JavaOS™ for Business™

JavaOS™ for Business™ Version 2.0

Network Operations

JavaOS™ for Business™ Version 2.0

Network Operations

Contents

Part 1. Planning and installation . . . 1

Chapter 1. Introduction to JavaOS for Business . . . 3
Network computers . . . 3
Features of JavaOS for Business . . . 3

Chapter 2. Network Planning for JavaOS for Business . . . 5
Network topology . . . 5
Network information gathering . . . 6
Networking services and protocols . . . 8

Chapter 3. Installing JavaOS for Business on the Server . . . 11
Server Hardware Requirements . . . 11
Server Software Requirements . . . 11
Server Software Installation . . . 12
Installing Microsoft Windows NT Server 4.0 . . . 12
Preparing to Load Server Software . . . 13
Installing Java Developer's Kit 1.0.2 . . . 13
Installing Java Developer's Kit 1.1.4 . . . 14
Installing HotJava Browser 1.1.2 . . . 14
Preparing for IBM Network Station Manager on National Language Versions . . . 15
Installing the IBM Intermediate Support Driver . . . 16
Installing IBM Network Station Manager . . . 17
Installing IBM TCP/IP Services . . . 17
Installing IBM Network Station Manager Service Pack 1 . . . 18
JavaOS for Business Software Installation and Initial Configuration . . . 18
Directory Structure Overview . . . 18
Making the files available . . . 19
Initial configuration tasks . . . 20
Configuring the IBM TCP/IP Configuration Utility . . . 20
Administrator Access in National Language Versions . . . 20
Creating an anonymous NFS User . . . 21
Creating an anonymous Microsoft Windows NT Server 4.0 user . . . 22
Configure IBM Network Station Manager TCP/IP Services . . . 23
Setting Windows NT File System Security Permissions . . . 27
National Language Font Configuration . . . 31
Configuring DHCP . . . 31
Starting the JavaOS System Database (JSD) Server . . . 33

Chapter 4. Network Computer Configuration . . . 35
Network Computer Hardware Requirements . . . 35
Configuration Task Overview . . . 35
Users and Groups . . . 36
Machines, Profiles, and Platforms . . . 36
Network Computer Configuration . . . 36
Starting the JavaOS Configuration Tool (JCT) . . . 37
Creating a platform definition . . . 38
Configuring required services . . . 38

Defining a Machine . . . 42
Importing a JSD File . . . 42
Creating the Master Configuration File . . . 45
Creating an Emergency Repair Disk . . . 45
User Configuration . . . 46
Creating Groups . . . 46
Creating a user on the Microsoft Windows NT Server 4.0 . . . 47
Create a home directory . . . 47
Defining an alias for the home directory . . . 48
Set Windows NT file permissions for home directory . . . 48
Create a JavaOS for Business user . . . 49
Configuring required services . . . 49
Creating the Master Configuration File . . . 50
Printing . . . 51
Testing . . . 53
Installation Troubleshooting and Common Problems . . . 53

Part 2. Configuration Planning . . . 55

Chapter 5. JavaOS System Database and JavaOS Configuration Tool . . . 57

Chapter 6. Customizing login processing . . . 59
Network computers and smart cards . . . 59
Login scenario . . . 60
Setting up authentication . . . 60
Setting up local authentication . . . 60
Setting up remote authentication . . . 62
Setting up network computers that do not require user IDs . . . 63
Setting the user inactivity time and logout warning time . . . 64

Chapter 7. Replacing the JavaOS for Business artwork . . . 65

Chapter 8. Changing and configuring the HotJava browser . . . 67
Using the HotJava browser at a non-US English server . . . 67
Keyboard considerations . . . 67
Using the JavaOS Configuration Tool . . . 67
Using the HotJava browser to run the JCT at the server . . . 68
Environment variable considerations . . . 68
Customizing help options . . . 68
Changing the address for the Comments option . . . 68
Changing the address for the Submit a Bug option . . . 69
Deleting the Comments or Submit a Bug option . . . 69
Configuring the HotJava browser for network computers . . . 69

Chapter 9. Implementing a default main application . . . 71
Using the HotJava browser as the main application . . . 71
Specifying a different startup application . . . 71
Guidelines for using application load settings . . . 72
How applications are loaded . . . 72

Chapter 10. Defining and specifying device drivers . . . 73
Adding a new device driver . . . 73
Boot time device drivers . . . 73

Drivers not required at boot time . . . 74
Supplied device drivers . . . 74

Chapter 11. Defining the keyboard, language, and time zone settings . . . 75
Selecting the keyboard settings . . . 75
Selecting the language settings . . . 75
Selecting the language format settings . . . 75
Selecting the region settings . . . 75
Selecting the time zone . . . 76

Chapter 12. Managing fonts . . . 77
Font files . . . 77
The FONT.LST file . . . 77
The font.properties file . . . 77
Font files included in JavaOS for Business . . . 77
Creating or adding a font . . . 78
Deleting a font . . . 78

Chapter 13. Mounting an NFS file system . . . 79

Chapter 14. Specifying keyboard and mouse drivers . . . 81
Keyboard driver . . . 81
Mouse driver . . . 81

Chapter 15. Specifying video information . . . 83
Configuring video resolution . . . 83
Configuring an SVGA monitor . . . 83
Configuring an S3 Trio 64 Monitor . . . 83

Chapter 16. Specifying multimedia information . . . 85

Chapter 17. Specifying printing information . . . 87
Local printing . . . 87
Parallel printer . . . 87
Serial printer . . . 87
Remote printing . . . 87
Network Information System printers . . . 87

Chapter 18. Specifying miscellaneous information . . . 89
Proxies . . . 89
System database location . . . 89
System image location . . . 89
SNMP server location . . . 89
TFTP location . . . 89
Parallel and serial port drivers . . . 89
Just in time (JIT) compiler . . . 89
Token ring and Ethernet drivers . . . 90

Chapter 19. Specifying troubleshooting services . . . 91

Part 3. JavaOS for Business system administration . . . 93

Chapter 20. Getting started with the JavaOS Configuration Tool . . . 95

Starting the JavaOS Configuration Tool . . . 95
JavaOS Configuration Tool main panel selections and terminology . . . 95
Planning an initial sequence of assigning resources and settings . . . 98
Groups and users . . . 98
Platform and machine resources and settings . . . 100

Chapter 21. JavaOS Configuration Tool edit panels . . . 101
Using the edit panels . . . 101
Modifying user information . . . 102
Changing machine information . . . 102
Adding a setting or resource . . . 102
Removing a setting or resource . . . 102
Changing a setting or resource . . . 102
Overriding a setting or resource . . . 103
Descriptions of the available settings and resources . . . 103
Application Load Settings . . . 103
Base Smart Card Services . . . 104
Conditional debug . . . 104
Error logger . . . 104
HotJava Configuration . . . 104
HotJava Load Settings . . . 104
iiimp setting . . . 105
Intel EtherExpress Pro 100 . . . 105
Just In Time (JIT) compiler setting . . . 105
Keyboard and Mouse Base Settings . . . 105
Keyboard driver setting . . . 106
Keyboard format settings . . . 106
Locale format settings . . . 106
Login Settings . . . 107
Mouse driver setting . . . 107
NC PCI token ring driver . . . 107
Network File System (NFS) . . . 108
NFS AutoMounter . . . 108
NIS Printers . . . 108
NS1000 Smart Card Driver . . . 108
NSL Authenticator for Login . . . 108
OpenCard . . . 109
Parallel Port Driver . . . 109
Parallel Printer . . . 109
PCI Token Ring Driver . . . 110
PowerPC Audio Driver . . . 110
Proxies . . . 110
Remote Printer . . . 111
S3 Trio 64 Driver . . . 111
Serial Port Driver . . . 111
Serial Printer . . . 111
Smart Card Authenticator Format . . . 112
SNMP Manager Address . . . 112
Sound Blaster 16 audio driver . . . 112
SVGA driver . . . 113
System Database Location . . . 113
System Image Location . . . 113
TFTP AutoMounter setting . . . 113
Time zone settings . . . 113

Trace tool settings . . . 114
Video driver . . . 114

Chapter 22. Managing users . . . 115
User's tasks . . . 115
Preparing for users . . . 115
User's problems . . . 115

Chapter 23. System Maintenance . . . 117
The master configuration file . . . 117
System settings . . . 117
JavaOS Configuration Tool logging . . . 119
Working with the JavaOS System Database . . . 119
JavaOS for Business properties . . . 119

Chapter 24. Using the Java command line . . . 121

Chapter 25. Using the netShell commands . . . 123

Part 4. Messages . . . 125

Chapter 26. Server messages . . . 127
Numbered server messages . . . 127
Unnumbered server messages . . . 133

Chapter 27. User messages . . . 135

Part 5. Troubleshooting JavaOS for Business . . . 137

Chapter 28. Determining the source of a problem . . . 139
The troubleshooting process . . . 139
Checking the problem environment . . . 140
Selecting a JavaOS for Business tool . . . 140
Error logs . . . 140
Conditional debug messages . . . 140
Debug trace . . . 140
ICAT debugger . . . 141
Isolating the Source of a Problem . . . 141
Hardware or firmware . . . 141

Chapter 29. Checking your software level . . . 143

Chapter 30. Reporting problems . . . 145

Chapter 31. Booting the network computer . . . 147
Typical machine information . . . 147
Typical user information . . . 147
Network computer bootup . . . 147
Starting the bootup process . . . 147
network computer bootup details . . . 148

Chapter 32. Using the error log . . . 149

Enabling error logging . . . 149
The error log daemon . . . 150
Using the error log viewer . . . 150
Starting the error log viewer . . . 150
Error log commands . . . 151
Error log entry contents . . . 151

Chapter 33. Using the debug trace . . . 153
The debug trace daemon . . . 153
Starting the debug trace daemon . . . 153
Using the debug trace formatter . . . 154
Turning on and off the traces . . . 154
Major and minor code name space for performance trace . . . 154
System resources used by traces . . . 155
Trace configuration parameters . . . 155
Using debug tools in the boot cycle . . . 155
jperf commands . . . 155
Trace hooks . . . 156

Chapter 34. Using the ICAT kernel debugger . . . 163

Chapter 35. Using the conditional debug (console output) messages . . . 165
Conditional debug commands . . . 165
Major and minor code name space for conditional debug . . . 165

Chapter 36. The JCT tool log . . . 167

Part 6. Appendixes . . . 169

Appendix A. Tested Network Computer Hardware Configurations . . . 171
Basic network computer configuration . . . 171
Video configurations . . . 171
Audio configurations . . . 171
Network adapters . . . 172
Printers tested . . . 172

Appendix B. Uninstalling the IBM Intermediate Support Driver . . . 173

Index . . . 175

Figures

1. A simple stand-alone network . . . 5
2. A connected network . . . 6
3. The myjavaos.com network . . . 7
4. Defining an Administrators group for national language versions . . . 15
5. Installing the IBM Intermediate Support Driver . . . 16
6. JavaOS for Business network computer binary tree . . . 19
7. Starting the NFS User Configuration Program . . . 21
8. Defining an anonymous NFS user . . . 21
9. Using the User Manager for Domains . . . 22
10. Creating a Windows NT User . . . 23
11. Starting the IBM TCP/IP Configuration Utility . . . 23
12. TFTPD Configuration . . . 24
13. Boot and Services Configured . . . 25
14. Making system services available . . . 25
15. Making applications available . . . 26
16. Setting Windows NT File System Security . . . 27
17. Giving administrators access to \javaos . . . 28
18. Removing everyone's access to \javaos . . . 28
19. Giving the NSMUser group access . . . 29
20. Sample dhcpsd.cfg File . . . 32
21. Verifying the correctness of dhcpsd.cfg . . . 33
22. The JavaOS Configuration Tool . . . 37
23. Importing a JSD File . . . 43
24. A sample JSD File . . . 44
25. A sample MCF for a JavaOS for Business network computer. . . . 45
26. Organization of home directories . . . 47
27. A sample MCF with the serial driver configured. . . . 51
28. The Use workstation Login section of the Login panel . . . 61
29. Login settings panel showing the definition of a guest user ID . . . 63
30. Two groups, each with several users. . . . 96
31. Machines organized into platforms. . . . 97
32. Machines organized in overlapping platforms and profiles . . . 98
33. The Assign Groups panel of the JavaOS Configuration Tool . . . 99
34. JavaOS Configuration Tool edit panel . . . 101
35. Settings panel for system settings in the JCT . . . 118
36. Flow of eliminating problems by solving small ones quickly . . . 139
37. Sample error log output . . . 151

About this book

This information describes the planning and installation steps for the JavaOS™ for Business™ operating system, including system bootup and user and network computer management.

This information also tells how to configure and manage the JavaOS for Business operating system, including system bootup, user management, and network computer management. It also includes descriptions of problem solving strategies and tools for the operating system. It describes the various debug tools, error logs, traces, and messages that a system administrator could use to solve problems.

Some operational procedures and problem solving strategy is hardware dependent. See the documentation supplied with your hardware for information about your specific hardware.

Who this book is for

The information in this book is for readers who:

- Plan for and deploy networks of network computers
- Install JavaOS for Business on a Microsoft Windows NT Server 4.0
- Perform an initial test of a new version of JavaOS for Business before deploying it in their enterprise
- Operate systems using the JavaOS for Business operating system
- Administer system access to users, groups, network computers, and servers
- Diagnose and solve system problems

This book is not for the users of applications running on the JavaOS for Business operating system although it lists messages a user might see.

Readers of this book should be familiar with:

- Network administration
- Operating systems
- Microsoft Windows NT Server 4.0 network and user administration concepts and tools
- Java Developer's Kit (JDK)
- Graphical user interfaces (GUIs)
- Online debuggers
- Network communication methods
- End user problem solving

Before using this information, you should be familiar with the content of the JavaOS for Business library:

Title	Audience and content
JavaOS for Business Network Operations	People who require planning and installation steps for the JavaOS for Business operating system, including system bootup and user and network computer management. System and network administrators who need to plan for, configure, and manage the JavaOS for Business system on a day-to-day basis.
JavaOS for Business Keyboard Reference	Administrators deploying JavaOS for Business in countries using different character sets and different keyboard layouts.
JavaOS for Business Device Driver Guide	Programmers wishing to add a new device driver or modify an existing device driver and make that driver available to JavaOS for Business network computer users.
JavaOS for Business Application Development Guide	Programmers wishing to create and deploy a new desktop application taking advantage of the features added to the JavaOS for Business operating system.

See any additional documentation supplied with your hardware platform for information about your specific hardware.

Conventions and terminology used in this book

This information uses the following typographical conventions:

italics	Used to specify a variable. Substitute your variable for the word in italics. For example, The *user_ID* can be up to 24 bytes long.
bold	Used to specify a command. For example, type **rdisk** to start a system dump.
`monospace`	Used to show system output. For example, the error message `Add paper to the printer` is displayed. Monospace is also used to show edit panel defaults.

Part 1. Planning and installation

Chapter 1. Introduction to JavaOS for Business

The JavaOS™ for Business™ operating system is designed for network computers and based on Java technology produced by Sun Microsystems, Inc. While other Java environments run on top of existing operating systems, JavaOS for Business provides just enough operating system support services so that a network computer can manage its resources and support a 100% Pure Java environment.

JavaOS for Business provides better performance and a substantially reduced memory footprint because it was designed from the beginning to support Java applications and does not need to provide backward compatibility with legacy workstation applications.

Network computers

As corporations moved from mainframe computing to personal workstations, the size and complexity of enterprise networks increased proportionally. The cost of purchasing a full-function workstation is only a small fraction of the total cost of ownership, which includes the support and maintenance expenses that are incurred over the workstation's lifetime, such as installing and updating system and application software, technical support and troubleshooting, and end-user education. Add to these costs the cost of the networking infrastructure necessary to interconnect the workstations and the result is an expensive and complex operation over which enterprises need better control.

The network computer reduces the total cost of owning workstations, particularly for large corporations. A network computer contains no hard disk or application software. When the network computer boots, it loads its operating system and application software over the network and then runs them locally.

As a result, system and network administrators do not have to maintain each workstation individually. Because the system and application software resides on one or more servers, the software needs to be updated only once on the servers. The next time the network computer needs to run an application, the most recent version is obtained from the server, making software updates to network computers automatic. The JavaOS for Business operating system software and device drivers are also maintained this way, allowing network computers to refresh their operating system software simply by rebooting from the server.

Users benefit also. Because they are freed from backing up their own data and managing new operating system and application software updates, they can concentrate on their important tasks.

Features of JavaOS for Business

JavaOS for Business builds upon the prior versions of JavaOS by providing the following new and improved features:

- Support for JDK 1.1.4 applications and applets
- Improved performance and memory management
- Dynamically loadable device drivers and applications
- A layered architecture allowing pieces of the operating system to be independently updated and replaced
- Centralized administration of network computers and associated applications from the server
- Improved reliability, availability, and serviceability of the JavaOS for Business operating system and applications running on the network computers

Chapter 2. Network Planning for JavaOS for Business

A network computer running JavaOS for Business must interact with one or more server machines to access and use different network services. These network services can be provided by different kinds of servers on a network. The examples shown here assume that all of the boot and administration services for JavaOS for Business are provided from a single Microsoft Windows NT Server 4.0 system. This server is referred to as the primary server to distinguish it from other servers on the network. Through network connections, the JavaOS for Business network computers can attach to a variety of servers, including AS/400, RISC System/6000, and System/390.

As the system and network planner, you need to plan the integration of JavaOS for Business network computers into your computing environment. This section lists the tasks that need to be completed before you install JavaOS for Business.

A good network requires good planning. Network planning is divided into two categories, the physical network topology and the detailed planning required for each network computer. Both of these topics are described below.

Network topology

Network topology refers to the physical interconnections between machines in a network. Network topologies range from simple networks to complicated networks of networks. There are many different types of networks that can be implemented. This document addresses two basic types of networks, the stand-alone network, and the connected network. More complex networks can be created and maintained, but describing them is outside the scope of this document.

The stand-alone network is the simplest network, consisting of at least one server and one or more JavaOS for Business network computers connected to the same network by the same adapter hardware, either token-ring or Ethernet adapters. All machines reside in the same IP subnet and the entire network is self-contained. This is the preferred choice for a test environment or for demonstration systems.

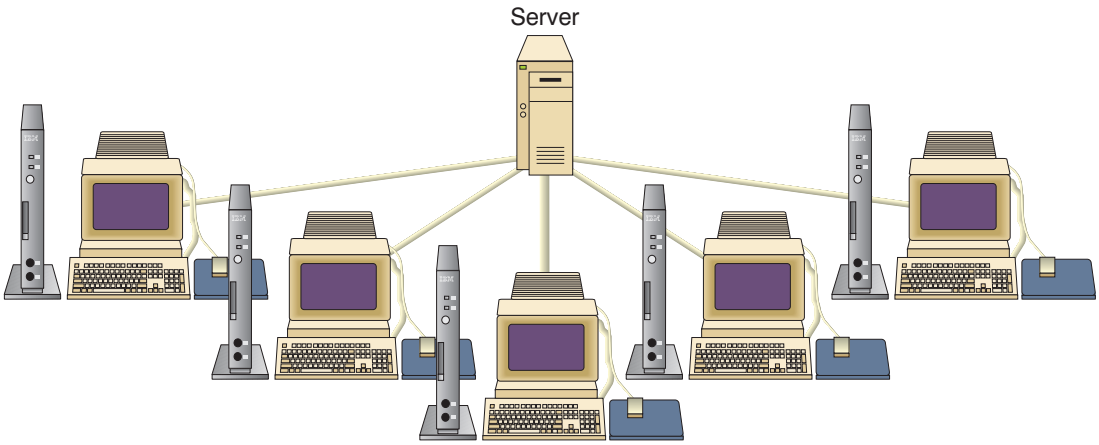

Figure 1. A simple stand-alone network

The connected network is a stand-alone network that is connected to the Internet or an intranet through an IP gateway. This network model assumes that the network computer and server are attached to a larger network and that the services provided by the larger network are therefore available to the server and network computers.

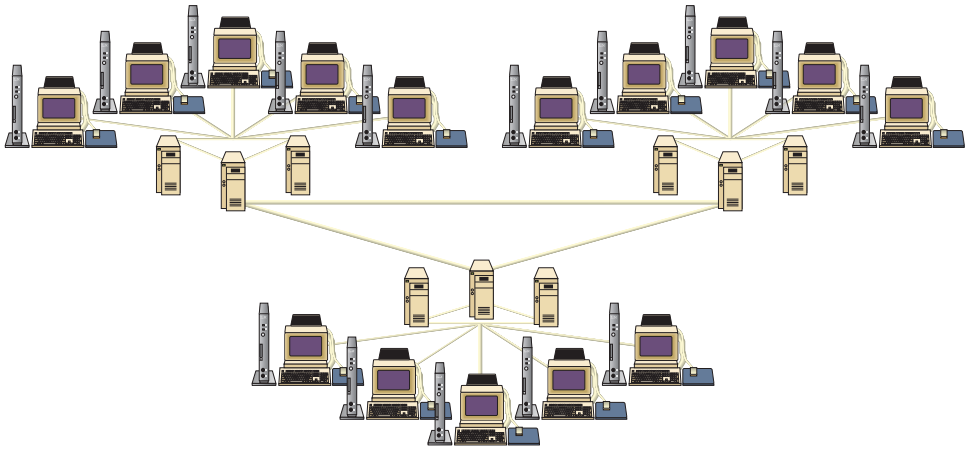

Figure 2. A connected network

In a connected network, the primary server and the network computer can reside on different subnets. When using these more complex configurations, be sure that DHCP relay is configured appropriately between the routers on the different subnets.

Network information gathering

After deciding on a network topology, collect the relevant configuration information for the network computers in the network. Depending on the complexity of the network selected, you will need the following information for each network computer in your environment.

One of the keys to success in a TCP/IP network is that each network computer must have its own unique address. This address is known as the Internet Protocol address, or IP address. The address must be globally unique, meaning that a network computer cannot have the same address as another network computer on the network. In the case of a stand-alone network, the network administrator is solely responsible for ensuring uniqueness. In a connected network, however, the network administrator must work with the owners of the connecting network to ensure that the addresses provided to the network computer are unique. If you want to connect to the Internet, the network addresses and domain names must be officially assigned by a central authority. The authority at the time of this writing is Network Solutions, Inc. Their address is:

```
Network Solutions, Inc.
InterNIC Registration Services
505 Huntmar Park Drive
Herndon, VA  22070
1-703-742-8411
Email:  hostmaster@internic.net
WWW: rs.internic.net
```

IP addresses, which are 32-bit addresses, are generally written as 4 decimal numbers separated by periods, such as 24.14.93.125. Each decimal number represents one byte of the 32-bit IP address, thus each value can range from 0 to 255. This IP address identifies the network and the network computer on that network. (Depending on the size of the network, either the first, second, or third set of numbers in the IP address identifies the network and the remaining numbers represent a particular network computer on that network. The examples in this book are Class C addresses in which the first 3 numbers represent the network and the last number represents the network computer on that network. Therefore, the examples in this book relate to a network of approximately 250 network computers.

