on the user's workstation.

If you select both ICA Client and RDP Client, and an application is configured to run in either type of session, an ICA client session will be used. For a summary showing the protocol (RDP or ICA) depending on the user's Client setting, an application's protocol setting, and a server's supported protocols, see [“Configuring the Application Object” on page 66](#).

Dynamic User: Select this option to have DeFrame use its DeFrame Dynamic User (DDU) functionality to dynamically create the user's local account each time the user launches an application from the server. DeFrame creates the user's account, using his or her Novell eDirectory username, when the user starts the application. After the user exits the application, DeFrame deletes the user account. If the terminal server is a member of an NT domain and the Domain Controller server name is included in the terminal server's Domain field (DeFrame Server object > DeFrame Server Environment tab > Environment page) and the user's eDirectory username and password matches the user's NT domain username and password, the local user account will be created using the NT domain account information.

IMPORTANT: If a user account already exists that matches the user's eDirectory username, DeFrame will use the existing account rather than create a new account. When the user exits the application, the account will be deleted, even though DeFrame did not create the account. If you have persistent accounts (local or domain) that you want to remain persistent, you should not use DDU.

If you don't enable this option, you need to ensure that the user has a permanent local or domain user account, or you need to use ZENworks for Desktops 3.2 Dynamic Local User (DLU) policies to create the user's account. If you create permanent user accounts for the user, the username and password for those accounts must match the user's eDirectory username and password.

Store Registry Profile: By default, with Dynamic User enabled, when the user is logged out of the terminal server, his or her user profile is removed from the terminal server. This includes the user's registry settings. You can use this option to specify that you want the user's registry settings stored in a persistent location, such as a c:\userprofile directory on the terminal server or a network location accessible by the terminal server, so that the registry settings will be restored the next time the user is logged in to the terminal server. The User Profile Directory field is used to specify the storage location.

Store Document Profile: By default, with Dynamic User enabled, when the user is logged out of the terminal server, his or her user profile is removed from the terminal server. This includes the user's document (c:\documents and settings\username) folder. You can use this option to specify that you want the user's documents stored in a persistent location, such as a c:\userprofile directory on the terminal server or a network

location accessible by the terminal server, so that the documents will be restored the next time the user is logged in to the terminal server. The User Profile Directory field is used to specify the storage location.

User Profile Directory: If you chose to store registry or document profiles, enter or browse for the directory where you want the profiles stored.

You can also include Windows environment variables in the path as well as a special DeFrame variable (%USERNAME%). 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%.

- 7** Click Finish to enable the users as DeFrame dynamic users and to apply the user settings.

B

Upgrading From DeFrame 2.0

The following sections provide information to help you upgrade from Novell® DeFrame™ 2.0 to DeFrame 2.01. To successfully complete the upgrade, you need to complete the tasks in the order listed below.

- ♦ [“Reviewing the DeFrame 2.01 Requirements” on page 115](#)
- ♦ [“Extending the eDirectory Schema” on page 115](#)
- ♦ [“Upgrading DeFrame Terminal Servers” on page 116](#)
- ♦ [“Upgrading User Workstations” on page 117](#)
- ♦ [“Removing Obsolete eDirectory Objects” on page 118](#)

Reviewing the DeFrame 2.01 Requirements

DeFrame 2.01 has different software and hardware requirements than DeFrame 2.0. For example, unlike DeFrame 2.0, DeFrame 2.01 does not support Windows NT servers.

You should review the DeFrame 2.01 requirements in [“Verifying Software Requirements” on page 17](#) to ensure that your current environment meets the requirements.

Extending the eDirectory Schema

DeFrame 2.01 includes Novell® eDirectory™ objects and attributes not present in DeFrame 2.0. To support these new objects and attributes, you need to extend the schema of the eDirectory tree in which your DeFrame objects are

located. You also need to install the DeFrame 2.01 snap-ins for ConsoleOne so that you can administer the new DeFrame objects.

