

Novell JVM for NetWare®

1.4.1

www.novell.com

INSTALLATION AND ADMINISTRATION
GUIDE



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 © 2002-2003 Novell, Inc. All rights reserved. No part of this publication may be reproduced, photocopied, stored on a retrieval system, or transmitted without the express written consent of the publisher.

U.S. Patent Nos. U.S. Patent No. 5,157,663; 5,349,642; 5,455,932; 5,553,139; 5,553,143; 5,572,528; 5,594,863; 5,608,903; 5,633,931; 5,652,854; 5,671,414; 5,677,851; 5,692,129; 5,701,459; 5,717,912; 5,758,069; 5,758,344; 5,781,724; 5,781,733; 5,784,560; 5,787,439; 5,818,936; 5,828,882; 5,832,274; 5,832,275; 5,832,483; 5,832,487; 5,859,978; 5,870,561; 5,870,739; 5,873,079; 5,878,415; 5,884,304; 5,893,118; 5,903,650; 5,903,720; 5,905,860; 5,910,803; 5,913,025; 5,913,209; 5,915,253; 5,925,108; 5,933,503; 5,933,826; 5,946,002; 5,946,467; 5,956,718; 5,956,745; 5,964,872; 5,974,474; 5,983,223; 5,983,234; 5,987,471; 5,991,810; 6,002,398; 6,014,667; 6,016,499; 6,023,586; 6,029,247; 6,052,724; 6,061,726; 6,061,740; 6,061,743; 6,065,017; 6,081,774; 6,081,814; 6,094,672; 6,098,090; 6,105,062; 6,105,069; 6,105,132; 6,115,039; 6,119,122; 6,144,959; 6,151,688; 6,157,925; 6,167,393; 6,173,289; 6,216,123; 6,219,652; 6,233,859; 6,247,149; 6,269,391; 6,286,010; 6,308,181; 6,314,520; 6,324,670; 6,338,112; 6,345,266; 6,353,898; 6,424,976; 6,466,944; 6,477,583; 6,477,648; 6,484,186; 6,496,865; 6,510,450; 6,516,325; 6,519,610; 6,532,451; 6,532,491; 6,539,381. Patents Pending.

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

www.novell.com

Novell JVM 1.4.1 for NetWare
[May 2003](#)

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

Novell Trademarks

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

NetWare Loadable Module and NLM are trademarks of Novell, Inc.

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

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

Third-Party Trademarks

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

Contents

About This Guide	3
Documentation Updates	3
Documentation Conventions	3
Included in this Manual	3
1 Novell JVM 1.4.1 For NetWare Overview	5
2 Setup and Operating Requirements	7
Important Java Files	7
Supported Video Controllers	7
Hardware Requirements	8
CPU	8
Memory	8
Hard Disk Space	8
Mouse	8
3 Installation Guide	9
Installation Prerequisites	9
Installing or Upgrading NJVM	9
Troubleshooting	10
Java GUI Applications Fail to Load	10
GUI Fails to Load	10
Mouse Does Not Respond	10
Applications Using java.net Do Not Work Properly	11
Supporting the Graphics Font	11
Modifying the Default Font Size	11
4 Administration Guide	13
Java Configuration Files	13
Java.cfg Configuration File	13
Autoexec.ncf	13
Loading and Unloading NJVM and NetWare GUI	13
Loading NJVM	13
Unloading NJVM	14
NetWare GUI Tasks	14
Starting the NetWare GUI	14
Closing the NetWare GUI	14
Viewing Multiple Instances of the Server Console Utility	14
Toggling between NetWare GUI and the Server Console	15
Activating the Taskbar Menu	15
Configuring the Taskbar Menu	15
Using the Menu Editor	16
Configuring X Server Using the GUI Environment Utility	16
Configuring Your System for an Accelerated Video Driver	17
Selecting a Background Pattern	17
Using NetWare GUI without a Mouse	17

Keystroke Actions for Java Text Areas and Text Fields	18
Applet and Application Tasks	19
Running an Applet	19
Running an Application	19
Binding an Application to a Specific Processor	20
Listing the Current Java Processes	20
Shutting Down a Running Java Process.	20
Viewing and Setting Environment Variables.	20
Viewing the Current Values of Environment Variables	20
Setting the DISPLAY Environment Variable	21
Using the Remote Display Feature	21
Setting the CWD Environment Variable	22
Setting the CLASSPATH Environment Variable	22

5 Developer's Guide 25

Prerequisites	25
Guidelines	25
Using Java Native Interface (JNI) on NetWare	25
Required Tools.	25
File Descriptions	25
Building a JNI Example	26
Installing a JNI Example	26
Running a JNI Example	27
Unloading JNI	27
JNI Notes	27
Using WATCOM 11.0 Compiler Flags for Native Method NLM Programs.	28
Source-Level Debugging	28
Required VM Options	29
Obtaining the Fully qualified Domain of the Local Host	30

About This Guide

This guide introduces Novell® JVM 1.4.1 for NetWare® and explains how to install and configure it on your system.