The IP address is a software-controlled value that can be changed by the network administrator. However, there is an unchangeable hardware-related address, known as the Media Access Control (MAC) layer address. The MAC address, which uniquely identifies a network adapter card, can be used to uniquely identify a network computer on the network. The only way to change a MAC address for a given network computer is to replace the network adapter with another one. The MAC address is sometimes referred to as the Ethernet address or the token ring address.

Because the IP addresses are difficult for users and administrators to remember, a name can be given to both the network (domain in TCP/IP terminology) and the network computer (host in TCP/IP terminology.) Together, the host and domain names uniquely identify a network computer on the network. To map these names back to addresses, a domain name server, or DNS, is used. In stand-alone networks, the name server is usually the same as the primary server.

In connected networks, a special computer containing two network adapter cards called the router or gateway manages communications between the local network and the Internet or an intranet. In the case of a stand-alone network, the router is usually the same as the primary server.

The following example shows a fictitious network with a domain name of myjavaos.com that has several network computers connected. Two of those network computers have been assigned names: **duke** and **bigblue**. The primary server is called **josserv**, and there is also a nameserver and a router, or gateway.

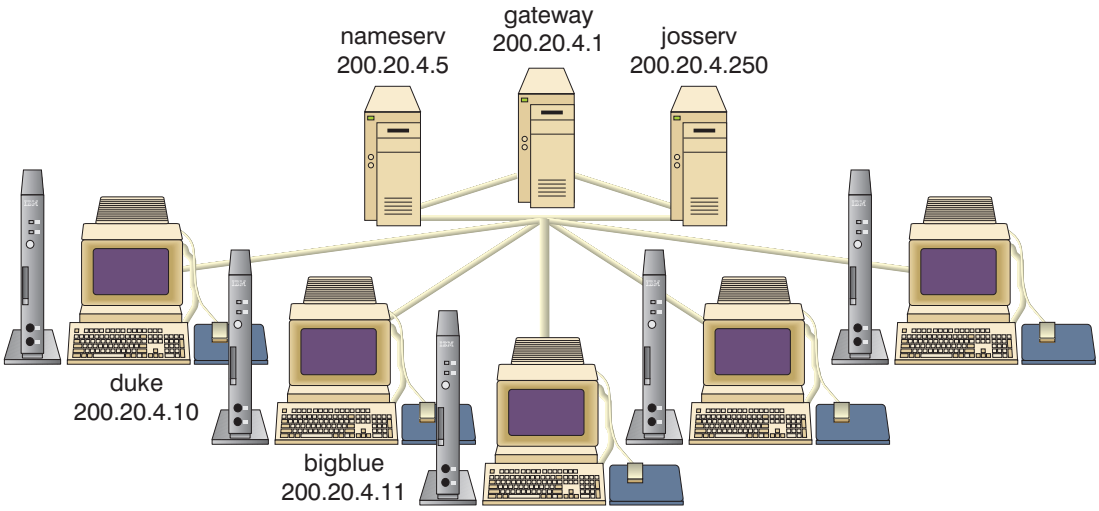

Figure 3. The myjavaos.com network

The following information is needed for planning and configuring the myjavaos.com network:

Host name	duke	bigblue	josserv	gateway	nameserv
MAC address	010060942551f9	010060942551e4	010060942551b2	01006094255111	01006094255109
IP address	200.20.4.10	200.20.4.11	200.20.4.250	200.20.4.1	200.20.4.5
Domain name	myjavaos.com	myjavaos.com	myjavaos.com	myjavos.com	myjavaos.com
Subnet mask	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0
Router or gateway	200.20.4.1	200.20.4.1	200.20.4.1	175.53.9.1	200.20.4.1
Domain name server	200.20.4.5	200.20.4.5	200.20.4.5	200.20.4.5	200.20.4.5

In a more complex example of a connected network, remember that if the network computer is located on a TCP/IP subnet which has a router between it and the boot server, you need to configure the router so that it forwards DHCP broadcast requests from that subnet to the boot server. The **dhcprd** command is generally used on a router to achieve this result.

Networking services and protocols

A network computer, by definition, has no persistent storage and no installed software other than for the BIOS or microcode that allows it to boot from a network server. Before a network computer can be used, the appropriate network services must be set up and configured on the Microsoft Windows NT Server 4.0 system. These services need not exist on a single network server, but are described that way in the examples that follow.

Microsoft Windows NT Server 4.0 does not provide all the essential TCP/IP services required to install and operate JavaOS for Business network computers. Therefore, the DHCP and TCP/IP services provided by IBM Network Station Manager must be installed.

The following networking services and protocols are needed by JavaOS for Business:

- Dynamic Host Configuration Protocol (DHCP) is a TCP/IP protocol that allows a network computer to request an IP address and the name of a file to load. DHCP provides a safe, secure, and reliable method of running a TCP/IP network by centrally managing IP addresses, which prevents address conflicts and conserves unused IP addresses. The Bootstrap Protocol (BOOTP) is a subset of DHCP that is not used by JavaOS for Business.

  The MAC addresses mentioned above can be used to create DHCP entries for assigning IP addresses, thus allowing each network computer to identify itself to the server. Alternately, DHCP can be configured to respond to any client's DHCP request through the use of the SupportUnlistedClients option. This option does not require the enumeration of MAC addresses for all network computers.

- Trivial File Transfer Protocol (TFTP) is a TCP/IP protocol that allows the transfer of files to and from a remote network server. TFTP is used in the initial transfer of the JavaOS for Business boot image from the server to the network computer.

- The Network File System (NFS) is a file sharing protocol designed for diskless workstations and network computers. NFS allows a server to make part of its file system available for use to its clients, that is network computers in the case of JavaOS for Business). Any reference to a file on the network computer results in an NFS request being sent to the server, the operation being performed on the server, and the results returned — all occur transparently to the network computer.

- The Line Printer Daemon (LPD) provides network computers with the ability to print to printers located on servers or other network computers.

Other protocols such as the Hypertext Transfer Protocol (HTTP), time services, and font services can also be provided on the server, depending on the network topology selected and the needs of the users on the network.

Chapter 3. Installing JavaOS for Business on the Server

JavaOS for Business requires specific hardware and software. This section describes the requirements and the steps you must take to install and configure the various components.

Server Hardware Requirements

The minimum hardware requirements for the server are:

- Microsoft Windows NT Server 4.0-compatible computer
- 64 MB RAM (128 MB RAM for better performance)
- 1 GB available disk space
- CD-ROM drive
- Network adapter card
  - Ethernet adapter
  - Token-Ring adapter

Server Software Requirements

The software requirements for the server are:

- Microsoft Windows NT Server 4.0
- Microsoft Windows NT Server 4.0 Service Pack 3
- Sun Java Developers Kit 1.0.2
- Sun Java Developers Kit 1.1.4
- Sun HotJava Browser 1.1.2
- IBM Network Station Manager TCP/IP 1.0
- IBM Network Station Manager 2.5
- IBM Network Station Manager 2.5 Service Pack 1

Service Pack 3 for Microsoft Windows NT Server 4.0 is currently available on the Internet from http://www.microsoft.com/ntserver/. IBM makes no representation or warranty that this service pack will continue to be available from Microsoft.

All of the other required software is included as part of the IBM Adaptation Kit for JavaOS for Business, which is referred to as the product CD in this book.

Some of the software needed is packaged as ZIP files. Use a file extraction utility that supports the creation of subdirectories when you need to extract files from a ZIP file. Currently, you can obtain an unzip utility from Info-Zip at http://www.cdrom.com/pub/infozip/Info-Zip.html. IBM makes no representation or warranty that an unzip utility will continue to be available from Info-Zip.

Server Software Installation

Making JavaOS for Business available to network computers requires the installation of several software packages and appropriate configuration changes on the server. These tasks are described in the following sections.

Installing Microsoft Windows NT Server 4.0

Install Microsoft Windows NT Server 4.0 on your server. You must refer to the Microsoft documentation that covers NT Server Setup in addition to the information provided in this document to ensure a successful installation. Because Microsoft Windows NT Server 4.0 does not always identify all installed hardware adapters on the server, have the latest device drivers for your video, network, and other adapters available on diskette during installation. Use the Have Disk option to load the drivers during installation.

Microsoft Windows NT Server 4.0 must be installed as a domain controller or as a stand-alone server due to the requirements of IBM Network Station Manager. If you intend to use protocols other than the Network File System (NFS), be sure to select the Microsoft Internet Information Server (IIS) to be installed. The TCP/IP protocol should also be selected in the networking options.

Create at least two partitions on your server. One partition, referred to as C: in this documentation, is formatted with the FAT file system and contains the Microsoft Windows NT Server 4.0 code. The other partition, referred to as D: in this documentation, must be formatted with the NTFS file system and contains the server support software and JavaOS for Business software. This partition arrangement can provide better recovery if a problem occurs in Microsoft Windows NT Server 4.0 or with the Windows NT Registry.

Configure TCP/IP appropriately on the server. If your server is acting as a gateway to another network, be sure to select **IP Forwarding**.

Install Microsoft Windows NT Server 4.0 Service Pack 3 on your server.

At this point, you should verify that your TCP/IP configuration is working. The easiest way to verify that your routing and domain name server (DNS) are configured properly is to use the **ping** command to test a connection to another computer on the network. If this command fails, correct your network connection before proceeding. Note that the **ping** command may be insufficient to completely verify your TCP/IP configuration because other considerations could cause other protocols, such as HTTP or TFTP, to fail.

Creating an Emergency Repair Disk

Create an Emergency Repair Disk at this point so that you can restore the Windows NT Registry, if necessary. To create one:

1. Select **Start**, **Programs**, and then **Command Prompt**.
2. Run the **rdisk** program.
3. Select **Update Repair information**.
4. Select **Yes** to continue.
5. Select **Yes** to create an Emergency Repair Disk.

Whenever you add or remove software components in Microsoft Windows NT Server 4.0, it is a good idea to create a new Emergency Repair Disk.

Preparing to Load Server Software

The following server software products are provided on the IBM Adaptation Kit for JavaOS for Business CD in a file called NTSRVBIN.ZIP:

- Sun Java Developer's Kit 1.0.2
- Sun Java Developer's Kit 1.1.4
- Sun HotJava Browser 1.1.2
- IBM Network Station Manager TCP/IP 1.0
- IBM Network Station Manager 2.5
- IBM Network Station Manager 2.5 Service Pack 1
- IBM Network Station Manager DHCP Driver with Fixes

To prepare for the installation of these products, do the following:

1. Insert the IBM Adaptation Kit for JavaOS for Business CD in the CD-ROM drive.
2. Locate the **NTSRVBIN.ZIP** file on the CD.
3. Create a temporary directory on the NTFS drive on the server:

   ```
   d:
   md \temp
   ```

4. Use your file extraction utility to unpack the files in **NTSRVBIN.ZIP** to the temporary directory:

   ```
   cd \temp
   unzip e:\ntsrvbin.zip
   ```

Installing Java Developer's Kit 1.0.2

The IBM TCP/IP Configuration Utility requires that the Java Developer's Kit (JDK) 1.0.2 for Microsoft Windows NT 4.0 be installed on the server. Install it as follows:

1. Locate the self-extracting executable file **JDK102.EXE** in the **JDK102** subdirectory of the temporary directory you created in “Preparing to Load Server Software.”
2. Copy this self-extracting executable file to the root directory of the NTFS drive on which you wish to install JDK 1.0.2 and run it:

   ```
   d:
   cd \temp\jdk102
   copy jdk102.exe d:\
   cd \
   jdk102
   ```

   This file. when unpacked, creates a directory tree called **\java**.
3. After the files are extracted, you can delete the self-extracting executable file from the root directory because it is no longer needed.

   ```
   del jdk102.exe
   ```

To verify that JDK 1.0.2 was installed successfully, run the TicTacToe applet. One way to do this is:

1. Select **Start**, **Programs**, and then **Command Prompt**.
2. Temporarily add the directory containing the JDK executable files to the PATH by using the SET command:

```
set path=d:\java\bin;%path%
```

3. Go to the drive where you installed JDK 1.0.2.
4. Change to the directory containing the TicTacToe sample

```
d:
cd \java\demo\TicTacToe
```

5. Run the sample:

```
appletviewer example1.html
```

If the applet does not run, correct your installation problem before proceeding.

Installing Java Developer's Kit 1.1.4

The tools used to administer JavaOS for Business on the server require Java Developer's Kit (JDK) 1.1.4 to be installed.

1. Locate the **JDK114.ZIP**, file in the temporary directory created in “Preparing to Load Server Software” on page 13.
2. Copy this file to the root directory of the drive where you will install the Java Developer's Kit.

```
copy d:\temp\jdk114\jdk114.zip d:\
```

3. Extract the files using your file extraction utility:

```
d:
cd \
unzip jdk114.zip
```

4. A directory called **\jdk1.1.4** is created on the target drive.
5. After the files have been extracted, you can delete the file from the root directory.

```
del jdk114.zip
```

To verify that JDK 1.1.4 was installed successfully, run the TicTacToe applet. One way to do this is:

1. Select **Start**, **Programs**, and then **Command Prompt**.
2. Temporarily add the directory containing the JDK executable files to the PATH by using the SET command:

```
set path=d:\jdk1.1.4\bin;%path%
```

3. Change to the directory containing the TicTacToe sample:

```
d:
cd \jdk1.1.4\demo\TicTacToe
```

4. Run the sample:

```
appletviewer example1.html
```

Installing HotJava Browser 1.1.2

The JavaOS Configuration Tool, which is used to administer JavaOS for Business network computers, runs in the context of the HotJava Browser. The HotJava Browser 1.1.2 is provided with JavaOS for Business because it supports Java Runtime Environment 1.1.4. Install the HotJava Browser as follows:

1. Copy the file containing the HotJava Browser 1.1.2 (located off of the temporary directory created in

"Preparing to Load Server Software" on page 13) to the root directory of the drive where you want to install the HotJava Browser:

```
copy d:\temp\hj112\hj112.zip d:\
```

2. Extract the files using your file extraction utility:

```
d:
cd \
unzip hj112.zip
```

3. A directory called **\hotjava** is created on the target drive.
4. You can delete the zip file from the root directory when done.

```
del hj112.zip
```

When the HotJava Browser is invoked by the JavaOS Configuration Tool, it stores its properties in the **\javaos\jossrv** directory by default. Do not allow other instances of the HotJava Browser to use these properties.

Preparing for IBM Network Station Manager on National Language Versions

The IBM Network Station Manager requires that there be a group called **Administrators**, in English, with administrator permission. In some national language versions of Microsoft Windows NT Server 4.0 the name of this group is translated causing IBM Network Station Manager not to work as desired. On these versions of Microsoft Windows NT Server 4.0, you must create an **Administrators** group in English as follows:

1. Select **Start**, **Programs**, **Administrative Tools**, and then **User Manager for Domains**.

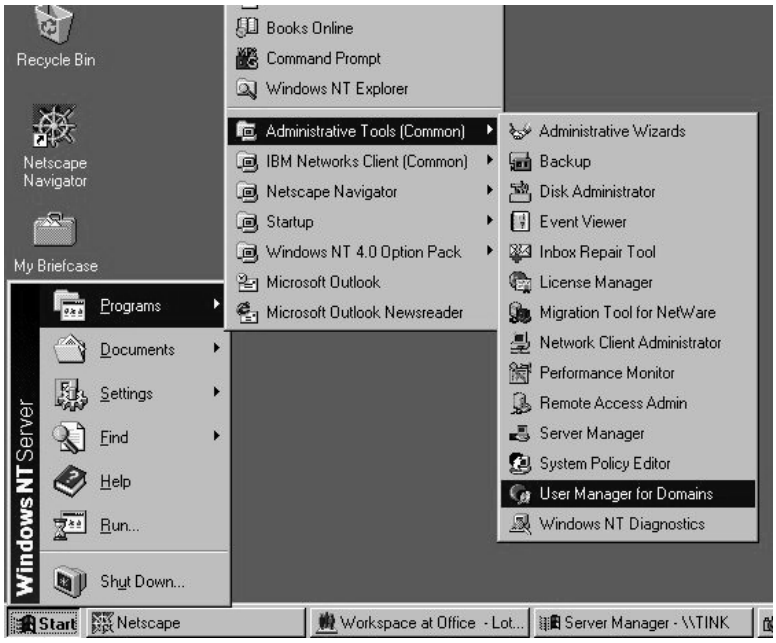

Figure 4. Defining an Administrators group for national language versions

2. Check for a group with the English name of **Administrators**. If one does *not* exist, continue with the following steps.
3. Create a new group called **Administrators**.

4. Give the new group full **Adminstrator** authority.
5. Add the userid for the Administrator to the group.
6. Close the dialog.

Installing the IBM Intermediate Support Driver

The DHCP (Dynamic Host Configuration Protocol) support that is included with Microsoft Windows NT Server 4.0 does not provide all the functions needed by JavaOS for Business. You must install the IBM Intermediate Support Driver to provide the necessary support. The driver must be installed before installing IBM Network Station Manager as follows:

1. Select **Start**, **Settings**, **Control Panel**, **Network** and then **Protocols**.
2. Select **Add** to add a new protocol.
3. Select **Have Disk** when prompted for the location of the protocol files.

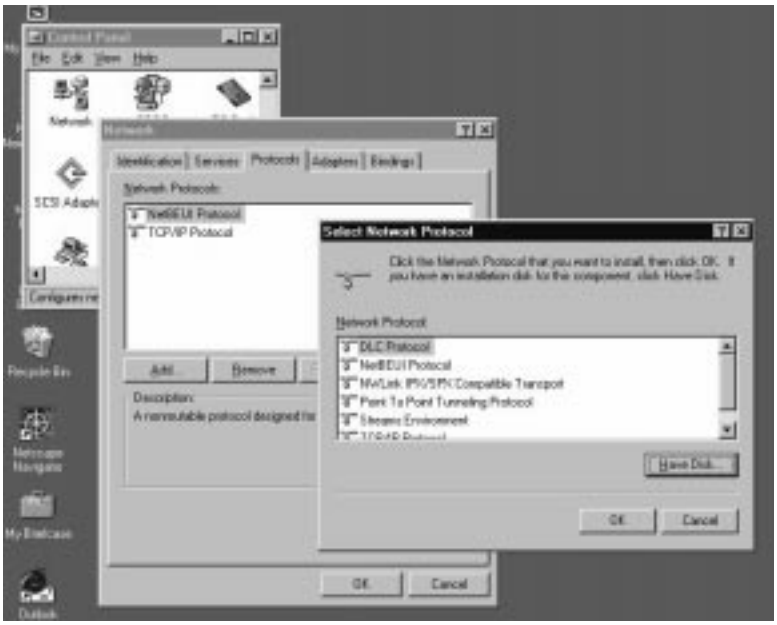

Figure 5. Installing the IBM Intermediate Support Driver

4. In the **Insert Disk** dialog, specify the path for the protocol files. These files were extracted from the Adaptation Kit CD in “Preparing to Load Server Software” on page 13 into this directory:

   ```
   d:\temp\wedge
   ```

5. Select the **IBM Intermediate Support Driver** and then select **OK**.
6. Select **Close**. When prompted to shut down and reboot, select **No**.

After the IBM Intermediate Support Driver is installed, the properties notebook for the TCP/IP protocol is not accessible. If you need access to this notebook, you will need to uninstall the driver, access the notebook, and then reinstall the driver as previously outlined. Instructions for uninstalling the IBM Intermediate Support Driver can be found in Appendix B, “Uninstalling the IBM Intermediate Support Driver” on page 173.

Installing IBM Network Station Manager

The IBM Network Station Manager provides additional function needed by JavaOS for Business to Microsoft Windows NT Server 4.0. To install this support:

1. Select **Start**, **Programs**, and then **Command Prompt**.
2. Run the IBM Network Station Manager installation program from the temporary directory previously created:

   ```
   d:
   cd \temp\nsm
   setup
   ```

3. Select **OK** to confirm the installation.

Disregard any error messages from the installation program indicating that the system is not a Windows NT Server.

4. Select **Next** to continue.
5. Modify the destination directory to be **d:\nstation**.
6. Select **Yes** when prompted to create the destination directory.
7. Select **Next** to begin the installation.

You will get no notification that the installation program has completed. When the installation dialog is dismissed, the installation has completed. Do not shut down the server at this point. (If you do shut down the server, when the system reboots, error messages are displayed during bootup. These error messages will be resolved after the TCP/IP services are installed in "Installing IBM TCP/IP Services."

Installing IBM TCP/IP Services

The TCP/IP support in Microsoft Windows NT Server 4.0 does not contain all the function necessary for JavaOS for Business. To install the time daemon (TIMED), the trivial file transfer protocol daemon (TFTPD), the network file system daemon (NFSD) and PORTMAP, do the following:

1. Select **Start**, **Programs**, and then **Command Prompt**.
2. Change to the directory containing the JDK 1.0.2 executable files:

   ```
   d:
   cd \java\bin
   ```

3. Run the TCP/IP installation program from this directory:

   ```
   d:\temp\nsmtcpip\install
   ```

4. Select **Install TCP/IP Applications** and select **Next**.
5. Select **Yes** if you agree with the terms and conditions of the license agreement.
6. Select **Next** to continue.
7. Select **No** to bypass reading the README for IBM Network Station Manager TCP/IP services. Refer to the README if you encounter problems with IBM Network Station Manager.
8. Change the drive to be the NTFS drive, **D:**
9. Select **Next** to begin the installation.

Disregard the error message indicating that the target file system does not support NT security permissions.

10. Select **Yes** to overwrite the target directory.

11. Select **Next** to confirm installation of the following services:
    - IBM DHCP Server
    - IBM TIMED Server
    - IBM TFTPD Server
    - IBM NFSD Server
    - IBM Portmap Server
12. Select **OK** to confirm that setup has completed.

The IBM Network Station Manager TCP/IP user interface is displayed. (This interface requires JDK 1.0.2 to be available and active on the server.)

13. Select **OK** to postpone the configuration until later.
14. Select **Yes** to confirm saving and exit.
15. Select **OK** to acknowledge that a shut down and reboot are necessary.
16. Shut down and restart the system.
17. After the system reboots, logon as Administrator.

Installing IBM Network Station Manager Service Pack 1

IBM Network Station Manager must have Service Pack 1 applied for use with JavaOS for Business. To install the service pack:

1. Select **Start**, **Programs**, and then **Command Prompt**.
2. Run the Service Pack installation program from the temporary directory:

   ```
   d:
   cd \temp\nsm.sp1
   setup
   ```

3. Select **OK** to confirm that you are the system administrator, that Service Pack 3 for Microsoft Windows NT Server 4.0 is installed, and that IBM Network Station Manager is installed.
4. Select **Next** to begin the installation.
5. Select **OK** to confirm completion.
6. Shut down and restart the system.
7. After the system reboots, logon as Administrator.

JavaOS for Business Software Installation and Initial Configuration

After installing and configuring the server support software, you must install the JavaOS for Business boot image and support files.

Directory Structure Overview

The examples in this book assume that all the software specific to JavaOS for Business is installed in the **d:\javaos** directory on the Microsoft Windows NT 4.0 Server. An overview of that directory structure is provided in Figure 6 on page 19.