The DeFrame 2.01 schema extensions are compatible with the DeFrame 2.0 schema. Therefore, you can install the DeFrame 2.01 extensions and continue to run your DeFrame 2.0 software on your terminal servers and workstations until you can upgrade them to DeFrame 2.01.

To extend the schema and install the DeFrame snap-ins:

- 1** Make sure you are logged in to the eDirectory tree as a user with rights to extend the schema.
- 2** Launch the setup.exe program located in the \deframe\ndssetup directory of the *Novell ZENworks OnDemand Services 2* CD.
- 3** Follow the on-screen prompts.

Upgrading DeFrame Terminal Servers

New functionality, such as contextless login support, has been added to the DeFrame 2.01 services. To implement this functionality, you need to update your DeFrame 2.0 terminal servers to DeFrame 2.01.

As you upgrade, DeFrame 2.01 and DeFrame 2.0 terminal servers can coexist in the same environment. This means that you can perform all upgrades at one time or you can take a phased approach. If you take a phased approach, please be aware of the following:

- ◆ Users who launch DeFrame terminal server applications through OnDemand Services will be able to continue to do so regardless of the terminal server's DeFrame version.
- ◆ The DeFrame 2.01 services and DeFrame 2.01 client are not compatible with DeFrame 2.0. Therefore, users who are running an the DeFrame 2.0 client will not be able to launch applications from DeFrame 2.01 terminal servers, and users who are running the DeFrame 2.01 client will not be able to launch applications from DeFrame 2.0 terminals servers.

To upgrade a terminal server:

- 1** Make sure the terminal server meets the minimum operating system and Novell Client requirements listed in [“Verifying Software Requirements” on page 17](#).

- 2** Shut down the DeFrame services (Control Panel > Services):
 - ◆ Novell DeACL Service
 - ◆ Novell DeWatch Service
 - ◆ Novell Disc Session Tracking Service
 - ◆ Novell Load Balancing Service
 - ◆ Novell Published App Sync Service
- 3** Use Add/Remove Programs to uninstall the DeFrame 2.0 software from the terminal server. It will be listed as DeServer 2.0.
- 4** Rename the c:\winnt\system32\dappx directory (for example, to dappx_old).

After you've installed DeFrame 2.01 to the terminal server and verified that it is functioning correctly, you can delete the renamed directory.
- 5** Make sure you are logged in to the eDirectory tree where the terminal server's DeFrame Server object is located, and that you have rights to modify the DeFrame Server object.
- 6** Launch the setup.exe program located in the \deframe\tssetup directory of the *Novell ZENworks OnDemand Services 2* CD.
- 7** Follow the on-screen prompts to install the software and create the terminal server's DeFrame object.

IMPORTANT: The Setup program will prompt you to create the terminal server's DeFrame Server object. If you specify the server's existing object (by using the same server name and same context), the Setup program will use the existing object. If you specify a different name and/or context, the Setup program will create a new object and you will need to reconfigure the object with application information, load balancing information, and so forth.

Upgrading User Workstations

If you installed the DeFrame 2.0 client to users' workstations, you need to update it to the DeFrame 2.01 client before users will be able to launch applications hosted on a DeFrame 2.01 terminal server.

To update the DeFrame client on a workstation:

- 1 At the workstation, launch the setup.exe program located in the \deframe\wksetup directory of the *Novell ZENworks OnDemand Services 2 CD*.

Removing Obsolete eDirectory Objects

The DEFAULT_CATALOG object and Alias User objects are not used with DeFrame 2.01. After you've updated eDirectory and your terminal servers and ensured that the system is functioning correctly, you can remove the DEFAULT_CATALOG object and any Alias objects created for DeFrame 2.0 users.