Documentation Updates

For the most recent documentation on using Novell JVM 1.4.1 for NetWare, see the [Novell JVM 1.4.1 for NetWare \(http://www.novell.com/documentation/lg/jvm141/index.html\)](http://www.novell.com/documentation/lg/jvm141/index.html) documentation.

Documentation Conventions

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

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

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

Included in this Manual

- ♦ [Setup and Operating Requirements \(page 7\)](#)
- ♦ [Installation Guide \(page 9\)](#)
- ♦ [Administration Guide \(page 13\)](#)
- ♦ [Developer's Guide \(page 25\)](#)

1

Novell JVM 1.4.1 For NetWare Overview

The Novell® JVM 1.4.1 for NetWare® (NJVM) runs Java* applications, servlets, and applets in a NetWare environment. The NetWare Graphical User Interface (GUI) lets users run graphical applications. NJVM also lets you export the display of X Client programs to any machine running an X server.

NJVM contains the following features:

- ◆ Multiserver install capabilities, which speeds up the installation process.
- ◆ Java does not need to be loaded to run the NetWare GUI.
- ◆ The GUI TaskBar is written in C++, improves performance, adds functionality, and requires less memory.
- ◆ The Server Console Utility lets you load multiple server consoles on the NetWare GUI. This lets you view all NetWare screens simultaneously.

2

Setup and Operating Requirements

This chapter contains information necessary for setting up NJVM. This chapter includes the following topics:

- ♦ [Important Java Files \(page 7\)](#)
- ♦ [Supported Video Controllers \(page 7\)](#)
- ♦ [Hardware Requirements \(page 8\)](#)

Important Java Files

Directory	File	Description
sys:\java\lib	rt.jar	Core Java classes.
	charsets.jar	Character-conversion classes.
	tools.jar	Support classes for SDK tools and utilities.
sys:\java\lib\ext	localedata.jar	Locale data for java.text and java.util.
sys:\java\lib\security	java.policy	Security policy.
	java.security	Security properties.
sys:\java\bin	Java.nlm	Executable files for launching Java applications, tools, and utilities.
sys:\java\nwgfx\bin	Startx.ncf	Starts the NetWare GUI.
	stopx.nlm	Closes the NetWare GUI.
	vesa_rsp.ncf	Script for resetting GUI to default configuration.
sys:\java\nwgfx\lib\x11\fonts		Fonts for displaying characters in a GUI window.
sys:\java\nwgfx\pixmaps		Images used for GUI background.

Supported Video Controllers

Novell[®] supports the VESA BIOS extensions. The Xvesa driver provides graphics support for VESA 1.2 and later compliant video controllers. The resolution for VESA video controllers is controller dependent.

NetWare uses Xvesa as the default X server. It also supports VESA 1.2 through 3.0. NetWare also includes Xi Graphic's Accelerated-X as an alternative X server, which supports over 500 specific video cards.

NJVM currently supports the following video controllers for up to 1600x1200 and True Color:

- ◆ VESA 1.x, 2.x, 3.x.

VESA 1.x and 2.x support covers a large percentage of recently manufactured controllers and provides nonaccelerated support for additional hardware modes.

- ◆ Over 500 specific video controllers.

For instructions on accessing the video card tab in the taskbar menu to view a complete list of supported controllers, see [“Configuring X Server Using the GUI Environment Utility” on page 16](#).

NOTE: Using an accelerated video driver can hinder usability. To learn how to configure your monitor for an accelerated video driver, see [“Configuring Your System for an Accelerated Video Driver” on page 17](#).

Hardware Requirements

The hardware requirements are determined by your version of NetWare.

CPU

- Minimum speed 550 MHz, intel*, Pentium*, or compatible

Memory

- 512 MB Minimum (Refer to NetWare 6.5 memory requirements).

Hard Disk Space

- 200 MB available.

Mouse

- PS/2 or USB.

NOTE: Some applications can function without a mouse, but it is not recommended.

3

Installation Guide

After you install NetWare® software, you can upgrade and start Novell® JVM for NetWare®. This chapter includes the following topics:

- ♦ [Installation Prerequisites \(page 9\)](#)
- ♦ [Installing or Upgrading NJVM \(page 9\)](#)
- ♦ [Troubleshooting \(page 10\)](#)
- ♦ [Supporting the Graphics Font \(page 11\)](#)
- ♦ [Modifying the Default Font Size \(page 11\)](#)

Installation Prerequisites

- NetWare 6 or later running with the latest Support Pack
- NJVM self-extracting executable file
- Client workstation running Windows* 2000/XP with Novell® Client™
- USB or PS/2 style mouse attached

Installing or Upgrading NJVM

- 1** Close the NetWare GUI if it is running.
See [“Closing the NetWare GUI” on page 14.](#)
- 2** Shut down NJVM if it is running.
See [“Unloading NJVM” on page 14.](#)
- 3** Unload xlib.nlm.
- 4** Attach to the target servers as a user with administrator rights to the sys: volume.
- 5** Run the NJVM self-extracting executable file on the client workstation.
This expands the archive and launches the InstallShield installation script.
- 6** On the Welcome screen, click Next.
- 7** Accept the license agreement.
Rejecting the license agreement stops the installation process.
- 8** On the Installation Components screen, click Next.
- 9** On the Installation Requirements and Information screen, review the installation prerequisites and product compatibility issues and then click Next.