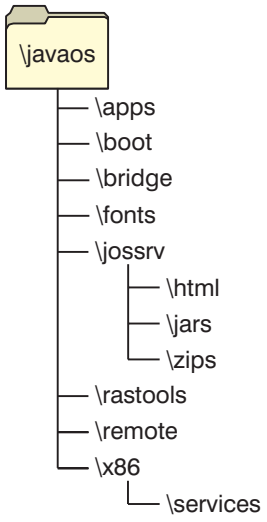

Figure 6. JavaOS for Business network computer binary tree

The **boot** subdirectory contains the bootable image of the JavaOS for Business operating system. This file is the one downloaded to the network computer during boot. This directory also contains the Master Configuration Files (*.MCF), and the PXE boot loader code.

The **apps** subdirectory contains the available desktop applications for the network computers. The hotjava.zip file, which contains the default desktop application and is a modified version of the HotJava Browser 1.1.2, resides in this directory.

The **jossrv** subdirectory contains the JavaOS System Database code, configuration beans for downloadable system services and device drivers, and the JavaOS Configuration Tool (JCT).

The **x86\services** subdirectory contains downloadable system services associated with the Intel® hardware platforms supported by JavaOS for Business.

The **fonts** subdirectory contains the available downloadable fonts

The **remote** subdirectory contains the Input Method Editors (IMEs), locales, code page converters, and keyboards.

Making the files available

A complete JavaOS for Business binary is provided on the IBM Adaptation Kit for JavaOS for Business CD, called **JOSBIND.ZIP** or **JOSBINE.ZIP**. **JOSBIND.ZIP** is the US domestic version; **JOSBINE.ZIP** is the export version. To install the binary image on the Microsoft Windows NT Server 4.0:

1. Insert the IBM Adaptation Kit for JavaOS for Business CD in the CD-ROM drive.
2. Locate the binary image file (**JOSBIND.ZIP** or **JOSBINE.ZIP**) on the CD.
3. Go to the root directory of the NTFS drive and unpack the files using your file extraction utility:

   ```
   d:
   cd \
   unzip e:\josbind.zip
   ```

   The directory structure similar to the one illustrated in Figure 6 is created.

Initial configuration tasks

There are several tasks that must be performed before defining specific users and network computers to JavaOS for Business. These tasks include:

- Configuring the IBM TCP/IP Configuration Utility
- Providing administrator access in national language versions.
- Creating an anonymous NFS user.
- Creating an anonymous Microsoft Windows NT Server 4.0 user
- Configuring IBM Network Station Manager TCP/IP Services.
- Configuring fonts for national language versions.
- Configuring DHCP.
- Starting the JavaOS System Database (JSD) server.

Configuring the IBM TCP/IP Configuration Utility

The shortcut for the IBM TCP/IP Configuration Utility must be updated to use JDK 1.0.2 by doing the following:

1. Select **My Computer**, **C Drive**, **winnt**, **Profiles**, **All Users**, **Start Menu**, **Programs**, and then **Network Station Manager TCP/IP**.
2. Highlight **Configuration Utility**, and click the right mouse button and select **Properties**.
3. Select the **Shortcut** tab, and modify the target field to be:

   ```
   d:\javaos\jossrv\tcpcfg.bat
   ```

4. Select **OK**.

Administrator Access in National Language Versions

In some national language versions of Microsoft Windows NT Server 4.0, the name of the **Administrators** group has been translated. If your national language version has a translated name, do the following to enable the proper operation of IBM Network Station Manager:

1. Select the **My Computer** icon on the desktop.
2. Select the **Drive D:** icon.
3. Highlight the **nstation** folder, but do not open it.
4. Use the right mouse button and select **Properties**, **Security**, and then **Permissions**.
5. Select the **Replace Permissions on Subdirectories** check box.
6. Select the **Replace Permissions on Existing Files** check box.
7. Select **Add**.
8. Select the translated Administrator's group.
9. Select **Full Access** as the type of access.
10. Select **Add**.
11. Select **OK**.

Creating an anonymous NFS User

An anonymous network file system (NFS) user must be defined to allow network computers to access the server's file system during the initial phases of boot processing. To define this user, do the following:

1. Select **Start**, **Programs**, **Network Station Manager TCPIP**, and then **NFS User Configuration**.

Figure 7. Starting the NFS User Configuration Program

2. Enter A to add a user.

Figure 8. Defining an anonymous NFS user

3. Specify a login name of **anonymous**, and press Enter for both the password and verify password.
4. Specify a UID of **60001** and a GID of **60001**. You may leave the Full User Name, Home Directory, and Shell fields blank.

5. Enter Q to quit user configuration and save changes.

Creating an anonymous Microsoft Windows NT Server 4.0 user

An anonymous Microsoft Windows NT Server 4.0 user must also be defined. To define this user, do the following:

1. Logon to the Microsoft Windows NT Server 4.0 as Administrator.
2. Select **Start**, **Programs**, **Administrative Tools (Common)** and then **User Manager for Domains**.

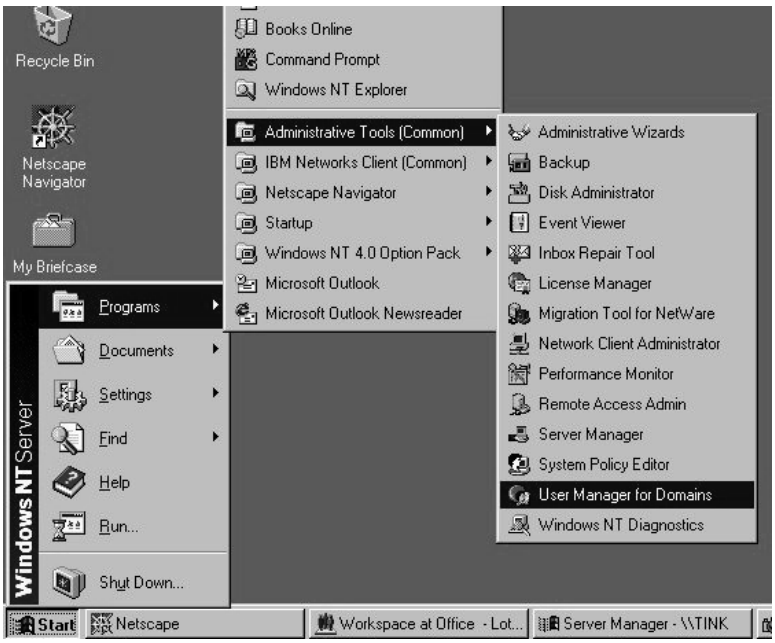

Figure 9. Using the User Manager for Domains

3. Select **User**, and then **New User**.
4. Specify a user name of **anonymous** and leave the password field blank.
5. Clear the check box for **User Must Change Password at Next Logon**.
6. Select **Groups**.
7. Select the **NSMUser** group.
8. Select **Add** to add the user to the NSMUser group.

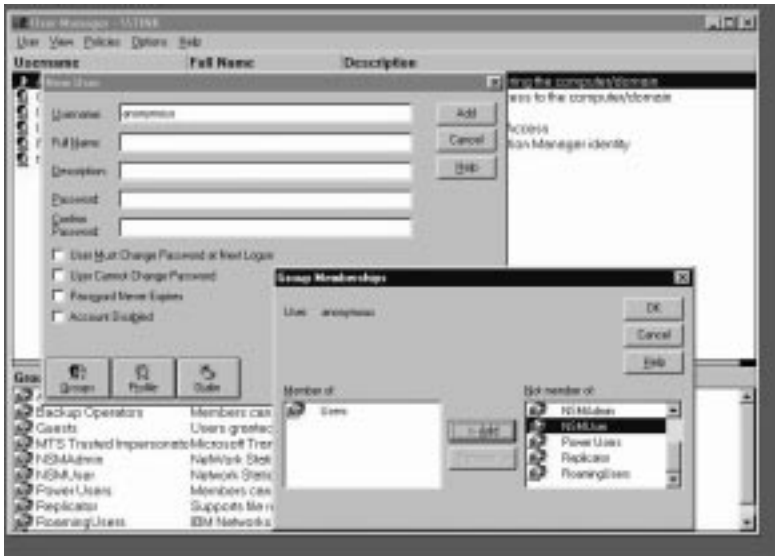

Figure 10. Creating a Windows NT User

9. Select **OK**.
10. Select **ADD** to add the user.
11. Select **Close** to close the window.

Configure IBM Network Station Manager TCP/IP Services

The trivial file transfer protocol daemon (TFTP), network file system daemon (NFSD), and the time daemon (TIMED) must be configured before use. Start the Configuration Utility by selecting **Start**, **Programs**, **Network Station Manager TCPIP**, and then **Configuration Utility**.

Figure 11. Starting the IBM TCP/IP Configuration Utility

Note: The DHCP server is configured by editing the **dhcpsd.cfg** file, not by using the Configuration Utility. A description of this procedure can be found in “Configuring DHCP” on page 31.

TFTP Configuration

To configure the Trivial File Transfer (TFTP) Daemon:

1. Select the **TFTPD** tab in the Configuration Utility.
2. Enable logging for improved diagnostics by selecting the Logging check box.
3. Set the Maximum Segment Size to 512 bytes.
4. Increase the Timeout value to 10 seconds.
5. Change the Retry Limit to 10.
6. Select **Add**.

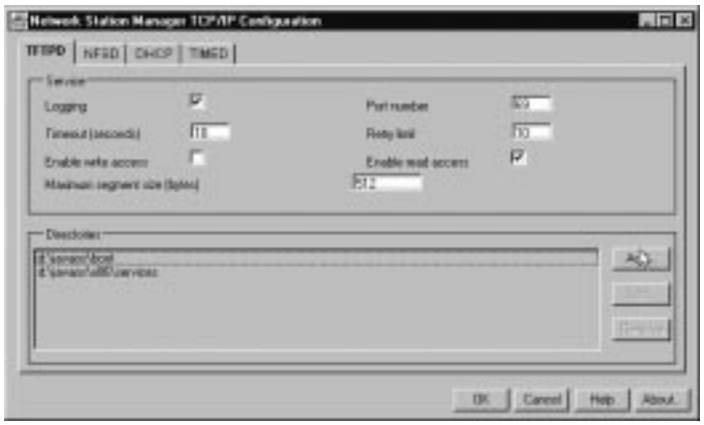

Figure 12. TFTPD Configuration

7. Enter the name of the JavaOS for Business boot directory in the Add TFTPD directory box:

   `d:\javaos\boot`
8. Select **OK**.
9. Select **Add** to add another directory.
10. Enter the name of the Intel® services directory in the Add TFTPD directory box:

    `d:\javaos\x86\services`
11. Select **OK**.

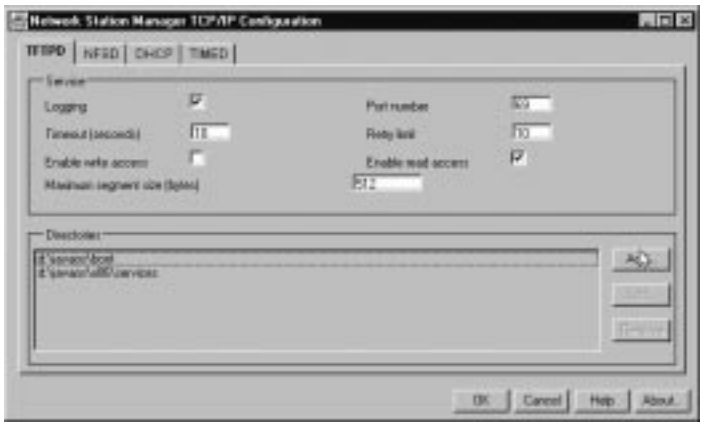

Figure 13. Boot and Services Configured

NFSD Configuration

To configure the Network File System (NFS) daemon:

1. Select the **NFSD** tab in the Configuration Utility.
2. Select the check box for **Use Windows NT Security**.
3. Select **Add** to add export directories for JavaOS for Business.
4. Add the following directory alias for downloadable system services and device drivers associated with the JavaOS for Business operating system:

   directory d:\javaos\x86\services

   alias /x86serv/

   comment JavaOS for Business x86 Services

   access read only

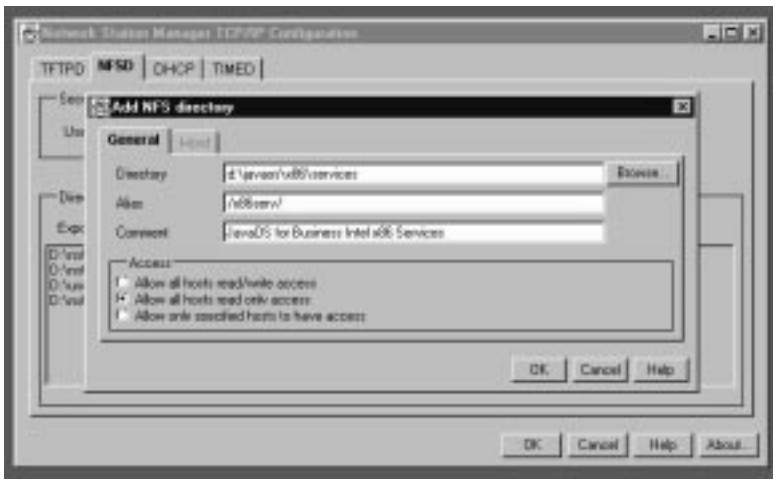

Figure 14. Making system services available

5. Select **OK**.
6. If you have made additional fonts available, set up an alias to the fonts directory:

   directory d:\javaos\fonts

   alias /fonts/

   comment JavaOS for Business Fonts

   access read only
7. Some national language versions of JavaOS for Business provide Input Method Editors (IME). If you have made IMEs or other remote services available to the network computers, define an alias for the remote services:

   directory d:\javaos\remote

   alias /remote/

   comment JavaOS for Business Remote Services

   access read only
8. In addition, you might want to add aliases for applications or other data that you want to share among users. For example, you could set an alias for the HotJava Browser 1.1.4 as follows:

   directory d:\javaos\apps

   alias /hotjava/

   comment JavaOS for Business HotJava Browser 1.1.4

   access read only

Figure 15. Making applications available

9. Select **OK**.

TIMED Configuration

To configure the Time Daemon:

1. Select the **TIMED** tab.
2. Clear the **Adjust to local time** check box, because JavaOS for Business uses GMT.
3. Select **OK** to exit the configuration tool.

Setting Windows NT File System Security Permissions

The **\javaos**, **\javaos\x86**, **\javaos\fonts**, and **\javaos\remote** directories on the server must be made available to network computers using JavaOS for Business.

To set the permissions on the **\javaos** directory for the administrator:

1. Select the **My Computer** icon on the desktop.
2. Select the **Drive D:** icon.
3. Highlight the **JavaOS** folder, but do not open it.
4. Use the right mouse button and select **Properties**, **Security**, and then **Permissions**.

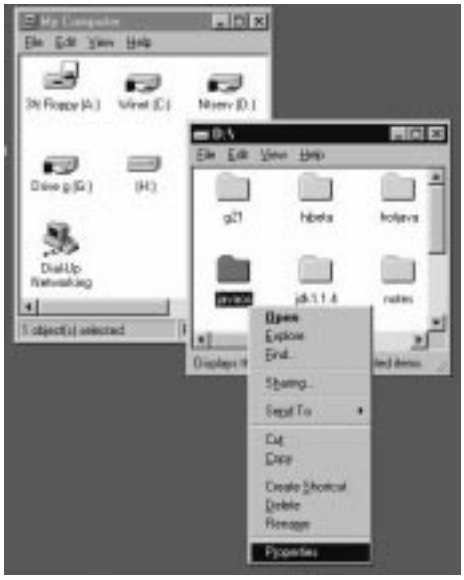

Figure 16. Setting Windows NT File System Security

5. Select the **Replace Permissions on Subdirectories** check box.
6. Select the **Replace Permissions on Existing Files** check box.
7. Select **Add**.
8. Select the Administrator's group.
9. If you have a national language version of Microsoft Windows NT Server 4.0 that has a translated Administrator's group, select the translated group also.
10. Select **Full Control** as the type of access.

Figure 17. Giving administrators access to \javaos

11. Select **Add**.
12. Select **OK**.
13. Select **Everyone**.
14. Select **Remove**.
15. Select **OK**.

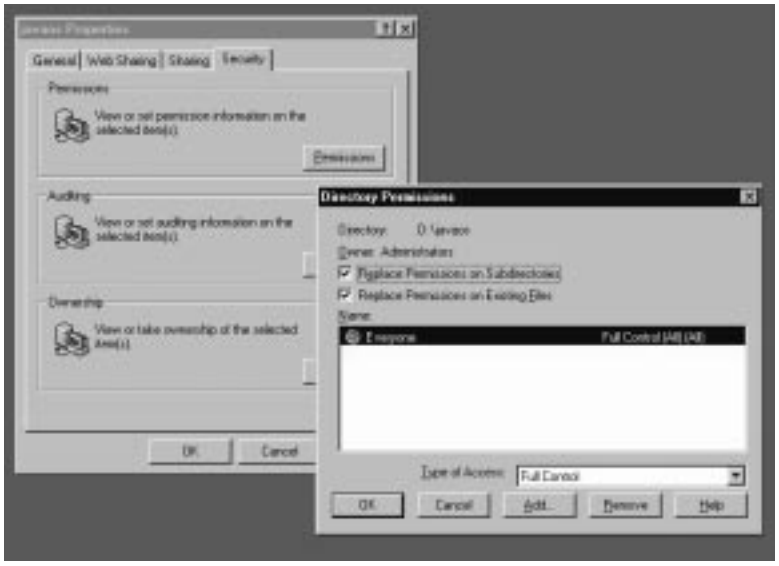

Figure 18. Removing everyone's access to \javaos

16. Select **Yes** to replace security information.
17. Select **OK**.

To set the permissions on the **\javaos\x86** directory:

1. Select the **My Computer** icon on the desktop.
2. Select the **Drive D:** icon.
3. Open the **JavaOS** folder.
4. Highlight the **x86** folder.
5. Use the right mouse button and select **Properties**, **Security**, and then **Permissions**.
6. Select the **Replace Permissions on Subdirectories** check box.
7. Select the **Replace Permissions on Existing Files** check box.
8. Select **Add**.
9. Select the **NSMUser** group.
10. Select **Read** access.
11. Select **Add**.
12. Select **OK**.
13. Select **OK**.
14. Select **Yes** to replace security information.
15. Select **OK**.

Figure 19. Giving the NSMUser group access

To set the permissions on the **\javaos\fonts** directory:

1. Select the **My Computer** icon on the desktop.
2. Select the **Drive D:** icon.
3. Open the **JavaOS** folder.
4. Highlight the **fonts** folder.
5. Use the right mouse button and select **Properties**, **Security**, and then **Permissions**.
6. Select the **Replace Permissions on Subdirectories** check box.
7. Select the **Replace Permissions on Existing Files** check box.

8. Select **Add**.
9. Select the **NSMUser** group.
10. Select **Read** access.
11. Select **Add**.
12. Select **OK**.
13. Select **OK**.
14. Select **Yes** to replace security information.
15. Select **OK**.

To set the permissions on the **\javaos\remote** directory:

1. Select the **My Computer** icon on the desktop.
2. Select the **Drive D:** icon.
3. Open the **JavaOS** folder.
4. Highlight the **remote** folder.
5. Use the right mouse button and select **Properties**, **Security**, and then **Permissions**.
6. Select the **Replace Permissions on Subdirectories** check box.
7. Select the **Replace Permissions on Existing Files** check box.
8. Select **Add**.
9. Select the **NSMUser** group.
10. Select **Read** access.
11. Select **Add**.
12. Select **OK**.
13. Select **OK**.
14. Select **Yes** to replace security information.
15. Select **OK**.

To set the permissions on the **\javaos\apps** directory:

1. Select the **My Computer** icon on the desktop.
2. Select the **Drive D:** icon.
3. Select the **javaos** folder.
4. Highlight the **apps** folder.
5. Use the right mouse button and select **Properties**, **Security**, and then **Permissions**.
6. Select the **Replace Permissions on Subdirectories** check box.
7. Select the **Replace Permissions on Existing Files** check box.
8. Select **Add**.
9. Select the **NSMUser** group.
10. Select **Read** access.

11. Select **Add**.
12. Select **OK**.
13. Select **OK**.
14. Select **Yes** to replace security information.
15. Select **OK**.

Repeat the above steps for any other common application directories.

National Language Font Configuration

The Japanese, Korean, Simplified Chinese, and Traditional Chinese national language versions of JavaOS for Business require some additional font configuration:

1. Select **Start**, **Programs**, and then **Command Prompt**.
2. Change to the **\FONTS\LIB** directory.

   ```
   d:
   cd \javaos\fonts\lib
   ```

3. Based on the national language version, do the appropriate **XCOPY** command:

   Japanese

   ```
   xcopy font.properties.ja font.properties
   ```

   Korean

   ```
   xcopy font.properties.ko font.properties
   ```

   Simplified Chinese

   ```
   xcopy font.properties.zh font.properties
   ```

   Traditional Chinese

   ```
   xcopy font.properties.zh_tw font.properties
   ```

Configuring DHCP

DHCP must be configured with the appropriate options before it is used. This configuration requires that you modify the **dhcpsd.cfg** file, either directly using the text editor of your choice, or indirectly by using the graphical user interface (GUI). For details on using the GUI, refer to the IBM Network Station Manager documentation on the IBM Network Station Manager product CD.

The **dhcpsd.cfg** file resides in the \ibmtcpip\etc directory on the server. Figure 20 on page 32 shows a sample **dhcpsd.cfg** file using the network setup introduced in "Network information gathering" on page 6 for the *myjavaos.com* domain. In general, it is a good idea to start with a working **dhcpsd.cfg** file, make a backup copy, and then make modifications.

```
logFileName d:\joslog\dhcpsd.log
logFileSize 4
numLogFiles 10
logItem ACNTING
logItem EVENT
logItem WARNING
logItem INFO
logItem TRACE
leaseExpireInterval 3 Minutes
leaseTimeDefault 8 Hours
supportBOOTP true
supportUnlistedClients false

subnet 200.20.4.0 255.255.255.192 200.20.4.10-200.20.4.14
{
  Option 15 myjavaos.com
  Option 6 200.20.4.5
  Option 3 200.20.4.1
  Option 1 255.255.255.192
  Option 4 200.20.4.4
  Option 60 PXEClient
  client 1 0004ACEB0160  200.20.4.10 (alias=duke
  {
     Option 67 /javaos/boot/jospxe.bin
  }
  client 1 006094257AF6  200.20.4.11 (alias=bigblue
  {
     Option 67 /javaos/boot/jospxe.bin
  }
}
Vendor JavaOS.Generic
{
  Option  101 "-djavaos.mJSD=server:3333"
}
```

Figure 20. Sample dhcpsd.cfg File

Highlights of the **dhcpsd.cfg** file are explained below.

The subnet line indicates that that IP addresses 200.20.4.10 through 200.20.4.14 are available for network computers. The subnet mask is 255.255.255.192 for the subnet 200.20.4.0.