C

Documentation Updates

This section lists updates to the *Installation and Administration* guide that have been made since the initial release of Novell® DeFrame™ 2.01. 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 (Patch 102002)” on page 119
- ♦ “December 20, 2002 (Support Pack 1)” on page 121

October 7, 2002 (Patch 102002)

The following updates were made to the *Installation and Administration* guide for the OnDemand Services 2 patch dated Oct 2002:

Location	Update
“Citrix MetaFrame Requirements (Optional)” on page 18	Added information about installing DeFrame on Citrix MetaFrame XP Feature Release 2 (FR2).
Chapter 3, “Installing DeFrame,” on page 23	In the introduction, under the description of the DeFrame Workstation Setup program, clarified that this Setup program only needs to be run if a user needs to launch thin-client applications from Novell Application Launcher.
“Setting Up eDirectory” on page 24	Added a step (step 17) about applying the OnDemand Services 2 October 2002 patch to update the DeFrame snap-ins for ConsoleOne.

Location	Update
Applying the October 2002 Patch	Added information about applying the OnDemand Services 2 October 2002 patch to update the DeFrame server files. NOTE: This section no longer exists. For Support Pack 1, it was replaced by “Applying Support Pack 1” on page 33 .
“DeFrame Dynamic User (DDU) versus ZENworks for Desktops Dynamic Local User (DLU)” on page 47	Under Local Group Assignments, corrected information about DDU. When a user is added by DDU, the user is not assigned to the Users group as previously stated. The user is not assigned to any group, but is still given access equivalent to the Everyone group.
“Deactivating DDU on a Terminal Server” on page 52	Added information to DDU section about how to deactivate DDU on a terminal server.
“Configuring the Application Object” on page 66	Added a list of characters that cannot be used in Application object names.
Setting Up the Published Application Sync Service in a Multiple Terminal Server Farm Environment.	Added this new section to provide instructions about how to set up the Published Application Sync Service in a multiple-server-farm environment. These instructions explain how to prevent one farm’s Application objects from overwriting another farm’s Application objects when they are added to Novell eDirectory™. NOTE: This section no longer exists. For Support Pack 1 (see “December 20, 2002 (Support Pack 1)” on page 121), the service became a utility and this section was replaced with “Using the DeFrame Published Application Synchronization Utility to Create the Application Object” on page 65 .
Configuring the Published Application Sync Service	Removed information about the classic and nonclassic switches. The classic switch instructed the Published Application Sync Service to use the Citrix master browser’s preferred server information for an application. The nonclassic switch instructed the service to ignore the master browser’s preferred server information so that the DeFrame Load Balancing service would determine an application’s preferred server. The classic mode is no longer an option. The Published Application Sync Service will always ignore the Citrix master browser’s preferred server information. NOTE: This section no longer exists. For Support Pack 1 (see “December 20, 2002 (Support Pack 1)” on page 121), the service became a utility and this section was removed because the startup switches no longer applied.

December 20, 2002 (Support Pack 1)

The following updates were made to the *Installation and Administration* guide for OnDemand Services 2 Support Pack 1:

Location	Update
“Windows Terminal Server Requirements” on page 18	Upgraded the supported Citrix MetaFrame XP version to Feature Release 2 (FR2). Previously, DeFrame supported only Citrix MetaFrame XP Feature Release 1. Added information about additional configuration steps included in a Novell Technical Information Document that need to be completed when installing to FR2.
Chapter 6, “Publishing Applications,” on page 63	The DeFrame Published Application Synchronization utility was changed from a Windows service to a Windows application. Added information about using the utility to import applications from the Citrix Published Application Manager into Novell eDirectory.
“Configuring ICA Client Settings” on page 70	Added information about configuring the ICA client settings on an Application object.
Chapter 7, “Configuring Load Balancing Services,” on page 77	The Load Balancing service and Watchdog service were combined into the Watchdog service. Added information about how load balancing now works.
“Adjusting the Load Calculation Weights” on page 81	Corrected inaccurate information regarding how the server loads are calculated.
“Modifying the Watchdog Service Configuration Settings” on page 85	Added information about using Windows registry settings to provide advanced configuration of the Watchdog service.
Chapter 9, “Controlling DeFrame Service Logging,” on page 93	Added section about how to determine the amount of information the DeFrame services output to their log files.