- 10** On the Destination Servers for Installation screen, select the servers you want to install NJVM on and then click Next.

If `xinit.nlm`, `java.nlm`, or `xlib.nlm` is still loaded, an alert screen appears.

- 11** (Conditional) If the Alert screen appears, close the NetWare GUI, unload the specified files and then click Next.

InstallShield displays an installation progress bar.

The installation process copies the `java.zip` file to the destination servers and unzips the file on each server.

When the installation is complete, a results screen displays. This screen lists the installation errors, servers that need to be rebooted, and servers that installed successfully. The information displayed can also be found in `c:\temp\InstallShieldLogs\njvm_server_install.log`. For more installation information, refer to the sequential log file in the same directory, `njvm_log.txt`.

The Readme document contains last-minute information about NJVM that is not in this documentation.

Troubleshooting

This section provides you with suggestions for resolving problems you might have when you load NJVM.

Java GUI Applications Fail to Load

If Java applications have trouble running in the NetWare GUI, ensure that the `DISPLAY` variable is set to `127.0.0.1:0.0` or to the IP address of the server. For more information, see [“Setting the DISPLAY Environment Variable” on page 21](#).

GUI Fails to Load

If the NetWare GUI fails to load after you install NJVM, run the following `.ncf` file at the server console:

```
vesa_rsp
```

This resets the NetWare GUI to its default configuration. For more information on the default configuration, see [“Configuring Your System for an Accelerated Video Driver” on page 17](#).

The GUI resolution can then be changed from the taskbar menu. For more information, see [“Configuring X Server Using the GUI Environment Utility” on page 16](#).

Mouse Does Not Respond

If the mouse that is properly attached to your server does not function in the NetWare GUI, it might not be configured correctly.

To automatically reconfigure the mouse:

- 1** Close the NetWare GUI.
- 2** At the server console, enter:

```
vesa_rsp
```

3 Start the NetWare GUI.

See “Loading NJVM” on page 13.

This automatically reconfigures the mouse and also resets the NetWare GUI to its default configuration. For more information on the default configuration, see “Configuring Your System for an Accelerated Video Driver” on page 17.

The GUI resolution can then be changed from the taskbar menu. For more information, see “Configuring X Server Using the GUI Environment Utility” on page 16.

Applications Using java.net Do Not Work Properly

If Java applications that use the java.net package have trouble resolving host names or IP addresses, ensure that the TCP/IP configuration includes the file /etc/resolv.cfg, and that WINSOCK is loaded on the server. The /etc/resolv.cfg file tells the server its fully qualified domain name (FQDN) and allows the server to locate name servers that can resolve names into IP addresses.

Another problem might be that the IP address and DNS name of the client machine need to be added to the /etc/hosts file. The proper entry format is

```
<IP address> <DNS name>
```

Supporting the Graphics Font

The font.properties file controls the Java font-to-native-font mapping in java\lib\font.properties. The font.properties file is platform specific and indicates the fonts that a particular platform uses for its Java virtual fonts. The font.properties file for NJVM is nearly identical to the Solaris* implementation. For a description, see the font properties on the [Sun Web site \(http://java.sun.com/j2se/1.4.1/docs/guide/intl/fontprop.html\)](http://java.sun.com/j2se/1.4.1/docs/guide/intl/fontprop.html).

Modifying the Default Font Size

For some character sets, users, and languages, the default font sizes specified by NJVM are either too small or too large for comfortable and convenient viewing. NJVM has the limited ability to scale up the defaults by a fixed number. This affects only the applications that use the default font size. For example, applications such as the NetWare Install have developer-specified font sizes; these remain unchanged.

To change the defaults, locate your font.properties file located in sys:\java\lib. NJVM defaults to a single font.properties file, but custom installations can have files with added extensions. Extensions are composed of the machine’s language and locale settings. For example, a machine using the English language located in New Zealand would use font.properties.en_nz, and a machine using the Russian language would use font.properties.ru. If the specific font.properties.language.locale combination does not exist, NJVM searches for a font.properties.language file. If that does not exist either, the default font.properties file is used.

After you find the appropriate font.properties file, add the following line to the file:

```
FontSizeCorrection=value
```

Replace *value* with the number you want to increase or decrease the default font sizes by. For example, to change the default Japanese default font size from 12 to 16, open the sys:\java\lib\font.properties.jp file and the following line:

FontSizeCorrection=4

You can use negative numbers, but these are not recommended because the default font sizes are determined based on the minimum size for readability.

4 Administration Guide

This chapter explains the following types of administrative tasks that you can do with Novell® JVM 1.4.1 for NetWare®.

This chapter includes the following sections:

- ♦ [Java Configuration Files \(page 13\)](#)
- ♦ [Loading and Unloading NJVM and NetWare GUI \(page 13\)](#)
- ♦ [NetWare GUI Tasks \(page 14\)](#)
- ♦ [Applet and Application Tasks \(page 19\)](#)
- ♦ [Viewing and Setting Environment Variables \(page 20\)](#)

Java Configuration Files

Java.cfg Configuration File

The java.cfg file is an optional configuration file that can be used to set environment variables. It is located in the sys:\etc directory, and is read when java.nlm is loaded.