The options in this file refer to:

Option	Description
1	Subnet mask
3	IP address of the router or gateway
4	IP address of the time server, the Windows NT Server in our examples
6	IP address of the name server
15	Domain name
60	PXE Client
67	Path and file name of the binary boot image for JavaOS for Business

101 Location of the JavaOS System Database (JSD) and the port number to use

Each network computer can be listed in this file with either a client 1 (Ethernet) entry or a client 6 (token-ring) entry. The first client entry shown in Figure 20 on page 32 indicates that the network computer with MAC address 0004ACEB0160 should be assigned an IP address of 200.20.4.10. The comment indicates that the name of this network computer is duke.myjavaos.com, though this is actually set by the name server.

If you would prefer to have the server dynamically assign IP addresses instead of explicitly listing every network computer in this file, set the SupportUnlistedClients option to true. For more information on this option, refer to the TCP/IP documentation on the IBM Network Station Manager product CD.

After you have made modifications to the **dhcpsd.cfg**, you should validate that the file is syntactically correct. The DHCP GUI can be used for this purpose:

1. Select **Start**, **Programs**, **Network Station Manager TCPIP**, and then **Configuration Utility**.
2. Select the **DHCP** tab.
3. Select **Launch...**.
4. Select **File...** and then **Open**.
5. Select **dhcpsd.cfg**.
6. Select **Open**.
7. Correct any errors reported, and then close the utility.

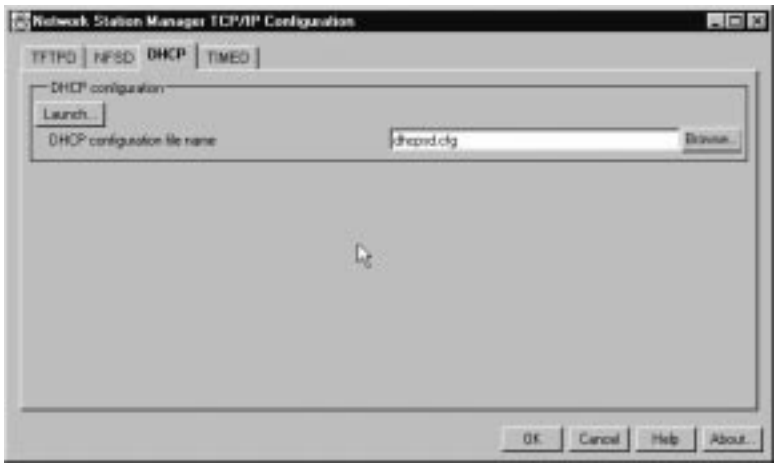

Figure 21. Verifying the correctness of dhcpsd.cfg

Starting the JavaOS System Database (JSD) Server

In order to use the JavaOS Configuration Tool (JCT), the JavaOS System Database (JSD) server must be started. The easiest way to start the JSD server is to create a shortcut on the desktop:

1. Click the right mouse button on the Desktop.
2. Select **New**, and then **Shortcut**.
3. Enter the following command:

   ```
   d:\javaos\jossrv\startjsd.bat
   ```

4. Select **Next**.
5. Enter some appropriate text, such as:

   ```
   Start JSD Server
   ```

6. Select **Finish**.

To start the JSD server, double-click the **Start JSD Server** icon on the desktop. The JSD server should start in a command window.

Note: The JSD is started after logging onto the server. If the user subsequently logs off, the JSD server stops.

To provide additional diagnostic information in the event of a problem with the JSD server, you might wish to modify the buffering of the command window. To do this:

1. Start the JSD server as outlined above.
2. Select the System Menu in the upper-left-hand corner of the window.
3. Select **Properties...**
4. Select **Layout**.
5. Adjust the Screen Buffer Size width to be 90 and the buffer height to be large, such as 1000. This width and length enables scroll bars in the command window and provides a scrollable window of 1000 lines of 90 characters.
6. Select **OK**.

Chapter 4. Network Computer Configuration

This section provides information on validating that the server installation was successful and that JavaOS for Business network computers can boot. Information on the recommended sequence of entering information into the JavaOS System Database is also provided.

Network Computer Hardware Requirements

The minimum hardware requirements for JavaOS for Business network computers are:

- Intel® Pentium®-based system
  - 166 MHz processor
  - PCI bus
  - BIOS dated 4/98 or later
- 32 MB RAM (64 MB RAM recommended for large applications or the running of multiple applets simultaneously)
- Video adapter
  - S3 Trio 64 V+
  - S3 Trio 64 V2/DX
  - Matrox Millenium MGA
- 1 MB Video RAM (2 MB or more recommended for applications requiring high resolution)
- PCI network adapter card such as the IBM Etherjet or Intel® EtherExpress™ Pro 100, that implements the Intel 82558 specification and supports PXE/DHCP boot. (The microcode level should be 1.46 or higher.)

A list of the tested network computer hardware configurations can be found in Appendix A, “Tested Network Computer Hardware Configurations” on page 171.

Notes:

1. In this release of JavaOS for Business, the bus manager supports only 8 cascaded PCI buses in a network computer.
2. Microsoft Windows NT Server 4.0 does not support network file services (NFS) over TCP connections. Network file services must use the UDP protocol when mounting to a Windows NT server.

Configuration Task Overview

The JavaOS System Database (JSD) is a repository for information about the JavaOS for Business operating system and its associated network computers. Information about individual machines, users, device drivers, applications, login attributes, and security settings are all stored in the JSD.

Access to the JavaOS System Database is provided through the JavaOS Configuration Tool, or JCT. The JCT allows the network administrator to add, change, delete, or view information in the JavaOS System Database. The JCT runs in the HotJava Browser 1.1.2 on the same Microsoft Windows NT 4.0 server that contains the JavaOS System Database server.

Users and Groups

Information related to a particular user of JavaOS for Business is contained in a user record in the JSD and possibly one or more group records. Each user of the system has a user record containing information specific about the particular user. In addition, the user must be defined to Microsoft Windows NT Server 4.0 using the same name that is used in the JSD.

Because multiple users of the system might share similar needs, such as access to the same desktop applications or a common keyboard or language, these similar traits can be stored in a group record in the JSD and the appropriate users can be associated with the group. This puts common information in one location, but allows it to effect multiple users.

A user can be a member of more than one group. For instance, a user in the human resources department in France requires access to the corporate jobs database. The user would be a member of the human resources group, with access to the necessary applications, and a member of the French branch office group, with access to the appropriate keyboard, fonts, and French language version of JavaOS for Business.

To expedite defining multiple users to the JSD, you should define the groups the users will belong to first, and then assign the users to the appropriate groups as you define them.

Machines, Profiles, and Platforms

Information about a particular network computer is stored in a machine record. Just as users can be combined into groups, individual network computers can be combined with other similar network computers into a platform, or possibly one or more profiles.

A machine record contains information associated with a particular network computer. This type of record contains information such as the Media Access Control (MAC) address and serial number.

A platform group represents a set of network computers of the same physical type. These may all be a particular model from a particular manufacturer, or all with the same hardware configuration, such as the same network adapter, sound card, display adapter, memory, and so on. A network computer should be a member of only one platform group.

A profile group is used to group a set of network computers in ways not necessarily related to physical type. For example, all the network computers in the Accounting department might be in a special accounting profile. Others located in a particular building or at a particular site may have specific attributes that can be grouped together in a profile. Profiles are optional. A machine does not need to be a member of any profile, or it could be a member of multiple profiles.

Network Computer Configuration

The following network computer configuration tasks are explained in this section:

- Starting the JavaOS Configuration Tool
- Adding a platform definition, if necessary.
- Importing a JSD file.
- Creating a machine record.
- Configuring required services.
- Building the Master Configuration File (MCF) for the machine.

- Updating **dhcpsd.cfg** (if you added or modified a machine record).

Starting the JavaOS Configuration Tool (JCT)

The easiest way to start the JavaOS Configuration Tool is to create a shortcut on the desktop:

1. Click the right mouse button on the Desktop.
2. Select **New**, and then **Shortcut**.
3. Enter the following command:

   ```
   d:\javaos\jossrv\startjct.bat
   ```

4. Select **Next**.
5. Enter some appropriate text, such as:

   ```
   JavaOS Configuration Tool
   ```

6. Select **Finish**.

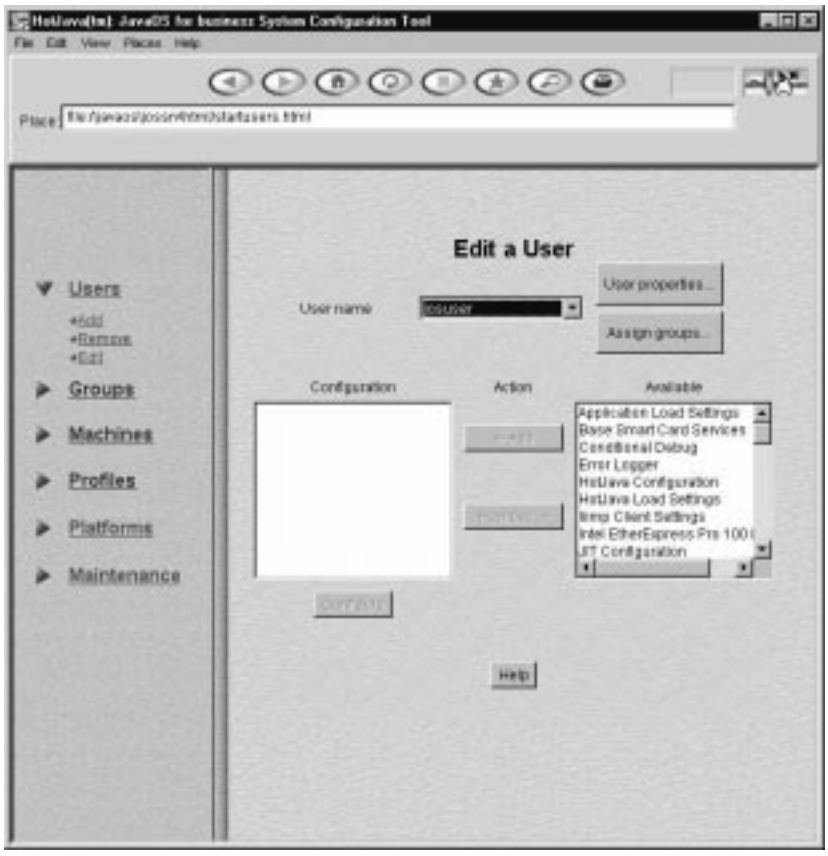

Figure 22. The JavaOS Configuration Tool

To start the JCT, double-click the **JavaOS Configuration Tool** icon on the desktop. The JCT starts the HotJava Browser automatically.

Note: The JCT must run on the same Windows NT Server that contains the JavaOS System Database. Do not configure socks or proxy servers in the HotJava Browser used for the JCT. Do not set the global *HOME* environment variable to point to the property files associated with the HotJava Browser for the JCT.

Creating a platform definition

Perform the following steps to create a new platform definition:

1. Start the JSD Server, if necessary. (See “Starting the JavaOS System Database (JSD) Server” on page 33 for details.)
2. Select **Start**, **Programs**, and then **Command Prompt**.
3. Start the JavaOS Configuration Tool (JCT).
4. Select **Platform**.
5. Select **Add**.
6. Fill in platform-specific information. The *Manufacturer Name* and *Platform Name* must be specified exactly as shown.

   Manufacturer Name com.ibm

   Platform Name x86PCRef

   Description x86 IBM PC Reference
7. Select **Create**.

Once the platform definition has been defined, you need to configure the required services.

Configuring required services

There is a set of required services that must be configured for a JavaOS for Business network computer. These services must be configured in a platform, a profile, or the machine record associated with the network computer. Early services are those services required at boot time. Standard services are used after boot. The required services are:

Early services:

- Network adapter driver, such as the Intel® EtherExpress™ Pro 100 Driver
- Location of JavaOS for Business binary boot image
- Network File System (NFS)
- Keyboard and mouse base driver
- S3 Trio V2+ or Super VGA display driver

Standard services:

- Keyboard driver
- Mouse driver
- NFS Automounter - SERVICES
- Login settings
- Network Station Login (NSL) Authenticator for Login
- Serial driver
- Video driver for resolution (optional)

Note: The serial driver can be established as an early service to make the serial communications port available during boot for debugging.

The tasks below assume that you are still in the JCT. The examples show these definitions being defined at a **Platform** level. These definitions can also be defined at a **Machine** or **Profile** level. The host name for the server used in the examples is **josserv**. You need to change this to the name of your server.

To begin configuring the services, select **Platform** and then**Edit**. Select the platform you wish to modify and then proceed.

Configuring the network adapter driver

1. Select the appropriate driver for the network adapter in the network computer. If you are using Ethernet, select the Intel® EtherExpress Pro™ 100 Driver.
2. Select **Add**.
3. Specify the fully qualified name of the network adapter driver:

   `file:/javaos/x86/services/EEPro100.jar`
4. Select **Save** and then **OK**.

Defining the location of the JavaOS for Business boot image

1. Select **System Image Location**.
2. Select **Add**.
3. Specify the fully qualified location of the binary boot image. For example:

   `file:/javaos/boot/javaos.x86`
4. Select **OK**.

Configuring Network File System (NFS)

1. Select **Network File System (NFS)**.
2. Select **Add**.
3. Specify the fully qualified name for the NFS driver, for example:

   `file:/javaos/x86/services/NfsFileSystem.jar`
4. Select **Save** and then **OK**.

Configuring the base keyboard and mouse driver

1. Select the **Keyboard and Mouse base driver**.
2. Select **Add**.
3. Specify the fully qualified name for the driver:

   `file:/javaos/x86/services/I8042.jar`
4. Select **Save** and then **OK**.

Configuring the video display driver

1. Select the appropriate video display driver, either S3 Trio 64 or Super VGA.
2. Select **Add**.
3. Specify the appropriate video driver.

   Super VGA or Matrox Millennium

   ```
   file:/javaos/x86/services/SVGAFrameBuffer.jar
   ```

   S3 Trio 64

   ```
   file:/javaos/x86/services/S3FrameBuffer.jar
   ```

4. Select **Save** and then **OK**.

Configuring the keyboard driver

To configure the keyboard driver:

1. Select the **Keyboard driver**.
2. Select **Add**.
3. Specify the name for the keyboard driver, for example:

   ```
   file:/SERVICES/I8042.jar
   ```

4. Select **Save** and then **OK**.

Configuring the mouse driver

To configure the mouse driver:

1. Select the **Mouse driver**.
2. Select **Add**.
3. Specify the name for the mouse driver, for example:

   ```
   file:/SERVICES/Uart16550.jar
   ```

4. Select **Save** and then **OK**.

Configuring the video resolution

By default, the video resolution is set to the highest one supported by the video adapter. If you want to specify a specific resolution, do the following:

1. Select the **Video driver**.
2. Select **Add**.
3. Set the desired resolution. A resolution of 1024 x 768 x 8 enables you to verify sufficient communications with the JSD.
4. Select **Save** and then **OK**.

Configuring the NFS Automounter for Services

To configure the NFS Automounter for services:

1. Select **NFS Automounter**.
2. Select **Add**.
3. Fill in the necessary information:

   Client mount type /SERVICES

   Protocol type UDP

   Mount path josserv:/x86serv
4. Select **Save** and then **OK**.

Configuring login settings

To configure login settings:

1. Select **Login Settings**.
2. Select **Add**.
3. Specify the default network login type, such as Network Station Login (NSL).
4. Select **Save** and then **OK**.

Configuring Network Station Login (NSL) Authentication for Login

To configure the NSL Authentication for Login:

1. Select **NSL Authenticator for Login**.
2. Select **Add**.
3. Set the Service URL to the location of the authenticator, for example:

   `file:/SERVICES/NSLAuthenticator.jar`
4. Select **Save** and then **OK**.

Configuring the serial driver

The serial driver provides access to the serial communications port. The serial port can be activated during boot by configuring the serial driver as an early service. Otherwise, you can configure it as a regular service. To configure the serial driver:

1. Select the **Serial driver**.
2. Select **Add**.
3. Specify the name of the serial driver based on its use:

   Early service:

   `file:/javaos/x86/services/Uart16550.jar`

   Standard service:

```
file:/SERVICES/Uart16550.jar
```

4. Select **Save** and then **OK**.

Defining a Machine

In addition to defining a platform, you may want to define individual machines. By making a machine a member of a platform, the machine definition inherits all the configuration aspects of the platform. You can then configure individual services for the specific machine.

Perform the following steps to create a new machine:

1. Start the JCT if necessary.
2. Select **Machines**, and then **Add**.
3. Fill in the appropriate information for the network computer. For example:

   Hardware Type Select a network type of Ethernet or token-ring.

   MAC address Enter the 12-character network adapter address in lowercase.

   Serial Number Optionally specify an identifier for the network computer.

   Location Optionally specify a location for the network computer.
4. Select **Create**.
5. Select **Machines**.
6. Select **Edit**.
7. Select the machine configured using the MAC address.
8. Select **Assign Platform**.
9. Select the appropriate platform for this machine.
10. Select **Save**.

Importing a JSD File

Instead of creating a new platform definition and then configuring all the necessary services, you can import an existing JSD file and then configure it with a minimal amount of effort. JavaOS for Business provides a **x86pcref.jsd** file as a starting point. It assumes the directory structure and alias names outlined in this book. To import this file, or any JSD file:

1. Start the JCT, if necessary.
2. Select **Maintenance**, and then **Import JSD File**.
3. Enter the fully qualified path to the JSD file, or use the **Browse** button to locate it.
4. Select **Save** to load the contents of the JSD file into the JavaOS System Database.

Figure 23 on page 43 shows the import of a JSD file.

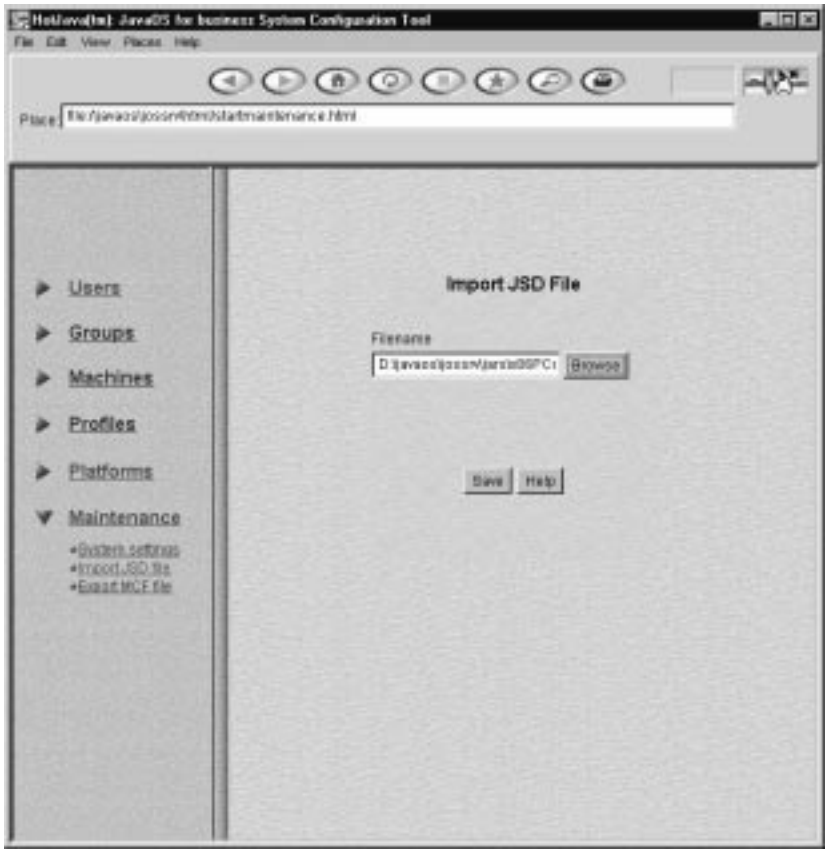

Figure 23. Importing a JSD File

JSD File Format

The JSD file is an ASCII file that contains configuration objects to be placed into the JavaOS System Database. These configuration objects are stored with their default settings. Figure 24 on page 44 shows the content of a sample JSD file.

```
Platform=com.ibm/x86PCref {

   LogonFw.jar
   nslauthenticatorcfg.jar
   I8042cfg.jar
   PCKeyboardcfg.jar
   PS2Mousecfg.jar
   EEPro100cfg.jar
   Uart16550cfg.jar
   NfsFileSystemcfg.jar
   MctCfg.jar
   S3cfg.jar
   SVGAcfg.jar
   SnmpManager.jar
   timezonecfg.jar
   AudioSBCompatiblecfg.jar
   Localecfg.jar
   keytablecfg.jar
   Parallel1284cfg.jar

}
```

Figure 24. A sample JSD File

The generic format of the JSD file consists of one or more entries in the following format:

```
namespace = name {
  name1.jar
  name2.jar
   .
   .
   .
}
```

where:

namespace is one of the following:

- User
- Group
- Profile
- Platform
- Identifier

name is a unique name associated with the **namespace**

name1.jar, *name2.jar*, **...** are the names of the JAR files associated with the configuration objects.

Service configuration

The **x86pcref.jsd** file provides much of the configuration necessary for a collection of JavaOS for Business network computers. However, you must:

1. Add an NFS automounter entry for the **SERVICES** alias.
2. Enable the login authenticator, if desired.
3. If you wish to use the serial port during boot, modify the mount point to specify the fully qualified name of the serial driver.

4. If the network computers reside in a different time zone than the server, configure the time zone.
5. Configure the SNMP manager, if needed.
6. Configure the audio driver, if needed.
7. Set the locale and keyboard if the default language is other than *US English*.

The other services default to the **SERVICES** alias and generally do not require configuration.

After importing the file and making the necessary changes, export the MCF file for the platform. (See "Creating the Master Configuration File.")

Creating the Master Configuration File

When a new machine or platform has been created, you must create a Master Configuration File (MCF) for it. The MCF gives the network computer access to files on the the server in the early stages of boot processing. To create the MCF:

1. Start the JCT if necessary.
2. Select **Maintenance**.
3. Select **Export MCF File**.
4. Select machine based on MAC address or manufacturer and platform.
5. Select **Build**.

Figure 25 shows a sample Master Configuration File.

```
;010060942551f9 configuration file
bootimage=/javaos/boot/javaos.x86
bootservice=file:/javaos/x86/services/EEPro100.jar
bootservice=file:/javaos/x86/services/NfsFileSystem.jar
bootservice=file:/javaos/x86/services/S3FrameBuffer.jar
bootservice=file:/javaos/x86/services/I8042.jar
bootproperties=/javaos/boot/010060942551f9\config.ser
```

Figure 25. A sample MCF for a JavaOS for Business network computer.

After defining machines, you must update the **dhcpsd.cfg** file. See "Configuring DHCP" on page 31 for a description of the contents of this file and instructions on modifying it and validating that it is syntactically correct.

Creating an Emergency Repair Disk

After installing all the necessary server support software for JavaOS for Business, it is a good idea to create an updated Emergency Repair Disk for the Microsoft Windows NT Server 4.0. To create one, follow the steps outlined in "Creating an Emergency Repair Disk" on page 12.

User Configuration

The following user configuration tasks are explained in this section:

- Create one or more groups in the JSD for users with common tasks, applications, or equipment.
- Create a user in Microsoft Windows NT Server 4.0.
- Create a home directory for the user on the server.
- Create an alias for the user's home directory.
- Give the user read/write authority to that user's home directory.
- Create a JavaOS for Business user in the JSD using the same name used for Microsoft Windows NT Server 4.0.
- Add the user to the appropriate groups in the JSD.

In the examples in this document, *josuser*, *josuser2* and similar names are used as sample userids. You can choose the userids for your system.

Detailed information on using the JavaOS Configuration Tool can be found in *JavaOS for Business Network Operations*.

Creating Groups

Use a group when you are adding a number of users with similar characteristics. A group consists of one or more users who use the same applications, require access to a specific printer, or use the same equipment. Groups are defined using the JavaOS Configuration Tool (JCT).

To create a group:

1. Start the JavaOS Configuration Tool, if it is not already started.
2. Select **Groups**.
3. Select **Add**.
4. Fill in the name of the group.
5. Select **Create**.
6. Select **Groups**.
7. Select **Edit**.
8. Select the group to configure, such as **HotJava**.
9. Select the services to configure and then configure them appropriately for the group. See “Configuring required services” on page 49 for a list of the services that must be configured.
10. Select **OK**.

Creating a user on the Microsoft Windows NT Server 4.0

Each JavaOS for Business user must be defined on the Microsoft Windows NT Server 4.0 in addition to being defined in the JavaOS System Database (JSD). To define the user on the Windows NT Server:

1. Select **Start**, **Programs**, **Administrative Tools (Common)**, and **User Manager for Domains**.
2. Select **User**.
3. Select **New User**.
4. Specify the user id and desired initial password for the user.
5. Clear the check box for **User Must Change Password at Next Logon**.
6. Select **Groups**.
7. Select the **NSMUser** group.
8. Select **Add** to add the group.
9. Select **OK**.
10. Select **Add** to add the user.
11. Select **Close**.
12. Select **Exit**.

To define other users with similar attributes, select **Users** and then **Copy** in the User Manager for Domains window.

Create a home directory

Create a home directory for the user on the server:

1. Select **Start**, **Programs**, and then **Command Prompt**
2. Create the home directory based on the userid of the user. For user *josuser*:

```
d:
cd \users
md josuser
```

The organization of directories for users is illustrated here:

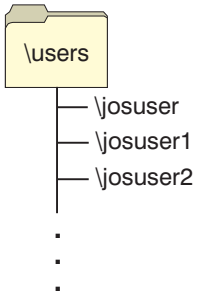

Figure 26. Organization of home directories

Defining an alias for the home directory

Define an alias for the user's home directory:

1. Select **Start**, **Programs**, **Network Station Manager TCP/IP**, and then **Configuration Utility**.
2. Select the **NFSD** tab.
3. Select the check box for **Use Windows NT Security**.
4. Select **Add** to add export directories for JavaOS for Business
5. Add a directory alias for the user's home directory. For the *josuser* user id:

   directory d:\users\josuser

   alias /josuser/

   comment Alias for JOSUSER home directory

   access read/write
6. Select **OK**.

Note: Before you can bring up the **josuser** network computer, you need to either:

- Reboot the server and restart the JSD server, or
- Stop all the services that you have changed (DHCP, TFTP, and so on) and then restart them.

Set Windows NT file permissions for home directory

Set the appropriate file permissions for the user's home directory:

1. Select the **My Computer** icon on the desktop.
2. Select the drive where the user directory resides.
3. Select the directory associated with the user.
4. Use the right mouse button and select **Properties**, **Security**, and then **Permissions**.
5. Select the **Replace Permissions on Subdirectories** check box.
6. Select the **Replace Permissions on Existing Files** check box.
7. Select **Everyone**.
8. Select **Remove** so that no one has access to the directory.
9. Select the **Replace Permissions on Subdirectories** check box.
10. Select the **Replace Permissions on Existing Files** check box.
11. Select **Add**.
12. Select **Show Users** and then select the appropriate user, such as *josuser*, the anonymous user id, and **Administrator**.
13. Select **Add**.
14. Select **Full Control** as the type of access.
15. Select **OK**.
16. Select **OK**.

17. Select **Yes** to replace security information.
18. Select **OK**.

Create a JavaOS for Business user

Perform the following steps to define the user previously created in Microsoft Windows NT Server 4.0 to JavaOS for Business:

1. Start the JavaOS Configuration Tool (JCT), if needed.
2. Select **Users**.
3. Select **Add**.
4. Fill in the User name, Full name, and Phone number fields. The information in these fields must match those specified in the Microsoft Windows NT Server 4.0 settings in “Creating a user on the Microsoft Windows NT Server 4.0” on page 47.
5. Select **Create**.
6. Select **Users**.
7. Select **Edit**.
8. Select the user to configure.
9. Follow the steps in “Configuring required services” to configure the required services.

Configuring required services

When a JavaOS for Business user has been created in the JSD, configure the minimum required services. These services must be configured in either the user record or in the record of a group that the user is a member of. The required services include:

- Establishing the home directory.
- Setting up a default desktop application.

In addition, some services previously configured in a platform or profile record can be overridden here. This makes it possible for a user to have a specific video resolution set, for example. Also, services which have not previously been configured can be established at a user level. For example, a proxy server could be set for a user or a group of users.

The tasks below assume that you are still in the JCT. Select **Groups**, and then **Edit**, and then proceed

Establishing the home directory

1. Select **NFS Automounter**.
2. Select **Add**.
3. Complete the necessary information for user *josuser*:

   Client mount name /HOME

   Protocol type UDP

   Mount path josserv:/josuser

Defining the HotJava browser as the default desktop application

1. Select **NFS Automounter**.
2. Select **Add**.
3. Complete the necessary information. For the HotJava Browser 1.1.4, possible values are:

   Client mount name /HOTJAVA

   Protocol type UDP

   Mount path josserv:/hotjava
4. Select **Save**.
5. Select **OK**.
6. Select **HotJava Configuration**.
7. Select **Add**.
8. Set the location of the default home page. For instance:

   `http://www.ibm.com/Java/`
9. Select **Save**.
10. Select **OK**.
11. Select **HotJava Load Settings**.
12. Select **Add**.
13. Set the location of the ZIP file

    `file:/HOTJAVA/hotjava.zip`
14. Select **Save**.
15. Select **OK**.

Choosing a proxy server

If your network is located behind a firewall, you will need to define a proxy server to allow users access to the Internet.

1. Select Proxies settings.
2. Select **Add**.
3. Enter the URL for the proxy server.
4. Select **Save**.
5. Select **OK**.

Creating the Master Configuration File

When a new machine or platform has been created, you must create a Master Configuration File (MCF) for it. The MCF gives the network computer access to files on the the server in the early stages of boot processing. To create the MCF:

1. Start the JCT if necessary.

2. Select **Maintenance**.
3. Select **Export MCF File**.
4. Select machine based on MAC address or manufacturer and platform.
5. Select **Build**.

Figure 25 on page 45 shows a sample Master Configuration File where the serial driver is configured as an early service.

```
;010060942551f9 configuration file
bootimage=/javaos/boot/javaos.x86
bootservice=file:/javaos/x86/services/EEPro100.jar
bootservice=file:/javaos/x86/services/NfsFileSystem.jar
bootservice=file:/javaos/x86/services/S3FrameBuffer.jar
bootservice=file:/javaos/x86/services/I8042.jar
bootservice=file:/javaos/x86/services/Uart16550.jar
bootproperties=/javaos/boot/010060942551f9\config.ser
```

Figure 27. A sample MCF with the serial driver configured.

After defining machines, you must update the **dhcpsd.cfg** file. See "Configuring DHCP" on page 31 for a description of the contents of this file and instructions on modifying it and validating that it is syntactically correct.

Printing

Network computers running JavaOS for Business can print to printers attached in the following ways:

- Print to a printer attached to the parallel port of the network computer.
- Print to a printer attached to a server.
- Print to a printer attached to another network computer running JavaOS for Business.
- Print from a server to a printer attached to the parallel port of the network computer.

Details on the configuration objects needed can be found in Chapter 17, "Specifying printing information" on page 87.

Printing to a locally attached printer

To print to a printer attached to the parallel port of the network computer requires that the **Parallel Port** and the **Parallel Printer Configuration** objects be configured.

Printing to a printer attached to a server.

To print to a printer attached to a server requires that the **Parallel Port**, **Parallel Printer Configuration**, and **Remote Printer** configuration objects be configured. In addition, the Line Printer Daemon (LPD) must be started on the server.

Perform the following tasks to set up a remote printer on Microsoft Windows NT Server 4.0:

- Install Simple TCP/IP Services and Microsoft TCP/IP Printing Services.
- Configure both services to start automatically.
- Define a printer queue on the server.

- Give the desired network computers access to the printer.

Installing TCP/IP Printing Services: Providing remote printer access from Microsoft Windows NT Server 4.0 requires that both the **Simple TCP/IP Services** and the **Microsoft TCP/IP Printing Services** are installed. To check whether they are installed and install them if necessary:

1. Select **Start**, **Settings**, **Control Panel**, and then **Network**.
2. Select the **Services** tab.
3. Scroll through the list of services. If **Simple TCP/IP Services** and **Microsoft TCP/IP Printing Services** are not installed, install them by continuing with the following steps.
4. Select **Add** to add a service.
5. Select **Simple TCP/IP Services** and select **Add**.
6. Select **Microsoft TCP/IP Printing Services** and select **Add**.
7. Select **OK** to close the dialog.
8. Shut down and reboot the server when requested.

Configuring TCP/IP Printing Services: After the server reboots, logon as *Administrator* and then configure **Microsoft TCP/IP Printing Services** to start automatically.

1. Select **Start**, **Settings**, **Control Panel**, and then **Services**.
2. Scroll through the list and locate **Microsoft TCP/IP Printing Services**.
3. Press **Startup...** to adjust the startup properties.
4. Select **Automatic** to have the **Microsoft TCP/IP Printing Services** start automatically.
5. Press **OK** and close the dialog.

Defining a Remote Printer: To define a remote printer in Microsoft Windows NT Server 4.0:

1. Select **My Computer**, **Printers**, and then **Add Printer**.
2. Select **Add Printer** and press **Next**.
3. Select the check box associated with the port the printer is connected to, such as LPT1. Press **Next**.
4. Select the appropriate manufacturer and then the appropriate printer. Press **Next**.
5. Select **Yes** if you want this printer to be the default printer on the server. Otherwise select **No**. Press **Next**.
6. Select **Shared** and then select the operating systems that require access to the printer. Use **Windows NT 4.0 MIPS** for JavaOS for Business network computers.
7. Choose a share name. The share name is used as the *printer name* when installing a remote printer on JavaOS for Business. Press **Next**.
8. Select **Yes** to print a test page. If the test page does not print correctly, correct your printer configuration problem before proceeding. Press **Next**.

If you already have a local printer defined on the Windows NT server, you can make it available to network computers as follows:

1. Select **My Computer**, **Printers**, and then highlight the printer you want to make available.
2. Select the printer with the right mouse button and select **Properties**.
3. Select **Sharing**, press **Shared**, and then specify the share name for the printer.

Giving Users Access to the Printer: After the printer has been defined to Microsoft Windows NT Server 4.0 as a shared printer, you must give the users access to the printer. Access is granted by:

1. Select **My Computer**, **Printers**, and then highlight the printer you want to give users access to.
2. Select the printer with the right mouse button and select **Properties**.
3. Select the **Security** tab.
4. Press the **Permissions** button.
5. Give the following users **Full Control** access to the printer:
   - Administrators
   - Translated administrators group, if on a national language version of Windows NT
   - NSMUser group

Printing to a printer attached to another network computer

To print to a printer attached to another network computer running JavaOS for Business requires that the parallel port, parallel printer, and the remote printer configuration objects be configured. In addition, the network computer with the printer attached needs to be suitably configured.

Testing

At this point, all the appropriate software should be installed on the Microsoft Windows NT Server 4.0 and all the necessary configuration should be completed. To verify that a JavaOS for Business network computer can boot from the server, you'll want to turn on one of the network computers you have configured. A possible sequence is:

1. Verify that all the servers are running and that they are cabled to the network.
2. Connect the monitor, keyboard, mouse, power, and network cables for each network computer.
3. Start the JavaOS System Database (JSD) server if it is not already started. (See “Starting the JavaOS System Database (JSD) Server” on page 33.)
4. Turn on the network computer. Depending on the network computer, and how it is configured, you might see introductory information from the manufacturer followed by a memory test or other diagnostics.
5. The network computer then sends a DHCP request with its MAC address out to the network.
6. One of the servers on the network responds to the DHCP request based on the MAC address specified and you should see TFTP of the JavaOS for Business boot loader begin. If not, the transfer fails with an error message.
7. After the download of the boot loader, the Master Configuration File (MCF) is downloaded using TFTP along with the JavaOS for Business binary boot image and the downloadable system services specified in the MCF.
8. If logon authentication is enabled in JavaOS for Business, you will be prompted for a user id and password.
9. After a successful logon, the main application specified for the user is loaded. By default, this is the HotJava Browser 1.1.4.

Installation Troubleshooting and Common Problems

If you encounter problems during the inital installation and configuration of JavaOS for Business, you can find additional debugging information and solutions to common problems in the following places.

DHCP Error Log

The path to the DHCP error log is defined in **d:\ibmtcpip\etc\dhcpsd.cfg**. Look here if network computers can not boot from the server.

TFTP Error Log

The TFTP Error Log can be found by:

1. Select **Start**, **Programs**, **Administrative Tools**, **Event Viewer**, **Log**, and then **Application**.
2. Check for entries with the string `TFTP` in them.

NFS Error Log

The NFS Error Log can be found by:

1. Select **Start**, **Programs**, **Administrative Tools**, **Event Viewer**, **Log**, and then **Application**.
2. Check for entries with the string `NFS` in them.

Boot Failures Without DHCP, TFTP, or NFS Errors

If a JavaOS for Business network computer does not boot and you do not find any DHCP, TFTP, or NFS related errors in the respective error logs, try the following:

1. Verify that the DHCP, TFTP, and NFS services are started on the server.
2. Recreate the Master Configuration File (MCF) associated with the network computer or platform.
3. Check the Windows NT file system permissions. Re-apply the appropriate permissions to the boot and services directories, and the user's home directory.

If a null pointer exception is taken after the network computer logs on but before the HotJava Browser starts, verify that the HotJava Configuration entry has been established for the user, group, profile, or platform associated with the failure.

Part 2. Configuration Planning

You configure JavaOS for Business using several methods:

- Editing and changing system files
- Running a graphical user interface tool, the JavaOS Configuration Tool.

This part of the book helps you make the decisions about configuration. It describes the relationships between the configuration choices you can make.

The second part of this book, Part 3, " JavaOS for Business system administration" on page 93, tells how to use the JavaOS Configuration Tool for configuration and for system administration.

Chapter 5. JavaOS System Database and JavaOS Configuration Tool

The JavaOS System Database and JavaOS Configuration Tool are the keys to system customizing, configuration, and administration. The JavaOS System Database is the system repository that contains the information about the JavaOS for Business system. This includes information about the network and the system itself; its machines, users, device drivers, applications, login information, security information, and so on.

You use the JavaOS Configuration Tool (JCT) to access, add, change, and delete information in the JavaOS System Database. The JavaOS Configuration Tool, a Java applet that runs in the HotJava browser, lets you easily access the information in the JavaOS System Database. To start the JCT, start the HotJava browser and then specify the URL for the JCT.

Configure the server copy of the HotJava browser with the same parameters as the network computer version of the browser as described in Chapter 8, “Changing and configuring the HotJava browser” on page 67.

The JCT is described in detail in Chapter 21, “JavaOS Configuration Tool edit panels” on page 101. The following is a quick overview of the main sections of the JCT:

Users Use this selection to define users and the system resources available to each user.

Groups Use this selection to define groups of users who have the same system resources.

Machines The Machines selection pertains to one particular network computer. You can define the login rules and system resources for the machine itself.

Profiles The Profiles selection lets you manage several machines with similar attributes. For example, you could have a workgroup that uses machines with have similar characteristics. You can define and maintain a profile for these machines.

Platforms The Platforms selection lets you manage a set of network computers that are physically similar in some way. For example, you could have a set of identical machines that you purchased from one computer manufacturer. You can define and maintain a platform for these machines.

Maintenance The Maintenance selection is a set of utilities for maintaining the system. See Chapter 23, “System Maintenance” on page 117 for details about these selections.

For more in-depth information about the five selections, see “JavaOS Configuration Tool main panel selections and terminology” on page 95.

After you add your various users, groups, machines, profiles, and platforms, you can then edit each of them and define the specific attributes they should have. When you click **edit**, a panel displays a list of available services for the item you are editing. The Available list includes a variety of services you can add, such as device drivers, applications, login settings, printers, multimedia, languages, proxy servers, and so on.

The information you define for the users, groups, and so on is stored in the JavaOS System Database and is readily available through the JavaOS Configuration Tool.

The JavaOS System Database provides an effective way of storing system data that you can easily manage. You can customize your system and replace the JavaOS System Database with your own repository.

Similarly, the JavaOS Configuration Tool is a useful tool to access and manage the information in the JavaOS System Database. You can also replace the JavaOS Configuration Tool with your own method of accessing, adding, changing, and deleting the system repository information.

Chapter 6. Customizing login processing

There are several ways you can customize login processing on the system.

You can specify what type of authentication you want to use for your network computers and users. For example, if your network computers have smart card readers, you can require smart cards for login. You can also specify the default network login type and default network login server. You can set up specific network computers that let anyone use the computer without having to use a user ID and password. For example, you might want a set of network computers running a new product demo or network computers that operate like a kiosk to provide information about an event people are attending. You can also set inactivity time-out values.

The topics below describe different aspects of the login process and how you can customize login.

Network computers and smart cards

Contact your IBM representative regarding the availability of the Smart Card Toolkit. To contact IBM, see the IBM Web site:

`http://www.ibm.com`

If a network computer has a smart card reader, users can use smart cards when logging into JavaOS for Business. The smart card provides additional security in accessing the system. When defining how you want login processing to work, you define security and login attributes for the network computer.

In addition, the smart cards you use can be configured in different ways so that some cards can be used only in specific network computers. For example, you might have network computers available for your executives that are defined with a higher level of security than the computers your sales people work with. Accordingly, the smart cards for the executives and the cards for the sales people would be configured with a level of security that matches the specific network computers each uses. Therefore, a sales person's smart card would not work in an executive's network computer.

If you use smart cards, you can optionally use a personal identification number (PIN) with the smart card. The smart card can be configured to require the use of a PIN. The use of smart cards and their PINs is based on the OpenCard Framework. JavaOS for Business supports the OpenCard Framework. For more information about the framework, see the following Web site:

`http://www.opencard.org/OCF/1.0/nc`

Similarly, when you define your network computers, you can specify whether the network computer requires the use of a smart card and PIN. When a user logs in, the system checks the PIN policy of both the smart card and the network computer. Four scenarios can occur:

- If neither the smart card or network computer requires a PIN, a PIN is not necessary.
- If both the smart card and network computer require a PIN, a PIN must be used.
- If the smart card is configured for a PIN, but the network computer does not require one, the user can still use the smart card. The system prompts the user for a PIN.
- If the smart card is not configured for a PIN, but the network computer requires a PIN, login processing fails. The system displays a panel requesting a PIN, indicating the network computer requires a PIN. In this case, the user is not using the right smart card for the network computer. For example, the network computer might require the executive's smart card, which is configured for a specific security level. If you use a sales person's smart card in the network computer instead, login would fail.

Login scenario

The following highlights the login process from a network computer.

1. The network computer system is powered on. The system displays the initial panel.
2. JavaOS for Business checks the smart card usage for the network computer.
   a. If smart cards are not required at the network computer, JavaOS for Business prompts the user for a network user ID, server, and password.
   b. If smart cards are required at the network computer:
      1) A panel is displayed asking the user to insert a smart card.
      2) The system checks whether the smart card or the network computer requires a personal identification number (PIN). If a PIN is required, the system prompts the user for one. The PIN is not displayed. The system compares the PIN that the user enters to the PIN stored on the smart card. If the PIN is correct, the system prompts the user for a network user ID, password, and server. If the user ID, password, or server are stored on the smart card, the system displays them on the panel. The user can use the displayed values or change them. The user can select one of the listed servers. If the PIN is not correct, the system asks the user to re-enter the PIN. If the user exceeds the number of attempts allowed for entering a PIN, the initial panel is displayed again and the user must start the login process again.
      3) If the system cannot read the smart card or the user has exceeded the number of attempts allowed, the system asks the user to remove the smart card. When the user removes the card, the system displays the Insert smart card panel.
3. JavaOS for Business authenticates the user with the network. If the user ID or password is incorrect, the system prompts the user for the information again. The number of times the user is allowed to enter the information depends on the server.
4. When login is successful, the main application is loaded and the user can start using the system.

Setting up authentication

There are two methods the system uses to authenticate users. The first method is local authentication, which is authentication at the network computer itself. Local authentication is using a network computer with a smart card reader and using smart cards at that computer.

The second method is remote authentication, which is authentication on the network itself.

You could also replace the local or remote authenticator with your own authenticator. See *JavaOS for Business Porting Guide* for the steps to take to replace an authenticator.

The following topics describe how to set up the authentication methods.

Setting up local authentication

To use local authentication, the network computer must have a smart card reader and any user using the network computer must use a smart card.

Note: If you use local authentication, you cannot automatically log in as a guest user.

If you do not use local authentication, the system displays the Network Login panel on the network computer and the user enters a user ID and password and selects a server.

You set up local authentication using the Login Settings in the JavaOS Configuration Tool. You can specify Login Settings for a machine, profile, or platform. On the Login Settings panel, select the Use Workstation Login check box to specify you want to activate local authentication. The authentication method is smart cards. Figure 28 shows the Use workstation login section of the Login Settings panel.