Autoexec.ncf

The following lines are added to the sys:\system\autoexec.ncf file during the installation. If somehow they do not get added, you can add them manually right after the “mount all” command in the file.

```
SEARCH ADD SYS:\JAVA\BIN
SEARCH ADD SYS:\JAVA\NWGFX\BIN
STARTX
```

HINT: The STARTX line is optional and should go at the end of the autoexec.ncf file. It causes the NetWare GUI to begin running at server startup. You should delete any other instances of these lines in the autoexec.ncf file.

Loading and Unloading NJVM and NetWare GUI

Loading NJVM

To load NJVM (java.nlm), at the Server Console enter

```
load java
```

This loads NJVM into memory and allows the command interpreter to recognize when a Java application or applet is launched.

Unloading NJVM

You can unload NJVM (java.nlm) from memory by entering the following command at the Server Console:

```
java -exit
```

This command is a clean unload. It lets the applications do a proper shutdown.

Or you can enter

```
unload java
```

This command sends a java -kill command on the applications.

NetWare GUI Tasks

Starting the NetWare GUI

To start the NetWare GUI, at the Server Console enter

```
startx
```

When the Novell button appears on the taskbar menu in the lower left-hand corner, the process is complete.

Closing the NetWare GUI

The following are three ways to shut down the NetWare GUI:

- ◆ Activate the taskbar menu, then click Close GUI > Yes.
For instructions on activating the taskbar menu, see [“Activating the Taskbar Menu” on page 15](#).
- ◆ Press Ctrl+Alt+Backspace.
- ◆ Run stopx from the Server Console.

Viewing Multiple Instances of the Server Console Utility

The GUI Server Console utility is now C-based and does not require Java to be loaded. Multiple instances of the Server Console utility can be loaded. This lets you view multiple server screens simultaneously in the GUI. For example, you can view the Logger screen, Edit screen, and the Console screen from three different windows.

To view multiple instances of the GUI, click the Server Console icon on the taskbar menu, or click Novell > Utilities > Server Console.

NOTE: Some screens have limited functionality and updates only when they are the active screen. You need to view these screens from within a server console that follows the active screen.

The first instance you launch follows the active screen. This means that it will immediately reflect changes made on the active screen. For all other instances, following the active screen is not the

default setting; you need to manually select this option by selecting click View > Follow Active Screen.

All instances following the active screen have an asterisk in front of the screen name.

Toggleing between NetWare GUI and the Server Console

- ◆ To toggle to the next screen, press Alt+Esc.
- ◆ To toggle to the NetWare screen selection list, press Ctrl+Esc.

Activating the Taskbar Menu

To activate the taskbar menu in NetWare GUI, click the Novell button in the lower left-hand corner of the screen.

Configuring the Taskbar Menu

The taskbar menu consists of items that are either menus or commands. Menus contain the following:

- ◆ **Default name**
- ◆ Default name
- ◆ International name (optional)

If you enter an international name, it appears in the taskbar menu; otherwise, the default name appears.

Internationalized names are stored in files with the .properties extension.

Properties File

A Properties file is a text file that contains key=value entries. For example, PROGRAMS=programs or UTILITIES=utilities. You must place these files in the sys:\java\lib\taskbar directory. The properties file also lets you specify characters that you are not able to enter from the server's keyboard. You do this by specifying the Unicode values on the right side of the equal sign (=). For example, you can specify the value RUN with any of the following values:

- ◆ RUNITEM=Run
- ◆ RUNITEM=\u0052\u0075\u006e
- ◆ RUNITEM=R\u0075\n

When you specify the Properties file, you specify the default properties file such as NWMenu.properties. Names are retrieved from that file according to the language and locale of the server. For example, if the NWMenu.properties file is used, the server is set to use German, and there is a NWMenu_de.properties file, then the NWMenu_de.properties file is searched first for the entry. If the entry is not found in this file, then the NWMenu.properties file is searched. If the entry is not found in either file, then the default name is used.

If you want to use a properties file for your names, you can use a default name such as MyNames.properties. If you are providing the product for an internationalized audience, you might want to ship properties files for each of the locales supported by the server.

IMPORTANT: Do not add your own name entries in the NWMenu.properties file. This file is overwritten when you install a newer version of NJVM.

Commands

Commands contain the following:

- ◆ Default name
- ◆ International name
- ◆ The command to execute

IMPORTANT: We recommend that you specify an .ncf file that contains actual commands to execute.

Using the Menu Editor

You configure the taskbar menu by using the Menu Editor. To access this tool, on the taskbar menu, click Settings > Menu Editor.

Creating a Menu Item

- 1** Right-click the location where you want to install the menu item.
- 2** Click either New Command or New Menu.
- 3** Type the information in the edit field.
- 4** Click File > Save on the taskbar menu.

Modifying a Menu Item

- 1** Click the menu item you want to change.
- 2** Modify the command information in the Edit panel.
- 3** Click File > Save on the taskbar menu.

Deleting a Menu Item

- 1** Right-click the menu item.
- 2** Click Remove.
- 3** Click File > Save on the taskbar menu.

Moving Menu Items

- 1** Right-click the menu item.
- 2** Click either Move Up or Move down.
- 3** Click File > Save on the taskbar menu.

If a menu is open, the selected menu item moves into that menu; otherwise, the menu item moves past the menu.

Configuring X Server Using the GUI Environment Utility

You use the GUI environment utility to configure VESA Super VGA and Accelerated-X Servers on NetWare. This utility lets you select the video board, keyboard, mouse, and monitor from a set of predefined hardware devices supported by the GUI environment utility.

- 1 Activate the taskbar menu.
See [“Activating the Taskbar Menu” on page 15](#).
- 2 Click Settings > GUI Environment.

See the GUI environment online help for further documentation on using this utility.

Configuring Your System for an Accelerated Video Driver

To achieve the best video quality when you configure your accelerated video driver, configure the monitor and select a refresh rate as high as possible.

- 1 From the taskbar menu, select GUI Environment.
- 2 Select your video board from the Video Board List.
This activates the Monitor tab. The default settings are Super VGA with 1024 x 768 resolution and Monitor tab deactivated.
- 3 Select your monitor from the list of monitors.
This enables all possible refresh rates for the selected monitor.
- 4 Select the highest available refresh rate, then set the video board to the desired resolution.
- 5 Select Test, then click OK.
- 6 After the test pattern displays, click OK > Yes > Yes.

Selecting a Background Pattern

- 1 Activate the taskbar menu.
For instructions on activating the taskbar menu, see [“Activating the Taskbar Menu” on page 15](#).
- 2 Click Tools > Backgrounds.
All files in the `sys:\java\nwgfx\pixmaps` directory display. The supported graphics formats are xpm, jpeg, and bmp. If you have background patterns (in the supported formats) that you want to use, place them in the `sys:\java\nwgfx\pix\maps` directory.
- 3 Select the desired background pattern and test it by clicking Test.
- 4 Click OK.

Using NetWare GUI without a Mouse

If you start NetWare GUI without a mouse driver attached to a PS/2, or USB, the GUI will start in mouseless mode. When the Mouse Device query screen appears, select No Mouse.

When the NetWare GUI system is executing in mouseless mode, use the keypad keys for mouse movement and button clicks.

IMPORTANT: NUMLOCK must be activated to enable keypad mouse movements.

KeypadKey	Function
Arrows	Move the mouse pointer.
Shift+arrows	Accelerate pointer movement.

KeypadKey	Function
5	Behaves like the default pointer button.
0	Locks the default pointer button down (for easy dragging).
. (decimal)	Unlocks the default pointer button (releases a drag).
+ (plus)	Double-clicks the default pointer button.
alt+tab	Switches to the next program.
alt+shift+tab	Switches to the previous program.

Keystroke Actions for Java Text Areas and Text Fields

The following table shows the keystroke actions for Java text areas and text fields as implemented for NJVM.

Action	Text Area	Text Field
Navigate out forward	Ctrl+Tab	Tab
Navigate out backward	Ctrl+Shift+Tab	Shift+Tab
Move up/down one line	Up-arrow, Down-arrow	
Move to prev/next char	Left-arrow, Right-arrow	Left, Right
Move to prev/next word	Ctrl+Left-arrow, Ctrl+Right-arrow	Ctrl+Left, Ctrl+Right
Move to start/end of line	Home, End	Home/End
Move to start/end of text area	Ctrl+Home, Ctrl+End	
Move up/down a page	PgUp, PgDn	
Select all	Ctrl+/	Ctrl+/
Deselect all	Ctrl+\	Ctrl+\
Extend selection up/down	Shift+Up-arrow, Shift+Down-arrow	
Extend selection left/right	Shift+Left-arrow, Shift+Right-arrow	Shift+Left, Shift+Right
Extend selection to start/end of line	Shift+Home, Shift+End	Shift+Home, Shift+End
Extend selection to previous/next word	Ctrl+Shift+Left-arrow, Ctrl+Shift+Right-arrow	Ctrl+Shift+Left, Ctrl+Shift+Right
Extend selection to start/end of text area	Ctrl+Shift+Home, Ctrl+Shift+End	
Extend up a page	Shift+PgUp	
Extend down a page	Shift+PgDn	

Action	Text Area	Text Field
Copy selection	Ctrl+C	Ctrl+C
Cut selection	Ctrl+X	Ctrl+X
Paste	Ctrl+V	Ctrl+V
Delete next character	Delete	Delete
Delete previous character	Backspace	Backspace
Insert tab	Tab	
Toggle Insert/Replace mode	Insert	Insert
Insert line break	Enter	
Submit entry		Enter

Applet and Application Tasks

Running an Applet

- ♦ To run an applet, enter:
`appletviewer html_file or URL`
- ♦ To run an applet sending any console output to a new screen, enter the following:
`applet -J-ns html filename`

For example, suppose an applet called MYAPP and its corresponding HTML file are found in `sys:\mapps\`. To run this applet, you would enter

```
applet SYS:\mapps\myapp.html
```

Running an Application

- ♦ To run an application, enter
`java application_name`
- ♦ To run the application in a new screen, enter
`java -ns application_name`

NJVM assigns a screen to the application, then executes the application. If the application requires a graphical interface, NJVM loads NetWare GUI and displays the application.