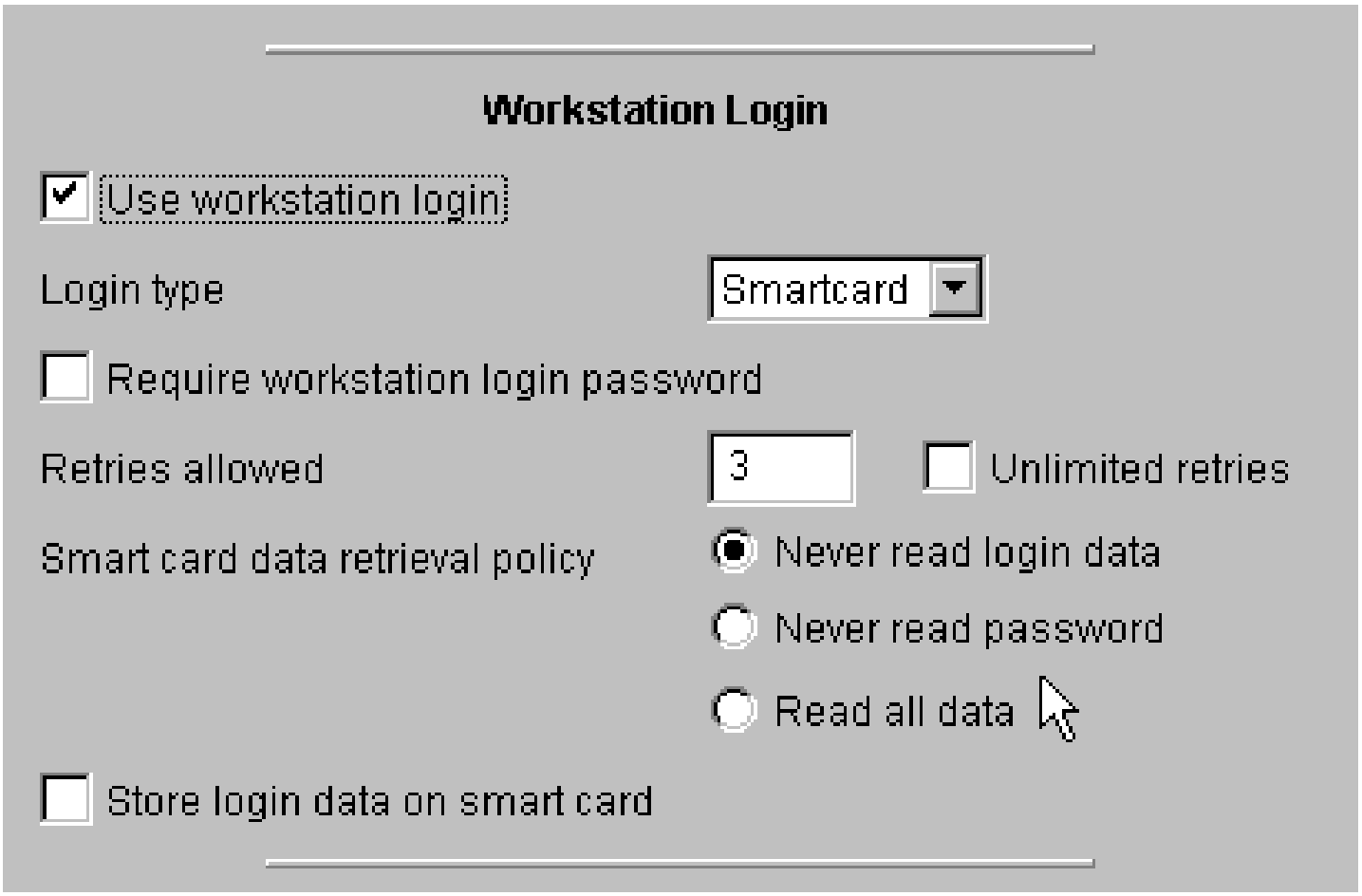

Figure 28. The Use workstation Login section of the Login panel

Select the Require Workstation Login Password check box to require the use of a personal identification number (PIN). You can also choose whether a user has a limited number of attempts to specify a PIN or has an unlimited number of tries.

Note: Some smart cards are defined with a maximum number of retries and disable the card when the limit is reached.

Reading from a smart card

You can specify what data, if any, the system reads from the smart card itself:

- Never read login data - the system does not read any data from the smart card
- Never read password - the system reads the login type, login server, and user ID from the smart card and displays the information on the login panel when the user logs in. The system does not read the password from the smart card.

  The “Never read password” setting forces users to always enter a password regardless of the smart card used. Even if the smart card contains a password, the system does not read it.
- Read all data - the system reads all the data on the smart card, including login types, login servers, user ID, and password. It displays the information on the login panel.

  If you use the “Read all data” setting, you should consider having the system always prompt users for a PIN for security reasons. If someone loses a smart card, this helps prevent someone from using the smart card on the network computer.

Storing data on a smart card

JavaOS for Business lets you store the login data that was used for a successful login on the smart card. JavaOS for Business writes the login type, login server, user ID, and password to the card. This section describes how JavaOS for Business handles writing information on a smart card:

1. When a user logs in, the system displays a list box with the available login servers. The first server displayed in the list is the server you most recently logged in with.
2. On a smart card, the system stores only one user ID per server. If a user uses a different user ID for a server, the system writes the user ID back to the smart card after a successful login. The system replaces the original user ID on the smart card with the ID the user logged in with.
3. When a user specifies a server that is not stored on the smart card, the system writes the server information to the smart card. If there is not sufficient space on the card, the system overwrites information on the smart card. The system overwrites the least recently used entry.
4. At some point, the system administrator might reset a user's password. When the user successfully logs in using the new password, the system writes the new password on the smart card. The system replaces the old password with the new one.

The ability to store information on the smart card is helpful if the user logs in to the same network computer on a regular basis. The stored information is automatically displayed on the login panel when the user logs in. The storing of information is also useful if a user logs into a variety of network computers and servers. JavaOS for Business displays the list of servers on the login panel and the user can easily select one.

To write login data on the smart card, select the Store Login Data on Smartcard check box.

Setting up remote authentication

Remote authentication pertains to the user logging into the network itself. On the Login Settings panel, specify the:

- Default network login type, which is the protocol the remote authenticator uses to authenticate the user ID and password. The default protocol is network station login (NSL).

  JavaOS for Business provides NSL, although you can add your own remote authenticator to the system. For more information, see *JavaOS for Business Porting Guide* . You can use one login type per server. Also, for smart cards, you can use only one request per server.

- Default network login server. This is the TCP/IP hostname of the default server to use when a user logs on.

When a user logs in, the login panel does not display the login type at the network computer. Login does display the default login server. If the user inserts a smart card, the system also displays any other login servers that are stored on the card from previous successful logins. The user selects the server to use. The system searches for all available choices. The server that the user selects determines the type of authentication.

Setting up network computers that do not require user IDs

You can set up network computers that let anyone access the network and use JavaOS for Business. For example, you might want to set up a group of machines that customers coming into your building can use to learn more about your company and products. The network computers used for this purpose cannot have a local authentication device, such as a smart card reader.

To set up one or more network computers that anyone can use, define a guest user ID and password using the JavaOS Configuration Tool. You can define the user ID for a machine, profile, or platform. Select Login Settings from the **Available** list of services. On the Login Settings panel, select the check box for Automatically login machine as guest user and specify the guest user ID and password. Figure 29 shows the panel with the check box selected.

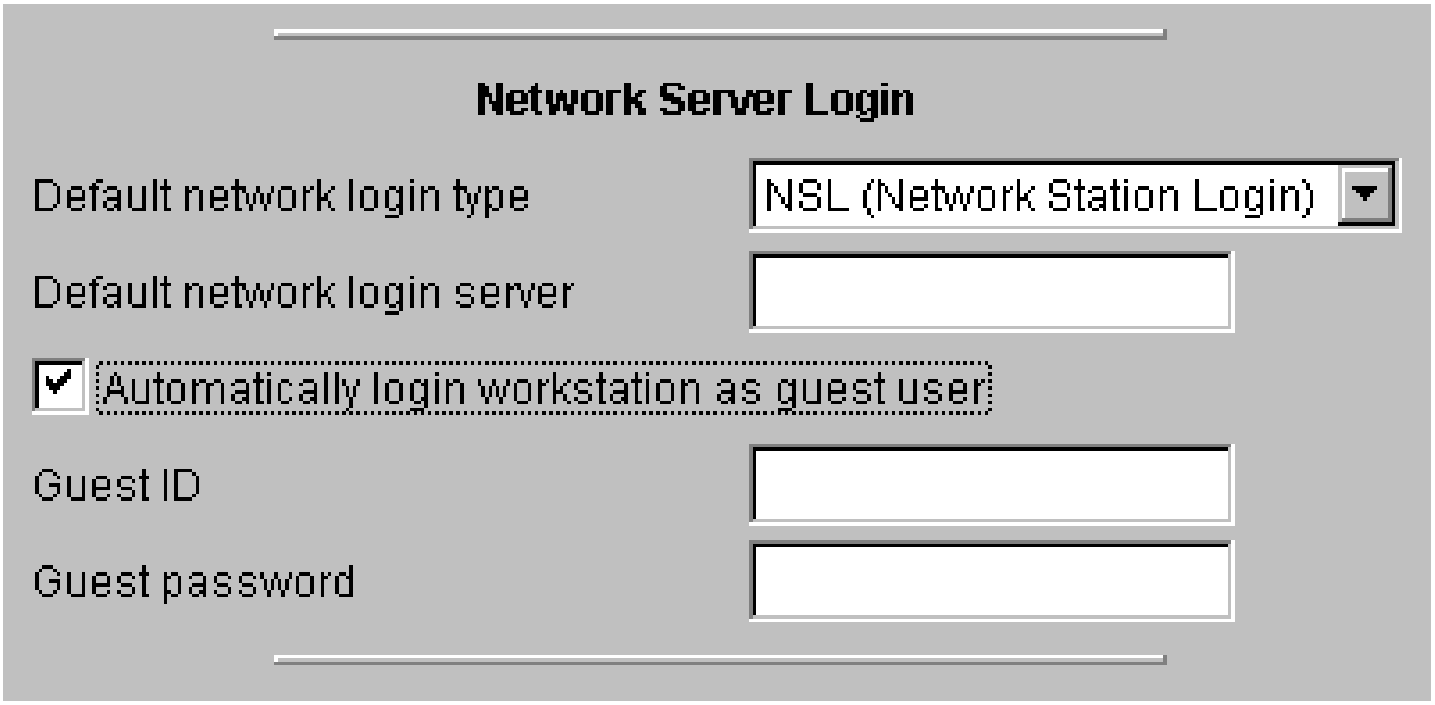

Figure 29. Login settings panel showing the definition of a guest user ID

When you define the guest user ID, the ID automatically receives the characteristics of the JavaOS System Database and the characteristics that you set up in the user accounts on Windows NT.

You can also define specific resources that the guest user ID can access. After you define the ID, use the JavaOS Configuration Tool to edit the user ID. Select the resources you want the user to be able to use. For example, you might have a local printer available to a machine that lets your customers print information about your company and products. You can define the printer for the guest ID.

Setting the user inactivity time and logout warning time

The system monitors the keyboard and mouse activity at the network computer. After a specific amount of time of inactivity, the user is warned that the network computer will be logged out in a set amount of time if there is no activity. You can set the amount of inactivity time before the warning is displayed. You can also set the logout warning time, which is the amount of time between when the warning is displayed and the network computer is logged out.

From the JavaOS Configuration Tool, select Login Settings. You can select Login Settings for a user, group, profile, platform, or machine. At the bottom of the panel in the Workstation Inactivity Timeout section, mark the check box to enable timeout. Specify the amount of time, in minutes, that the network computer can remain inactive before the system prompts the user.

To disable inactivity timeout, specify -1.

Specify the amount of time, in minutes, between the time the logout warning is displayed and the network computer is logged out. The default is one minute.

Chapter 7. Replacing the JavaOS for Business artwork

The system uses artwork that you can replace with your own artwork. The artwork is used in:

- The first panel that is displayed after a network computer is powered on, but before the login panel is displayed, contains artwork stored in:

  `src/javaos/lib/logonfw/logo.jpg`

- The JavaOS Configuration Tool panels also contain stored in:

  `src/server/com/ibm/jossrv/jct/html/logo.JPG`

  Note that the file extension of the file name, JPG, is capitalized.

To use different artwork, store the artwork in a file with the same name as the one you want to replace. You must rebuild the system in order for the system to use the new artwork. Information about rebuilding the system is in *JavaOS for Business Porting Guide*.

Chapter 8. Changing and configuring the HotJava browser

JavaOS for Business includes two pre-customized versions of the HotJava browser, one for use at the server and one for network computers. This chapter describes changing and configuring the two versions of the HotJava browser. You can make the same or different changes to the server and network computerversions of the browser with the exceptions noted below.

Using the HotJava browser at a non-US English server

The HotJava browser for the server is Version 1.1.2, available only in US English. If your server is configured for French, German, Italian, or Spanish, you should be aware of the following considerations and procedures when using the JavaOS Configuration Tool.

Keyboard considerations

Although your keyboard, for other operations, can be configured for your language, you must use the US keyboard when using the JavaOS Configuration Tool. To type characters in your language, press Alt + Shift to toggle between the US keyboard and your language's keyboard.

To configure the US English keyboard on a server already configured for French, German, Italian, or Spanish, do the following before using the JCT:

- Click Start
- Click Settings
- Click Control Panel
- Open the keyboard icon
- Click the Input Locales tab
- Click Add
- Select English (United States)
- Click the Left Alt + Shift button
- Install the keyboard driver with the Windows NT 4.0 Server disk
- Click OK

These steps install the keyboard.

Using the JavaOS Configuration Tool

While using the JCT, press Alt + Shift to toggle between the US keyboard and the configured language. You can enter your languages characters, you cannot enter characters with Alt + Gr.

If your JCT is configured for French, German, Italian, or Spanish, do the following steps to add a service to a user, group, machine, platform, or profile:

- Select the service to add.
- Click the ← to add an available service.

- Press Alt + Shift to toggle to US English.
- Configure the service.

Do these steps again for each added service. You must also toggle to US English each time before you click on an already added service to change its configuration.

Using the HotJava browser to run the JCT at the server

Use only the provided scripts to start the HotJava browser. Issue **startjsd**, then issue **startjct**. When you issue startjct, it reads from a preference file provided with JavaOS for Business to start the browser. That preference file is unique to an instantiation of the HotJava browser used to run the JCT. The preferences you may have set for other instances of the browser are not picked up when you use the provided scripts and may not result in a succesful instance of the JCT.

This provided set of preferences does *not* have SOCKS configured. Running the JCT with SOCKS configured can produce unpredictable results.

However, there is only one executable of the browser. You can continue to use the HotJava browser as a normal browser.

Environment variable considerations

Some applications at the server might use the environment variable HOME=pathname for that application. Or, HOME=pathname might be set for the server as a whole. However you do set the HOME environment variable, do not assign a value for the `\javaos\jossrv` directory where you installed JavaOS for Business.

Customizing help options

If you use the HotJava browser as the main application at the network computers, you must customize the Help pull-down options. From the HotJava browser Help pull-down menu, a network computer user can select several options that, by default, contain a dummy address. These options are:

- Comments
- Submit a Bug

You can change the dummy address to reflect the address you want your customers to use for sending comments and for reporting problems. You can also decide that you do not want the Comments option, the Submit a Bug option, or both available from the Help pull-down menu.

The following topics describe how to change the addresses for the options and how to delete the options from the Help menu.

Changing the address for the Comments option

To change the address in the Comments option:

1. Edit the file `comments.html`. The file is in the hotjava.zip file in the lib\hotjava directory. Unzip the hotjava.zip file to make your changes to the comments.html file.You can obtain an unzip utility as freeware from Info-ZIP at http://www.cdrom.com/pub/infozip/Info-Zip.html.
2. In the comments.html file, locate the following line:

```
<param name="to" value="hotjava-comments@your.site.here">
```

3. Change the line and insert your own address.
4. Rezip the hotjava.zip file.

You might also want to change the wording for the Comments option. If so, you can make any wording changes you want in the HTML file.

Changing the address for the Submit a Bug option

To change the address in the Submit a Bug option:

1. Edit the file `mailbug.html`. The file is in the hotjava.zip file in the lib\hotjava directory. Unzip the hotjava.zip file to make your changes to the mailbug.html file.
2. In the mailbug.html file, locate the following line:

   ```
   <param name="to" value="hotjava-bugs@your.site.here">
   ```

3. Change the line and insert your own address.
4. Rezip the hotjava.zip file.

You might also want to change the wording for the Submit a Bug option. If so, you can make any wording changes you want in the HTML file.

Deleting the Comments or Submit a Bug option

To delete the Comments option, Submit a Bug option, or both options from the HotJava browser Help pull-down menu:

1. Edit the file `hjResourceBundle.properties`. The file is in the hotjava.zip file in the directory classes. Unzip the hotjava.zip file to make your changes to the properties file.
2. In the properties file, search for the string:

   ```
   helpmenu=HELP\
   ```

3. Depending on the option you want to remove from the Help, delete one or both of the following lines:

   ```
   go doc:\lib\hotjava\Comments.html=Comments... \
   go doc:\lib\hotjava\MailBug.html=Submit a Bug... \
   ```

4. Rezip the hotjava.zip file and load it.

Configuring the HotJava browser for network computers

On the JavaOS Configuration Tool edit panel, add “HotJava Configuration” on page 104 and “HotJava Configuration” on page 104 to users or machines to configure the HotJava browser.

Chapter 9. Implementing a default main application

The main application is the default application that the system loads and runs after a user logs in. By default, JavaOS for Business provides the HotJava browser as a *startup application* or main application. You can use a different Java application, for example, one you write yourself or one you purchase.

Regardless of the application you use, including the HotJava browser, you must specify it using the JavaOS Configuration Tool.

The following topics describe startup application considerations.

Using the HotJava browser as the main application

To use the HotJava browser as the main application, use the JavaOS Configuration Tool to specify that application for a user, group, or profile. For example, suppose you are defining the HotJava browser as main application for a group of users. Using the JavaOS Configuration Tool, select the group you want to edit. Then choose the HotJava load settings and add them for the group. The HotJava browser is now the application the system loads when anyone from this group logs in. Then add the HotJava browser configuration settings.

Chapter 8, “Changing and configuring the HotJava browser” on page 67 describes changing help options.

Note: JavaOS for Business supplies two versions of the HotJava browser, one for the system administrator at the server, and a modified version designed to be configurable from the JavaOS Configuration Tool. You should configure users, with the “HotJava Load Settings” on page 104, to run the modified version so that you can configure the application for users.

Specifying a different startup application

To load a different application as the startup application on JavaOS for Business, the application must be written in Java. The following are also required:

- The application must be developed using the Java Development Kit (JDK) 1.1.4. Only 100% pure Java applications must run on 1.1.4.
- The application class files must be archived into a zip file.

You can also include the JavaOS for Business extensions in a Java application you want to use as the main application.

Use the JCT edit panels to add and configure the Application Load Settings available resource. See “Application Load Settings” on page 103 for more information.

Specify the settings for the application:

- The Service URL. This is the NFS mount point.
- The name of the main class of the Java application.
- URL location of the zip file

  Applications are bundled as zip archives. The system refers to them using URLs. You can store your applications anywhere that a URL can be stored; on the network or locally at the network computer in flash

ROM (read-only memory). Because the archive file is a URL, the system can easily download from many places.

An application consists of class files and all the other files for the application, such as GIFs, JPGs, TXT, HTML, properties, and so on. To load your applications in this way, you must archive the class files under the \classes directory and archive the other files for the application under the \lib directory. This lets the system know where your classes are and adds them to the system CLASSPATH.

- Application URL base path; code base.
- The application home property name. For example, for your application, this might be acctdesk.home. In the application startup properties text area, which is described below, you must identify where the home property is located.
- Key/value pairs separated by the equal sign (=). You can use the text area to specify arguments for the application and the location of the home property. For example, the location of the home property might be:

  ```
  acctdesk.home = file:/HOME/deskapps
  ```

  Enter each argument on a separate line.

Guidelines for using application load settings

This method of loading is basically for the quick loading of applications for testing and for trial applications. To use JavaOS for Business dynamic class loading and configure an application using the JavaOS Configuration Tool, you must use the JavaOS for Business Software Development Kit. The kit contains examples of code for loading and configuring an application.

How applications are loaded

This method of application loading has some restrictions. The class files are not dynamically loaded from the server. All files that must be found in the CLASSPATH must be archived into the zip file under /classes. This includes any resource bundles that are found in CLASSPATH.

Chapter 10. Defining and specifying device drivers

You can either write your own device drivers, purchase them, or use the supplied device drivers.

Adding a new device driver

See *JavaOS for Business Device Driver Guide* for information about writing a device driver. After you create or purchase the device driver, you must define it to JavaOS for Business in order to access it and use it through the JavaOS Configuration Tool.

The files for a device driver are:

- JAR file for the device driver.
- JAR file for the JCT configuration bean, known as the configuration JAR file.

Install the configuration JAR file in the /jars subdirectory of the JavaOS Configuration Tool. You can configure the subdirectory using the JCT maintenance panel. After you install the file, the device driver is available from the JavaOS Configuration Tool in the Available selections list.

Install the JAR file for the device driver on a network server. Make a note of the URL where you installed the file. After you install the file, use the JavaOS Configuration Tool to configure the platform, profile, machine, group, or user and add the device driver as one of the services. When you specify the device driver in a JCT edit panel, the configuration panel for the device driver is displayed. As an example, “Parallel Port Driver” on page 109 describes one of the supplied drivers.

The first heading is Service Installation Information. Specify the service URL where you installed the JAR file for the device driver. In this context, the JCT considers a device driver to be a service that you make available on the system. The URL you specify depends on whether the device is a boot time device or a non-boot time device.

Boot time device drivers

A boot time device is one that is made available to the network computer when the system is booted for the machine. Basically, boot time services are services that the network computer and JavaOS for Business require for booting.

A non-boot time device is one that becomes available after the network computer is up and running JavaOS for Business. Non-boot time services are services that are not required during the boot process.

The boot time services JavaOS for Business provides are listed in “Resources required at bootup” on page 100.

If the device driver you are defining is required at bootup, on the JCT panel, File:/ is displayed for the service URL. Specify the URL relative to where the JavaOS Configuration Tool is running because the master configuration tool must build the master configuration file from the server and not on the network computer. The master configuration file contains the necessary information for the network computer to boot the system. For more information about the master configuration tool and file, see “Platform and machine resources and settings” on page 100.

Drivers not required at boot time

If the device driver you are defining is not required at bootup, on the JCT panel, the value FILE:/SERVICES is displayed for the service URL. Specify the URL that the network computer will use to locate the JAR file for the device driver. You can use the default FILE:/SERVICES URL. If you use this URL, you must configure /SERVICES as an NFS mount point. For more information, see Chapter 13, “Mounting an NFS file system” on page 79.

Supplied device drivers

JavaOS for Business includes device drivers for typical devices. The supplied device drivers include:

- Parallel port driver, described in “Parallel Port Driver” on page 109
- Serial port driver, described in “Serial Port Driver” on page 111
- PCI Token Ring Driver, described in “PCI Token Ring Driver” on page 110
- PowerPC Audio Driver, described in “PowerPC Audio Driver” on page 110
- S3 Trio 64 Driver, described in “S3 Trio 64 Driver” on page 111
- Sound blaster 16 audio driver, described in “Sound Blaster 16 audio driver” on page 112
- SVGA driver, described in “SVGA driver” on page 113
- Video driver, described in “Video driver” on page 114
- Intel® EtherExpress™ Pro 100, described in “Intel EtherExpress Pro 100” on page 105
- Keyboard driver, described in “Keyboard driver setting” on page 106
- Mouse driver, described in “Mouse driver setting” on page 107
- NC PCI token ring driver, described in “NC PCI token ring driver” on page 107
- NS1000 Smart Card Driver, described in “NS1000 Smart Card Driver” on page 108

Chapter 11. Defining the keyboard, language, and time zone settings

You can set specify settings for keyboard language mapping, language, and timezone. You can these settings for a certain platform and then override them for a particular group or user.

This chapter helps you plan to use the JavaOS Configuration Tool to set up these definitions.

Selecting the keyboard settings

You can specify a keyboard mapping for the network computer. You can configure the keyboard for one or more platforms and then change the keyboard for a group or user.

Use the JavaOS Configuration Tool to define the keyboard settings. Select the Keyboard Format Settings and choose the keyboard to use. Use the up and down arrows to display each available keyboard.

The country codes and numbers for the keyboards follow International Standards Organization (ISO) standards. See “Keyboard format settings” on page 106 for a list of the supported keyboards.