IMPORTANT: If the application requires console keyboard input, the `-ns` option must be used.

While the application is running, you can use the server console utility to view any output on the console screens. For instructions using the Server Console utility, see [“**Toggling between NetWare GUI and the Server Console**” on page 15](#).

All GUI applications appear in the same GUI screen.

Binding an Application to a Specific Processor

To force an application to run on a specific processor, use the `-mp` flag. For example, if you want to assign a JVM to a processor, the processor number should follow `-mp`. To assign a JVM specifically to processor 3, enter the following at the Server Console:

```
java -mp3 application_name
```

To assign a JVM to processor 0, enter the following at the Server Console:

```
java -mp0 application_name
```

NOTE: Using the `-mp0` flag is equivalent to disabling multiprocessor support. This flag should be used only with applications that are considered `mp-unsafe`.

Listing the Current Java Processes

To display a list of currently running processes and their process IDs, enter the following at the Server Console:

```
java -show
```

Shutting Down a Running Java Process

- 1 Get a list of the running Java processes.

See [“Listing the Current Java Processes” on page 20](#).

- 2 At the Server Console, enter

```
java -shutdown process_ID
```

For example:

```
java -shutdown610
```

Replace 610 with the ID of the process to be killed.

- 3 To shut down all running processes, enter the following at the Server Console:

```
java -shutdownall
```

NOTE: The `-shutdown` option is new in NJVM 1.4.1. When you use `-shutdown`, Java calls the application's registered shutdown hooks. This provides a more graceful shutdown than the Java `-kill` command, which closes the application without calling the application's shutdown hooks. It is the application's responsibility to register its shutdown hooks.

Viewing and Setting Environment Variables

Viewing the Current Values of Environment Variables

- 1 Start NJVM.

See [“Loading NJVM” on page 13](#).

- 2 At the Server Console, enter

```
envset
```

When you unload NJVM (`java.nlm`), the process deletes the Environment variables that were set using `ENVSET`. You can add Environment variables in the `sys:\etc\java.cfg` file, which is read when Java loads.

NOTE: The file SYS:\ETC\JAVA.CFG is not required for NJVM. However, it might be convenient to create it and populate it with common environment variables.

Setting the DISPLAY Environment Variable

IMPORTANT: It would be helpful to understand the UNIX* XWindows environment to set up the DISPLAY environment variable.

1 Start NJVM.

See [“Loading NJVM” on page 13](#).

2 At the Server Console, enter

```
envset DISPLAY=IP_address:display
```

The default value for *display* is 127.0.0.1:0, which is the address of the loopback interface and the default value for the display and screen you want to use.

Using the Remote Display Feature

The Remote Variable Display feature lets you export the display of any X client programs to any machine on the network running an X server. Because NetWare GUI is X based, all Java GUI applications and applets are considered X clients and are able to use the remote display feature.

Allowing Remote Access to a Server

You must configure your destination server to allow remote access. The following are two methods to authorize access of a remote X client to a local X server:

♦ Enter **-ac** at the X server command line.

This turns off access control and lets any other client access the server. Be cautious with this method. It lets anyone who can connect to your X server run programs on your display.

♦ Run xhost on the local system.

You use this command to add hosts to the list of client workstations that you allow to access your server. You also use this command to delete workstations from this list. For more information on determining the best method of allowing remote access, see your X server user manual.

Displaying a Java Applet or Application on a Local X server

1 Start X server at the local system.

2 Allow the remote NetWare system to access the local X server.

See [“Allowing Remote Access to a Server” on page 21](#).

3 At the local system, start RCONJ and log in to the NetWare server.

4 If Java is not loaded on the NetWare server, do the following in the RCONJ window:

4a Load Java.

4b Enter:

```
envset display=ip_address_of_local_x_server:0
```

4c Start the Java applet or application.

- 5 (Conditional) If Java is loaded on the NetWare server, enter the following in the RCONJ window and then start the Java applet or application.

```
envset display=ip_address_of_local_X_server:0
```

- 6 (Conditional) If you want the application to always open to the GUI screen, enter the following at the command line of the NetWare server:

```
envset display=ip_address_of_local_X_server:0.0
```

All Java GUI applications will now display on the remote client workstation on the GUI screen. If you do not want the application to always open to the GUI screen, remove the “.0” from the end of the IP address.

NOTE: The Server Console screen does not automatically switch to the GUI screen when a GUI application is opened on the remote exports or foreign IP addresses.

Setting the CWD Environment Variable

You can use the Current Working Directory (CWD) variable to run an application in a specific directory even if it is not part of a package. The CWD variable is set to the root when Java is loaded; however, you can set it to any desired directory.

- 1 Start NJVM.

See [“Loading NJVM” on page 13](#).