For more information about the keyboards, see *JavaOS for Business Keyboard Reference*.

Selecting the language settings

You can specify the language JavaOS for Business uses. You can specify the language and region. The language setting not only specifies the language JavaOS for Business uses, but also determines other items, such as the punctuation usage and the format of the currency.

You can also select a region to further refine the language definition. The region is the geographic location. For example, you might select `fr` for language and `Canada` for region.

Selecting the language format settings

Use the JavaOS Configuration Tool to define the language settings. When you select Language Format Settings, the panel displays the lowercase two letter codes corresponding to the language. The codes are defined by ISO standards.

The language codes and languages that JavaOS for Business supports are listed in “Locale format settings” on page 106.

Selecting the region settings

Use the JavaOS Configuration Tool to define the region settings, also known as the country code. Each region corresponds to a specific language. If you select a region that does not match the language setting you defined, the region is ignored. The region codes that JavaOS for Business supports and the languages and regions the codes correspond to are listed in “Locale format settings” on page 106.

Selecting the time zone

You can set the time zone for network computers. You can define the time zone for a platform, profile, machine, group, or user.

Use the JavaOS Configuration Tool to define the time zone. When you select Time Format Settings, the panel displays the uppercase three letter abbreviation for each time zone.

The abbreviations and time zones are listed in “Time zone settings” on page 113. Be sure to consider whether a location observes daylight saving time.

Chapter 12. Managing fonts

You can add and delete fonts and manage them on the server.

Font files

You must have a \FONTS subdirectory on the server. All of the information about the system's fonts are stored in the \FONTS subdirectory. You must also have a \lib subdirectory in the \FONTS subdirectory. The font files are stored in the \lib subdirectory.

There are two files associated with fonts; FONT.LST and font.properties.

The FONT.LST file

The FONT.LST file contains a list of the system's fonts and maps the font names to their file names. There is one entry per line. The format of each entry is:

```
FontName style type FileName
```

- `FontName` specifies the name of the font.
- `style` specifies the font's style, such as PLAIN, BOLD, ITALIC, or BOLDITALIC.
- `type` specifies the font's type, which is usually TrueType.
- `FileName` specifies the name of the file.

A sample entry is:

```
helv  BOLD    truetype  helvbd.ttf
```

The font.properties file

The font.properties file contains the font that will be used at your system. The file must be in the \FONTS\lib subdirectory.

Font files included in JavaOS for Business

The supplied font file, `tmrmt30.ttf`, is Monotype Corporation's Times New Roman MT30 Unicode font. Fonts for the following double-byte character set languages are also provided:

- Korea
- Japan
- People's Republic of China
- Taiwan

The files for these fonts are identified by the local suffix that is added to the font.properties file name, for example:

Korea `font.properties.ko`

Japan `font.properties.ja`

People's Republic of China `font.properties.zh`

Taiwan `font.properties.zh_TW`

You must either copy the font to the font.properties file or rename the file to font.properties.

Creating or adding a font

You can create fonts, add fonts to the system, and delete fonts from the system.

To create a font, use a font editor. For example, you can use the Windows 95 font editor, which lets you define TrueType fonts.

To add a new font, add an entry for the font to the list of available fonts in the FONTS.LST file. You must also add a file in the format

`font.properties.xx`

where `xx` differentiates your new font file from the other files. The file name for the font must be font.properties and the file must be in the \FONTS\lib subdirectory.

Deleting a font

To delete a font, you can simply remove its entry from the FONTS.LST file. However, you should also consider removing the font file itself from the system because of the amount of space it uses.

Chapter 13. Mounting an NFS file system

Use the Network File System AutoMounter to mount NFS file systems. Select NFS AutoMounter on the JavaOS Configuration Tool edit panel to configure the NFS AutoMounter and to specify the file system to mount. You also need to add and configure the Network File System (NFS) resource. See “NFS AutoMounter” on page 108 and “Network File System (NFS)” on page 108 for configuration information.

Note: When configuring NFS, do not select TCP option for a Windows NT server.

The server might be a code server for the device drivers or other services.

The network computer usually cannot access the Java classes using the mounted file systems.

Chapter 14. Specifying keyboard and mouse drivers

Most network computers will have an attached keyboard and mouse. You must configure the base keyboard and mouse driver for each network computer equipped with a keyboard or a mouse. Then configure the individual keyboard driver and PS/2 mouse driver.

Keyboard driver

For a machine with a keyboard, configure the keyboard driver. This setting is in addition to the keyboard format setting.

Mouse driver

For a machine with a PS/2 mouse, configure the mouse driver. This setting is in addition to the base keyboard and mouse driver.

If you plan to support a different type of mouse, you must provide the driver. See *JavaOS for Business Device Driver Guide* for information about creating device drivers.

Chapter 15. Specifying video information

You need to configure the video resolution of your network computers. You also must configure a driver for specific video monitors.

Configuring video resolution

Add the video settings to a machine, profile, or platform to set the height, width, and color depth of displays. If you add these settings to a group or user, they are ignored.

Configuring an SVGA monitor

Add the generic SVGA driver to machines, profiles, or platforms with SVGA monitors.

Configuring an S3 Trio 64 Monitor

Add the S3 Trio 64 Driver driver to machines, profiles, or platforms with S3 Trio 64 V+ or S3 Trio 64 V2/DX monitors.

Add the generic SVGA driver for other monitors.

Chapter 16. Specifying multimedia information

If your network computer hardware includes sound cards, add the appropriate driver resource to the machine, profile, or platform.

The available sound card drivers are:

- "PowerPC Audio Driver" on page 110
- "Sound Blaster 16 audio driver" on page 112

Chapter 17. Specifying printing information

Assigning the available configuration resources for printing involves selecting both drivers and print configuration resources.

Local printing

Local printing is printing at a printer attached directly to a network computer. You can configure more than one network computer to send print jobs to a specific local printer.

Parallel printer

To support a parallel printer you must add the parallel port driver, “Parallel Port Driver” on page 109, and “Parallel Printer” on page 109 to configure the parallel printer.

If you want other users to be able to send print jobs to this printer, click Share in the edit panel. Then enter the TCP/IP hostnames of each printer that is allowed to send print jobs to this printer.

You can also specify paper size, printer language, and paper tray.

Serial printer

To support a serial printer you must add the serial port driver, “Serial Port Driver” on page 111, and configure the parallel printer, “Serial Printer” on page 111.

If you want other users to be able to send print jobs to this printer, click Share in the edit panel. Then enter the TCP/IP hostnames of each network computer that is allowed to send print jobs to this printer.

You can also specify paper size, printer language, and paper tray.

Remote printing

Add the Remote Printer resource, described in “Remote Printer” on page 111 to specify that a user can make remote print requests.

You specify the name of the remote printer and whether it is a Network Information Service (NIS) printer. You can change the name of the printer, but if it is an NIS printer, the name must match the name in the NIS printer map.

Specify whether this printer's name should appear in the list of available printers when the user makes a print request.

Network Information System printers

The names of NIS printers are listed in a printer map. The default map name is `printers.conf.byname`. You can accept or change this name. The TCP/IP host name of any NIS printer you assign should be in this map. “NIS Printers” on page 108 describes configuring NIS.

Chapter 18. Specifying miscellaneous information

This section lists some available services you might want to add or change at your system. The required services are so noted.

Proxies

Add the proxy and SOCKS information to define how a system accesses external services such as the Internet or other systems. See “Proxies” on page 110.

System database location

You can specify where the JavaOS System Database is stored. You have to configure this boot-time information for each machine. See “System Database Location” on page 113 for details.

System image location

You must specify the location of the operating system image for each machine. You can accept or change the default location. See “System Image Location” on page 113 for more details.

SNMP server location

Add this setting to define the SNMP manager address for machines. See “SNMP Manager Address” on page 112.

TFTP location

Add this setting to specify a mount point for TFTP. See “TFTP AutoMounter setting” on page 113.

Parallel and serial port drivers

If your network computers will have parallel or serial devices, such as printers or modems installed, you must configure drivers for those devices. See “Parallel Port Driver” on page 109 and “Serial Port Driver” on page 111 for descriptions of the drivers.

Just in time (JIT) compiler

Your system might include a Just In Time (JIT) compiler. See “Just In Time (JIT) compiler setting” on page 105.

Token ring and Ethernet drivers

Add the driver for the connection method and hardware type at your system. You can add the settings for:

- Intel EtherExpress Pro 100. See “Intel EtherExpress Pro 100” on page 105.
- NC PCI token ring driver. See “NC PCI token ring driver” on page 107.
- PCI token ring driver. See “PCI Token Ring Driver” on page 110.

Chapter 19. Specifying troubleshooting services

JavaOS for Business includes several troubleshooting resources that you can add. Adding the resource for each is described in Part 3, " JavaOS for Business system administration" on page 93 and using the tools is in Part 5, "Troubleshooting JavaOS for Business" on page 137.

The tools are:

- "Conditional debug" on page 104
- "Error logger" on page 104
- "Trace tool settings" on page 114

These tools can help you identify and solve system problems.

Part 3. JavaOS for Business system administration

This part of the information tells how to use the applications and utilities provided with JavaOS™ for Business™.

As a system administrator you set or change system values and manage system resources for all users and network computers. The network computer users have access only to those applications, resources such as printers, and browsers that are defined for them by a system administrator. This section tells how you can define users, their access to system resources and applications, and how you can manage users and devices.

JavaOS for Business provides several methods for you to manage system resources:

- The JavaOS Configuration Tool, described in Chapter 21, "JavaOS Configuration Tool edit panels" on page 101, is the graphical interface tool for most user and resource-related tasks.
- The command line described in Chapter 24, "Using the Java command line" on page 121, lets you enter specific JavaOS for Business commands.
- The netShell described in Chapter 25, "Using the netShell commands" on page 123, lets you enter other JavaOS™ for Business™ commands from a network computer.

This section begins with a description of the JavaOS Configuration Tool, a graphical user interface for system administration. If you are setting up a new installation, start with Part 2, "Configuration Planning" on page 55, an overview of JavaOS for Business customization and configuration.

Chapter 20. Getting started with the JavaOS Configuration Tool

When you get started with a new installation, you need to prepare for users, the applications they will run, the resources they will use, and the way you will manage their security. You also need to prepare for the hardware, such as the network computers and printers at your installation. This section describes the main panel of the JavaOS Configuration Tool and gives some suggestions for getting started at a new installation.

Starting the JavaOS Configuration Tool

The JavaOS Configuration Tool is a HotJava browser application. First start the HotJava browser, and then specify the JavaOS Configuration Tool URL. The main menu is displayed in the browser window.

You can start the JavaOS Configuration Tool from any browser that has full Java 1.1 capabilities.

JavaOS Configuration Tool main panel selections and terminology

The JavaOS Configuration Tool uses specific terms. This section describes the terms used in the main panel and the first level of selections you can make.

From the top of the JavaOS Configuration Tool index on the left of the main panel, the selections are:

Users A user is one individual, who might use any network computer or system resource, or who might have any setting defined.

Click User to see an expanded list of actions you can take concerning users.

Add Use this option to define a new user.

Each user ID must also be defined at the Microsoft Windows NT Server 4.0. The Windows NT user ID must be an exact match with the JavaOS for Business user ID. If you want a user to have a password, assign the password when you define the Windows NT user ID.

Remove Use this option to remove an existing user.

Edit Use this option to assign resources or to specify settings for a user.

Groups A group is one or more users who have the same resources and settings. A group might be users who access the same application, need access to a specific printer, who have the same equipment and thus need the same machine settings, and so on. Figure 30 on page 96 shows the relationship of groups and users.

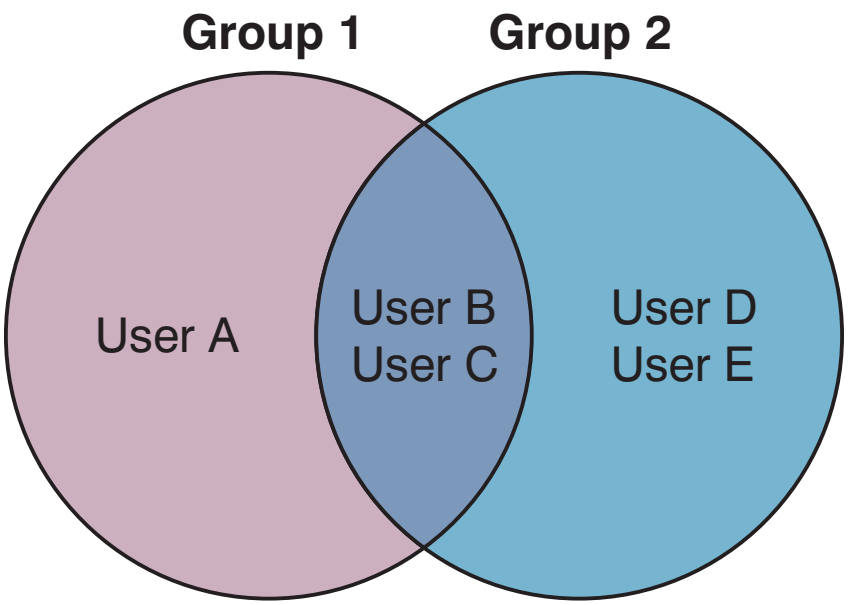

Figure 30. Two groups, each with several users.

Click Group to add a group, to specify the application started when a user in the group logs in, and to specify other system resources and settings for that group of users.

Add Use this option to define a new group at your installation.

Remove Use this option to remove an existing group.

Edit Use this option to assign resources or to specify settings for a group.

Machines A machine is one specific network computer.

Select Machine to add a machine, to specify the the login rules for that machine, and to specify other system resources and settings for that machine. See, also, platforms on page 97 and Figure 31 on page 97.

Add Use this option to define a new network computer. See "Platform and machine resources and settings" on page 100 for the sequence of adding network computers.

Remove Use this option to remove an existing network computer.

Edit Use this option to change settings for a network computer. After changing settings, follow the steps in "Platform and machine resources and settings" on page 100. You must do this operation each time you change machine settings.

Profiles A profile consists of one or more machines that have some arbitrary similarity. A profile might be machines in the same location, machines typically used by one group, or machines grouped by any other common factor. A profile may have no major hardware implications; use **Platform** to specify machines with common physical properties, such as the same manufacturer or the same architecture. Figure 31 on page 97 and Figure 32 on page 98 show the relationships between machine, profiles, and platforms.

A profile can overlap a platform; profiles are another way of grouping machines. A machine can be defined as a member of a platform, even if no profiles are defined. A machine may belong to more than one profile.

A machine inherits settings and resources first from its platform, and then from any profiles it belongs to.

Add Use this option to define a new machine profile at your installation.

Remove Use this option to remove an existing machine profile.

Edit Use this option to assign resources or to specify settings for a machine profile.

Platforms A platform consists of one or more specific network computers, usually from the same manufacturer or with the same architecture.

A platform can overlap a profile; platforms are different ways of grouping machines. Use platform to group machines by physical type; use profile to group machines by who uses them, location, applications used, or other non-physical criteria. A machine can be a member of a platform, even when no profiles are defined. Figure 31 and Figure 32 on page 98 show the relationships between machine, profiles, and platforms. Although a machine might be in several profiles, it should be in only one platform.

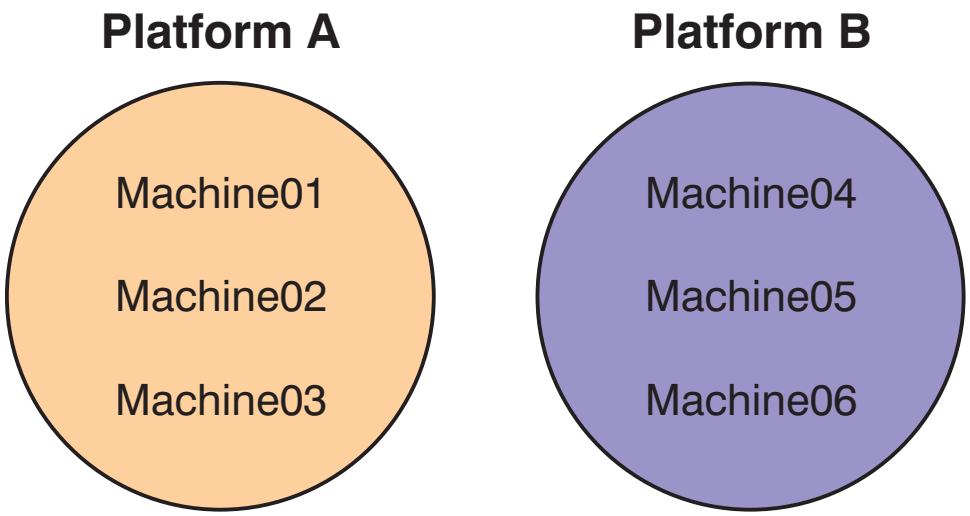

Figure 31. Machines organized into platforms.

A machine inherits settings and resources first from its platform, and then from any profiles it belongs to.

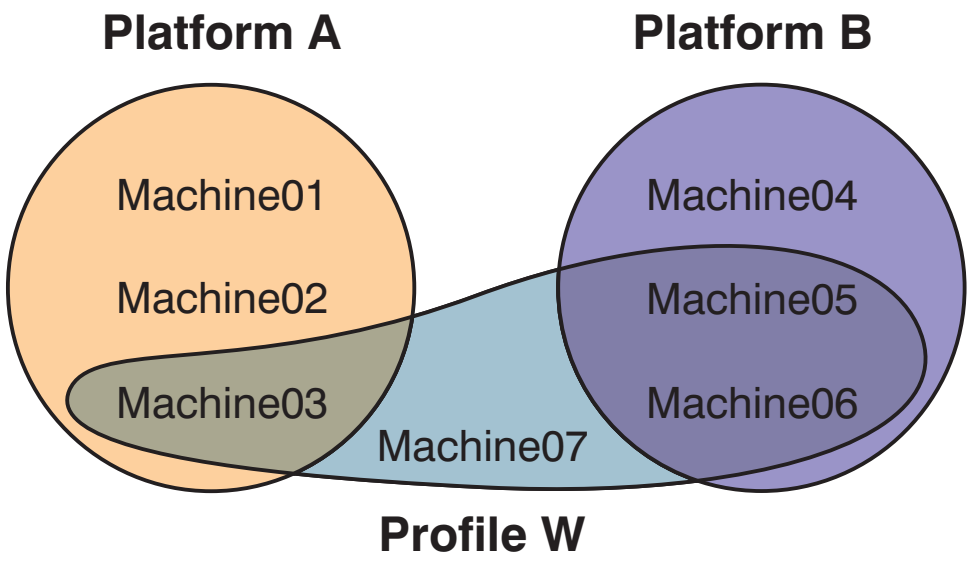

Figure 32. Machines organized in overlapping platforms and profiles

Add Use this option to define a new network computer. See “Platform and machine resources and settings” on page 100 for the sequence of adding a network computer.

Remove Use this option to remove an existing network computer.

Edit Use this option to change settings for a network computer. After changing settings, follow the steps in “Platform and machine resources and settings” on page 100. You must do this operation each time you change machine settings.

System Maintenance System maintenance is a collection of utilities that affect the installation as a whole. See Chapter 23, “System Maintenance” on page 117 for the system maintenance utilities.

Planning an initial sequence of assigning resources and settings

These suggestions are for system adminstrators at a new installation. Later, when the installation is defined and users are already working efficiently, you probably will make more individual changes.

On the whole, start with the larger units of resources and work down to individual users or machines. Some resources and settings must be defined in a specific order.

Groups and users

To add a large group of users with similar characteristics, start by creating a group. You can give the group a name and a brief description.

After you have created a group and assigned it resources and settings, you can populate it by editing individual users to add them to the group and then making any individual edit users changes required.

Each user ID must also be defined at the NT server. The NT user ID must be an exact match with the JavaOS for Business user ID.

A user can be a member of multiple groups. Groups can be users with similar assignments; they may all start up with the same application or need the same printer. A group may be users who need multimedia settings configured for them.

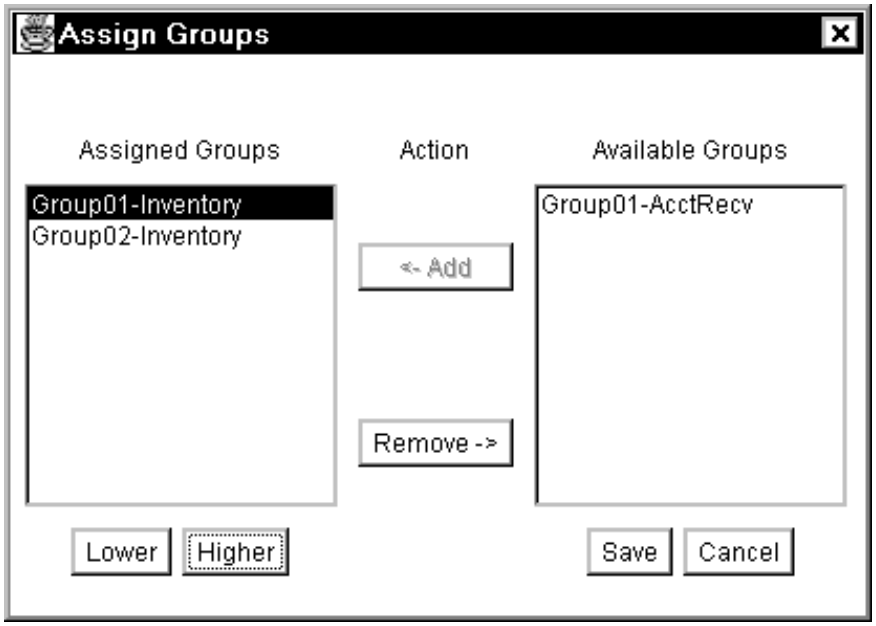

Figure 33. The Assign Groups panel of the JavaOS Configuration Tool

If you add user to multiple groups, you can specify the priority of resources the user can access by ranking the priority of the groups:

1. On the JavaOS Configuration Tool main panel, click Users.
2. Click Edit.
3. Select a user from the drop-down list.
4. Click Assign groups
5. Click the first group in the Available Groups column to add, click the left arrow to add the group to the Assigned Groups column.
6. Click the next group to add, click the left arrow to add the group.
7. Continue to assign groups to this user until you have completed your selections.
8. Select the group whose priority you want to raise.
9. Click the Higher button at the bottom of the list. That group's resources and settings take priority over groups lower on the list.
10. Continue to use the Higher and Lower buttons until you have the groups arranged in your preferred order.
11. Close the Assign Groups window.

Platform and machine resources and settings

This section describes an orderly sequence of defining and configuring individual network computers

You must add and configure a machine before a user can log in and use that machine. The configuration information about a machine is stored in the Master Configuration File (MCF). This file holds the information required to boot a machine such as the location of the system image, the hardware communications type, and other bootup information.

The following sequence describes configuring one machine. Follow the same sequence to configure a platform of machines. The sequence is:

1. Add a machine. Use the JavaOS Configuration Tool to add a machine. Enter the machine's hardware type, its MAC address, the serial number and, optionally, the machine location.
2. Edit the machine configuration. You can assign the machine to a profile. Editing the machine configuration lets you add available resources and settings for that machine.