- 2 To set the CWD for all applications, enter the following at the Server Console:

```
envset CWD=pathname
```

For example:

```
envset CWD=SYS:\MYJAVA\MYAPP
```

The preferred method is to set the CWD for the current application by using the -env option. For example:

```
Java -envCWD=SYS:\MYFILES MYAPP
```

Setting the CLASSPATH Environment Variable

A default CLASSPATH variable is set when Java loads. If the CLASSPATH variable is set incorrectly, your Java applications might not run. You can improve the performance by reducing the size of the CLASSPATH variable.

- 1 Start NJVM.

See [“Loading NJVM” on page 13](#).

- 2 Enter the following at the Server Console:

```
envset CLASSPATH=<current CLASSPATH value>;path_to_append
```

For example, entering **envset** could produce the following for the CLASSPATH variable:

```
CLASSPATH=SYS:\JAVA\LIB;SYS:\
JAVA\CLASSES
```

To append the path sys:\myclasses to this CLASSPATH, enter

```
envset CLASSPATH=SYS:\JAVA\LIB;
SYS:\JAVA\CLASSES;SYS:\MYCLASSES
```

Or substitute *current CLASSPATH value* with \$CLASSPATH, as in the following:

```
envset CLASSPATH=$CLASSPATH;SYS:\MYCLASSES
```

The \$ symbol in \$CLASSPATH allows the substitution of the current value of the CLASSPATH environment variable.

The CLASSPATH variable provides a default path. The directory to this path, sys:\java\classes, is created by the install program. To use this default path, or any path inside the CLASSPATH variable, place your classes in the directory pointed to by the path.

NOTE: The default CLASSPATH variable includes (.), which means to look in the current working directory for the application. It also includes the CWD specified with the -env option. If the CWD variable is not set correctly, the application might not run. For more information, see [“Setting the CWD Environment Variable” on page 22](#).

5

Developer's Guide

This chapter contains information for developing Java applications with Novell® JVM for NetWare®.

Prerequisites

- NJVM running
See [“Loading NJVM” on page 13.](#)

Guidelines

- ♦ [Using Java Native Interface \(JNI\) on NetWare \(page 25\)](#)
- ♦ [Using WATCOM 11.0 Compiler Flags for Native Method NLM Programs \(page 28\)](#)

Using Java Native Interface (JNI) on NetWare

The java\bin directory includes a file named jni.zip. When you extract this file, it demonstrates simple native method examples for Java 1.x. This section is an example of how to write native methods for NJVM. It is not a comprehensive tutorial on writing native methods.

Required Tools

- NJVM
- JDK 1.4.1 for Win32
- NetWare SDK March 2003
- Watcom 11.0 (10.6 is not supported.)
- Microsoft* NMAKE (Watcom supplies one, but use it at your own risk.)
- MKS Utilities (grep, cp, rm, sed) in your PATH

File Descriptions

File	Description
MAKEFILE	Make file for NetWare
NWNATIVE.JAVA	Native method Java class
NWTEST.JAVA	Test class

File	Description
README.TXT	
NWIMPL.C	Native method C implementation
NWMAIN.C	CLIB NLM™ wrapper - main()

Building a JNI Example

After you unpack the package, complete the following:

- 1 Configure the following parameters in Makefile:

WIN32JAVABASE - Specify where the Win32 JDK is installed.

NWJAVABASE - Specify where the NetWare JDK is installed (typically the mounted SYS: drive).

NLMSDKBASE - Specify where the NetWare NDK is installed.

WATCOMBASE - Specify where Watcom 11.0 is installed.

- 2 Enter the following at a Windows command prompt:

```
nmake
```

The following are the file descriptions:

- ◆ **MAKEFILE.** Make file for NetWare
- ◆ **NWNATIVE.JAVA.** Native method java class
- ◆ **NWTEST.JAVA.** Test class
- ◆ **README.TXT**
- ◆ **NWIMPLE.C** Native method C implementation
- ◆ **NWMAIN.C** CLIB NLM wrapper - main ()

- 3 To clean the build, enter the following at a Windows command prompt:

```
nmake clean
```

Installing a JNI Example

After you build the example and map your server volume sys: to drive G:, complete the following steps to install JNI:

- 1 Check to see if G:\java\classes exists as a directory. If not, enter

```
mkdir G:\java\classes
```

- 2 Copy the NLM to G:\java\bin by entering:

```
copy nwnative.nlm G:\java\bin
```

- 3 Copy the classes to G:\java\classes, then at the command prompt in Windows enter:

```
copy *.class g:\java\classes
```

Running a JNI Example

If your CLASSPATH variable is correct, you can run JNI by entering the following at the command prompt:

```
java NWTest
```

Unloading JNI

To unload JNI, at the Server Console enter

```
java -exit
```

JNI Notes

This section contains notes and examples that might be helpful with the JNI process:

1. What you can do in your main() function depends on how you build your NetWare Loadable Module™ (NLM) program. If you include the following option in your Makefile, your NLM program can use a synchronized startup:

```
Option SYNCHRONIZE
```

With the synchronize option, you can initialize any global information your NLM program might contain in the main() function as long as you call the following function after initialization has completed:

```
void SynchronizeStart();
```

For example:

```
main()
{
  /* Do global initialization */
  SynchronizeStart(); /*MUST BE CALLED */
  ExitThread, 0); /* MUST BE CALLED*/
}
```

If you choose to not use a synchronized startup for your NLM program, you must limit your main() function to the following:

```
main ()
{
  ExitThread (TSR_THREAD,);
}
```

2. If you use the Metroworks IDE, the Synchronized linker option can be turned on by clicking the following:

```
ALT+F7 > NLM Linker > Flags > Synchronize
```

3. Do not use the standard malloc(), realloc(), or free() calls directly. Java provides the following macros in sys_api.h instead:
 - ◆ sysMalloc: Same parameters as malloc()

- ◆ sysFree: Same parameters as free()
- ◆ sysRealloc: Same parameters as realloc()
- ◆ sysCalloc: Same parameters as calloc()

Using these macros gives you free resource tracking. This also lets the memory used by your NLM program use Virtual Memory in NetWare 6.5. In some instances, you might prefer memory returned from malloc, such as buffers used for callbacks or ECBs.

4. When you link your NLM, you might get the following errors:

```
Warning! W1008: cannot open math387s.lib: No such file or directory
Warning! W1008: cannot open noemu387x.lib: No such file or
directory
```

```
Warning! W1008: cannot open emu387x.lib: No such file or directory
Warning! W1008: cannot open clib3s.lib: No such file or directory
```

If you find the correct libraries to correct for these warnings, you'll link in a bad prelude.obs and things will not work properly. If you are using C++, see Item 5 below. To prevent these warnings, add the following Makefile option:

```
Option NoDefaultLibs
```

5. If you are using C++, add the following lines to the link file:

```
LIBPath $(WATCOM)\lib386;$(WATCOM)\lib386\netware;
```

```
LIBFile $(WATCOM)\lib386\plbx3s.lib
```

Remove the \$(PRELUDE) entry from the file directive.

6. If you used the word *stub* in the name of any of your native method classes, change sedscript to ensure that it doesn't corrupt your .EXP file.

Using WATCOM 11.0 Compiler Flags for Native Method NLM Programs

Optimized flags are

```
/zp=1 /ri /ei /5s /or /ot /w3 /s /zq /ez
```

Debug flags are

```
/zp=1 /ri /ei /d2 /od /3s /w1 /s /zq /ez
```

IMPORTANT: /ri and /ei are critical for building native method NLM™ programs.

Source-Level Debugging

You can debug applications running on NetWare using any JDPA-compliant debugger or debugging IDE. The following are available debuggers:

- ◆ NetBeans (www.NetBeans.org)
- ◆ Forte (forte.sun.com)
- ◆ Novell's Silverstream IDE

Remote and local debugging of Java applications are available through the socket transport (dt_socket).

Required VM Options

To debug applications, you need the following debug invocation options:

- ◆ **-Xdebug** enables debugging.
- ◆ **-Xrunjdpw *sub_options*** loads the JPDA reference implementation of JDWP. This library resides in the target VM and uses JVMDI and JNI to interact with it. It uses a transport and the JDWP protocol to communicate with a separate debugger application. The following table describes the specific *sub_options*:

Sub_option	Required	Default Value	Description
Help	No	N/A	Prints a brief help message and exits the VM.
Transport	Yes	None but then must be dt_socket	Name of the transport to use in connecting to the debugger application.
Server	No	N	If Y, listen for a debugger application to attach; otherwise attach to the debugger application at the specified address. If Y and no address is specified, choose a transport address to listen to a debugger application at and print the address to the standard output stream.
Address	Yes, if <i>server = n</i> ; otherwise, no		Transport address for the connection. If <i>server=n</i> , attempt to attach to debugger application at this address. If <i>server=y</i> , listen for a connection at this address.

-Xrunjdpw Examples

```
-Xrunjdpw:transport=dt_socket,server=y,address=8000
```

Listen for a socket connection on port 8000. Suspend this VM before main class loads (*suspend=y* by default). After the debugger application connects, it can resume the VM.

```
-Xrunjdpw:transport=dt_socket,address=8000
```

Start the application without suspending. Then listen for a socket connection on port 8000. When the debugger application connects, it can suspend the VM.

Examples

```
java -Xdebug  
-runjdpw:transport=dt_socket,server=y,address=8000  
-Xbootclasspath/a:SYS:/myapp/foo.jar HelloWorld
```

Listen for a socket connection on port 8000. Suspend this VM before the main class loads. After the debugger application connects, it can send a JDWP command to resume the VM of the HelloWorld application.

Obtaining the Fully qualified Domain of the Local Host

With NJVM, the call `java.net.InetAddress.getLocalHost().getHostName()` will return the name of the NetWare server. To obtain the fully qualified domain (DNS) name, (FQDN) of the local host, use the following:

```
InetAddress localhost = InetAddress.getByName ( InetAddress.getLocalHost().getHostAddress() );
```

The object, `localhost`, now contains the correct FQDN and can be obtained via the call, `localhost.getHostName`.