   You must select System Image Location for the machine and add it to the machine's list of resources. In the System Image Location panel, enter the pathname for the local path at the server.
3. After you have added all the resources and set all the settings you want for this machine, click Save to make the changes.
4. Click System Maintenance on the main JavaOS Configuration Tool menu. Then click Export MCF.
5. Select the machine you just configured from the list of configured machines and platforms. Click Build to create the file information for this machine.

   JavaOS for Business uses the machine configuration information to build a basic file to use when the machine is booted up.

Error occurences are recorded in `mcf.log`.

Resources required at bootup

You must use the System Maintenance and Export MCF procedure each time you change or add one of the following resources or settings of a machine or platform so that the correct information is sent to the machine at bootup:

- System Database, described in “System Database Location” on page 113
- Intel EtherExpress Pro 100, described in “Intel EtherExpress Pro 100” on page 105
- Network File System (NFS), described in “Network File System (NFS)” on page 108
- PCI Token Ring Driver, described in “PCI Token Ring Driver” on page 110
- NC PCI Token Ring Driver, described in “NC PCI token ring driver” on page 107
- Keyboard and Mouse Driver, described in “Keyboard and Mouse Base Settings” on page 105
- Just in Time compiler (JIT), described in “Just In Time (JIT) compiler setting” on page 105
- Conditional Debug, described in “Conditional debug” on page 104
- Error Logger, described in “Error logger” on page 104
- Trace tool settings, described in “Trace tool settings” on page 114

Chapter 21. JavaOS Configuration Tool edit panels

Use the edit panels to configure resources and to set or change settings for system users or network computers (called machines on the panels).

The information here describes the services, the resources and settings displayed on the edit panels when you select User, Groups, Machines, Profiles, or Platforms and then Edit. The online help and flyover balloons help you do the actual tasks.

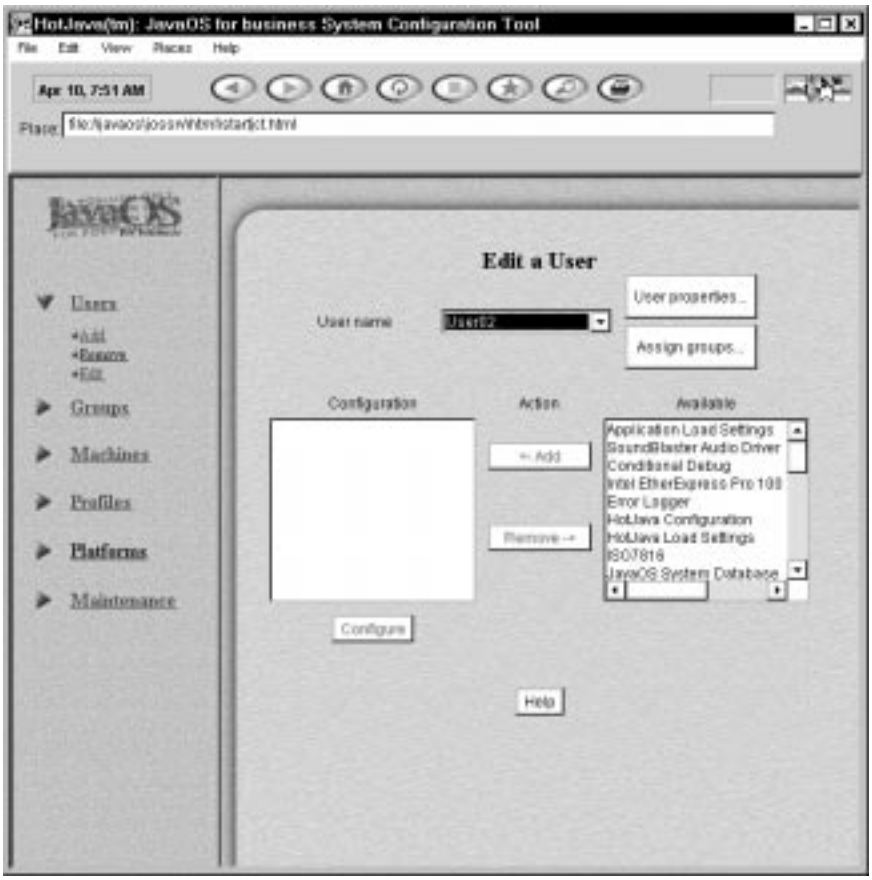

Figure 34. JavaOS Configuration Tool edit panel

The edit panel displayed is about the same, whether you are editing a user or a machine or a group of users or machines. The panel lets you change the properties of a user; the user's name or telephone number. You can also assign a user to a group, a machine to a profile, or to a platform.

Using the edit panels

Use the edit panels to add, delete, or change resources and settings available to users and machines. The edit panels list all the available resources and settings, whether they are appropriate to the user or machine you are defining. If you add and configure settings that cannot be used, they are ignored.

The edit panels are described here in alphabetical order although the order may differ on the actual panels.

Modifying user information

The Properties button next to the **user name** lets you view or modify the user properties such as the full name or phone number.

The Assign Groups button lets you assign that user to an existing group. Select the group from the dropdown menu.

Changing machine information

The Properties button next to the machine name lets you view or modify the machine's properties such as MAC address.

You can also assign a machine to an existing profile or platform.

Adding a setting or resource

The main part of the edit panel lets you add or remove settings from a user, group, machine, profile, or platform. In each case, to add a setting or resource, you select it from the list on the right and click on the arrow (←) to add it to the list on the left. If the setting needs configuration, a popup panel lets you specify settings. Some settings may, in turn, have several related panels. Make the changes you require in these auxiliary panels to complete the setting.

Some additions require you to add related resources. Adding a language might require you to add the Region settings. Changes to machine components might require an update to System Maintenance. See “Platform and machine resources and settings” on page 100 for the steps to take. Make the changes you require in the related settings to complete your operation.

Removing a setting or resource

To remove a setting or resource, click on it to select it from the Configuration list on the left, and click on the right arrow (→) to remove it from the list of assigned services.

Some changes require related changes. Changes to the language format might require changes to the Region settings. Changes to machine components require an update to System Maintenance. See “Platform and machine resources and settings” on page 100 for the steps to take. Make the changes you require in the related settings to complete your operation.

Changing a setting or resource

To change a setting or resource:

1. Click on the setting to select it from the Configuration list on the left.
2. Click on the Configure button at the bottom of the list.
3. Make your changes in the popup panels for that setting.

Some changes require related changes. Changes to the language format might require changes to the Region settings. Changes to machine components require an update to System Maintenance. See “Platform and machine resources and settings” on page 100 for the steps to take. Make the changes you require in the related settings to complete your operation.

Overriding a setting or resource

Because the settings assigned may be mutually exclusive, you can specify the order in which one setting overrides a conflicting setting. For example, the video setting for a machine may be set to *value_A*. The video setting for one user is *value_B*, a conflict with the machine setting. "Groups and users" on page 98 tells how to set priority for a users assigned to multiple groups.

Descriptions of the available settings and resources

The following sections describe, in alphabetical order, all the settings and resources in the **Available** list. Any setting or resource noted as valid for a user is also valid for a group of users. Settings valid for a machine are also valid for a profile or platform of machines.

Assign only those settings appropriate to your hardware. For example, add sound card settings only to machines with sound cards.

The order of available settings and resources might differ on your edit panels. You can always look at the configuration panels for a setting or resource before you decide to add it to a user or machine. Click **Cancel** to close a panel without making any changes.

Application Load Settings

Select Application Load Settings to specify the start-up application. This setting can be specified for a user, group, machine, profile, or platform. For example, you might want one group of users to start with a banking application while another group starts, at login, with an accounts receivable application.

Use this resource to load a general application. The HotJava browser has its own resource settings, described in "HotJava Load Settings" on page 104 and Chapter 8, "Changing and configuring the HotJava browser" on page 67.

The start-up application must be a Java application, not, for example, one with C++ in it. The class files for the application which must be accessible through the paths specified in CLASSPATH must have been archived in a zip file.

Using this setting to specify a startup application for a single network computer can be an efficient way to test a new application. If no startup application is specified, the system hangs after a user logs in.

Note: If more than one application meets the requirements to be a start up application, the system loads the first one found meeting the startup application requirements.

For example, to specify the HotJava browser as the startup application, you can fill in the Application Load Settings as follows:

```
Main Class                sunw.hotjava.Main
Location of ZIP file      file:/NFSMOUNT/hotjava.zip
URL base path
Application home property hotjava.home
```

Base Smart Card Services

Add this resource to machines with smart card support.

In the Service URL field, either accept the default provided, `/SERVICE/iso7816.jar`, or specify a different location of the service URL.

This resource is required by the NS1000 smart card driver. See “NS1000 Smart Card Driver” on page 108 for that resource.

Conditional debug

Add this setting to specify conditional debug flags. You can specify these settings for a user or for a machine. See “Conditional debug messages” on page 140 for a description of the conditional debug messages.

This information is required by a machine at bootup. Follow the steps in “Platform and machine resources and settings” on page 100 after you add this setting.

Error logger

Use this setting to configure the error logger. In the popup panel, click State to turn error logging on or off. The Host Name is the server where the error log daemon is running. The default port is 7786. You can also set the error logger to ignore some events. See Chapter 32, “Using the error log” on page 149 for a full description of the error logger.

This information is required by a machine at bootup. Follow the steps in “Platform and machine resources and settings” on page 100 after you add this setting.

HotJava Configuration

Use this setting to configure the HotJava browser. You can specify the start up home page, the location of the button bar, the screen and print typefaces, and other typical browser settings.

You can click to open subpanels with options for the display, mail, applet security, and raw properties.

Use this setting in conjunction with “HotJava Load Settings”

If you do not add and configure these settings, each user would have to set them individually.

When you add and configure this resource, you have configured the HotJava browser, but it is not loaded unless you also select “HotJava Load Settings.”

To change the HotJava browser help options, see Chapter 8, “Changing and configuring the HotJava browser” on page 67.

HotJava Load Settings

Add this setting to a user or group to specify the HotJava browser as the main startup application. The setting tells where the application is located. If you add this, you should also add the HotJava configuration settings described in “HotJava Configuration.” If no startup application is specified, the system hangs after a user logs in.

Add this setting to a user or group although you could add it to a machine, profile, or platform.

iiimp setting

Add this setting to a user or group to specify the location of a server maintaining the double-byte character support (DBCS) for a machine, profile, or platform. This server may be the same or separate from the the boot image server. This support is for the Japanese language environment.

Intel EtherExpress Pro 100

Specify this setting for each machine connected with an Intel EtherExpress Pro 100 communications card. You can accept or change:

Service URL This is the location and name of the driver. The default is `/export/services/EEPro100.jar`.

Matching Name The default is `pci8086,1229`.

Logical Name The default is `Ethernet Driver`.

Service Load Time Click Customize to specify when this service is loaded. Select from:

- When discovered.
- When matched. This is the default.
- Upon connection.

You must follow the instructions in “Platform and machine resources and settings” on page 100 for each machine with this card before booting the machine.

Just In Time (JIT) compiler setting

Add this setting to specify the use of a Just In Time (JIT) compiler. Specify *off* to turn off use of an installed compiler, or specify a threshhold number to specify when the compiler should become active.

This information is required by a machine at bootup. Follow the steps in “Platform and machine resources and settings” on page 100 after you add this setting.

Keyboard and Mouse Base Settings

Add this setting for machines or platforms that will support a keyboard or a mouse. You must also add the individual keyboard and mouse drivers. This setting adds support for any keyboard. “Keyboard format settings” on page 106 defines the specific keyboard.

Notes:

1. Add the Keyboard configuration setting to set the keyboard interrupts if you plan to support a keyboard. See “Keyboard driver setting” on page 106.
2. Add the Mouse configuration setting to set the mouse interrupts if you plan to support a PS/2 mouse. See “Mouse driver setting” on page 107
3. After adding this setting, follow the directions in “Platform and machine resources and settings” on page 100.

Keyboard driver setting

Configure this setting to set the interrupts for a keyboard.

This setting requires that “Keyboard and Mouse Base Settings” also be added. Use “Keyboard format settings” on page 106 to specify a keyboard mapping.

Keyboard format settings

Select Keyboard Format to specify a keyboard mapping. The keyboard formats are:

Country	Keyboard ID	Valid Keyboards
Brazil	275	BR275
Brazil	274	BR274
Czech	243	CZ243
Denmark	159	DK159
Finland	153	FI153
Japan	194	JA194
Greece	319	GR319
Greece	459	GR459
Hungary	208	HU208
Netherlands	143	NL143
Norway	155	NO155
Poland	214	PL214
Poland	444	PL444
Portugal	163	PT163
Russia	443	RU443
Russia	441	RU441
United States	103	US103
Turkey	179	TR179
Turkey	440	TR440

Press UpArrow (↑) and DownArrow (↓) to scroll through illustrations of the available keyboards, or use the right mouse button to select a keyboard from the pull-down menu. A square box on a keyboard denotes a character missing at the server displaying the illustration. Black lettering shows the primary set of characters, alternate character keys are in gray.

Select a keyboard format for a group of users, then, if desired, override for individual users.

Locale format settings

Select Language Format to specify a language for displayed messages and system-generated text. Some languages support reginal dialects or variants. Once you have selected a language, select an appropriate dialect or variant, if available.

The languages available are:

```
de          German
en          English
es          Spanish
fr          French
it          Italian
ja          Japanese
ko          Korean
sv          Swedish
zh          Chinese
```

Region format settings

Select Region Format to specify a regional dialect of a language for displayed messages and system-generated text.

The regions available are:

```
de_AT          German    Austria
fr_BE          French    Belgium
en_CA          English   Canada
fr_CA          French    Canada
de_CH          German    Switzerland
it_CH          Italian   Switzerland
zh_CN          Chinese   PRC
en_GB          English   United Kingdom
en_IE          English   Ireland
ja_JP          Japanese  Japan
zh_TW          Chinese   Taiwan
```

Login Settings

Add Login Settings to a machine, platform, or profile to specify how a user must log in.

If you select Use workstation login, the available login type is smart card. You can specify use of a password.

Note: The user's ID and password must be added with the server's operating system tools. That user ID and the JavaOS for Business user ID must be an exact match.

You can also specify NSL Network Server Login, to add network authentication for users at a machine.

You can specify that guest logins are allowed. This specification and Use network login are mutually exclusive.

You can also specify an inactivity timeout. If there is no user activity for a specified time (in minutes) the user is warned. If there is no further activity, the user is automatically logged out.

Mouse driver setting

Configure this setting to set the interrupts for a PS/2 mouse.

NC PCI token ring driver

Specify this setting for each machine connected with an NC PCI token ring driver. You can accept or change:

Service URL This is the location and name of the driver. The default is `/export/services/NCPCITr.jar`.

Number of receive buffers The default is `16`.

Transmit/receive buffer size The default is 1600.

Number of transmit buffers The default is 16.

Memory-mapped I/O Select true or false from the dropdown box.

Matching Name The default is pci1014,3e.

Logical Name The default is IBMPCITokenRing.

Service Load Time Click Customize to specify when this service is loaded. Select from:

- When discovered.
- When matched. This is the default.
- Upon connection.

After you configure the token ring driver, follow the instructions in “Platform and machine resources and settings” on page 100.

Network File System (NFS)

Add this setting if you want a user to have access to the Network File System. You can accept the default service URL /export/services/NfsFileSystem.jar or change it. You can change the file location, but you cannot change the name.

You must also add NFS AutoMounter to specify a mount point. See “NFS AutoMounter.”

NFS AutoMounter

Add this resource to specify a mount point for the NFS files. The settings are:

Client mount name The default is /HOME.

Protocol type Select either UDP or TCP in the dropdown box.

Note: If using a Windows NT server, select UDP.

Mount path The default is josa80:/export/home/josa81.

NIS Printers

Add this setting to specify the name of the map of Network Information System printers. The default is printers.conf.byname.

NS1000 Smart Card Driver

This setting is part of your smart card configuration for specific hardware. Specify a URL that is the location of the driver. The default is /SERVICE/sc8002.jar.

You must also add and configure base smart card services. See “Base Smart Card Services” on page 104.

NSL Authenticator for Login

Add this resource to specify NSL authentication. The default service URL is /SERVICE/NSLAuthenticator.jar. The logical name is NSLAuthenticator.

OpenCard

Add this resource to specify OpenCard settings. You can accept or change the default service URL. The default is `/SERVICE/opencard.jar`.

Click Customize to configure card agents and terminals. In the Configure Card Services/Terminals you can add or delete or replace classes in the CardTerminals or CareServices lists. Click done when you have completed customizing the settings. There are no defaults for CardTerminals. The information shown below is for planning purposes only:

class `com.ibm.javaos.smartcard.ns1000.NS1000TerminalFactory`

name `SC8002CardTerminal`

type `SC8002`

address `COM2`

There are no defaults for CardServices. The information shown below is for planning purposes only:

class `com.ibm.opencard.service.MFCCardServiceFactory`

Parallel Port Driver

Add a parallel driver to a machine if you plan to attach a parallel device such as a parallel printer or remote printer.

Accept or change the following default parameters:

Service URL The default is `/SERVICES/Parallel284.jar`

Buffer size The default is `4096`

Mode The default is `0`

Timeout (in milliseconds) The default is `300000` milliseconds.

Device I/O port The default is `0x378`

Device IRQ The default is `7`

Logical name The default is `LPT1`

Parallel Printer

Add this resource if you are attaching a parallel printer to a machine.

This resource requires a parallel port driver. See “Parallel Port Driver.”

The parallel printer settings are:

Output port From the dropdown menu, select a port. The default is `LocalLPT1.`

Share Click Share if other machines can send print jobs to this printer.

Host access list Enter the TCP/IP host names of machines that can send print jobs to this printer.

Paper size From the dropdown menu, select a paper size. The default is `Letter` (8.5-inches × 11-inches).

Paper input slot From the dropdown menu, select the paper input mechanism. The default is `Cassette`.

Page description type From the dropdown menu, select a page definition language. The default is `PostScript`.

PCI Token Ring Driver

Add this setting to configure a token ring driver for a machine. If you add this setting for a user or group, it is ignored.

Specify this setting for each machine connected with a PCI token ring driver. You can accept or change:

Service URL This is the location and name of the driver. The default is `/export/services/PCITr.jar`.

Number of receive buffers The default is `16`.

Transmit/receive buffer size The default is `1600`.

Number of transmit buffers The default is `16`.

Memory-mapped I/O Select `true` or `false` from the dropdown box.

Matching Name The default is `pci1014,3e`.

Logical Name The default is `IBMPCITokenRing`.

Service Load Time Click Customize to specify when this service is loaded. Select from:

- When discovered.
- When matched. This is the default.
- Upon connection.

After you configure the token ring driver, follow the instructions in “Platform and machine resources and settings” on page 100.

PowerPC Audio Driver

Add this setting to a user, group, machine, profile, or platform to specify the PowerPC audio driver audio settings.

The PowerPC audio driver settings are:

Service URL The default is `/SERVICES/PPCAudioCS4231.jar.`

Record input device The default is `Mic`.

Record input level Enter a record input level between 0 and 100. The default is `100`.

Output volume Enter an output volume between 0 and 100. The default is `50`.

Device I/O port The default is `0x830`

Device IRQ The default is `10`

Logical name The default is `Audio`

Proxies

Proxies and SOCKS settings define how a system accesses external services such as the Internet or other systems, usually through a firewall. Add proxy and SOCKS information for a group of users or for a profile of machines.

There are no defaults.

Remote Printer

Add this resource to specify a remote printer. You can name the printer (with the name from the NIS map file).

You can specify, in the **Hostname** field, the machines that can use this printer and whether it should be listed in the dialog when those machines make a print request.

Note: This resource requires the “Parallel Port Driver” on page 109 resource.

The parallel printer settings are:

NIS printer If this is an NIS printer, click here.

Printer name Enter the name of this printer. The default is `printer`.

Hostname Enter the TCP/IP hostname of this printer. The default is `hostname` which you can change.

Show in dialog Click here if you want this printer to be listed in the user's list of available printers when a print job is submitted.

Paper size From the dropdown menu, select a paper size. The default is `Letter` (8.5-inches × 11-inches).

Paper input slot From the dropdown menu, select the paper input mechanism. The default is `Cassette`.

Page description type From the dropdown menu, select a page definition language. The default is `PostScript`.

S3 Trio 64 Driver

If your machines include the S3 Trio 64 V+ or S3 Trio 64 V2/DX graphics hardware, configure this video driver.

For other video devices, configure “SVGA driver” on page 113.

Serial Port Driver

Add this resource if you plan to attach serial-attached devices to a machine. This resource is not valid for a user or group.

Accept or change the following default parameters:

Service URL The default is `/SERVICES/Uart6550.jar.`

Matching name The default is `COM1`

Device I/O port The default is `0x318`

Device IRQ The default is `4`

Logical name The default is `COM1`

Serial Printer

Add this resource to configure a serial printer.

The serial printer settings are:

Output port From the dropdown menu, select a port. The default is `Local-COM1`.

Com mode This entry specifies the asynchronous communications parameters. Click to change any default. Then select the new value from the dropdown lists on the pop-up panel. The defaults are:

Baud rate 9600

Data bits 8

Stop bits 1

Parity none

Flow control hh

Share Click Share if other machines can send print jobs to this printer.

Host access list Enter the TCP/IP host names of machines that can send print jobs to this printer.

Paper size From the dropdown menu, select a paper size. The default is `Letter` (8.5-inches × 11-inches).

Paper input slot From the dropdown menu, select the paper input mechanism. The default is `Cassette`.

Page description type From the dropdown menu, select a page definition language. The default is `PostScript`

This resource requires a serial port driver. See "Serial Port Driver" on page 111.

Smart Card Authenticator Format

Add this resource to specify the smart card authenticator for login. The defaults are:

Service URL `/SERVICES/SCAuthenticator.jar`

Property name `ibm.javaos.logonfw.sc.dataPrefix`

Service MRI resource URL `/SERVICES/SCAuthenticatorResources.jar`

Logical name `SCAuthenticator`

SNMP Manager Address

Add this setting to define the SNMP manager address for a machine, profile, or platform of machines. If specified for a user, this setting is ignored. Specify the SNMP manager address; the default port is `162`.

Sound Blaster 16 audio driver

Add this setting to a user, group, machine, profile, or platform to set the SoundBlaster 16 audio settings.

The audio settings are:

Service URL The default is `/SERVICES/PPCAudioSBCompatible.jar`.

Record input device The default is `Mic`.

Record input level Enter a record input level between 0 and 100. The default is `100`.

Output volume Enter an output volume between 0 and 100. The default is `50`.

Device I/O port The default is `0x220`.

Logical name The default is `SBAudio`.

Service Load Time Click Customize to specify when this service is loaded. Select from:

- When discovered. This is the default.
- When matched.
- Upon connection.

SVGA driver

Add and configure the generic SVGA driver to specify video settings.

If your machines include the S3 Trio 64 V+ or S3 Trio Trip 64 V2/DX graphics hardware, see “S3 Trio 64 Driver” on page 111.

System Database Location

Add this setting to specify the location of the system database. This information is required by a machine at bootup. Follow the steps in “Platform and machine resources and settings” on page 100 after you add this setting.

System Image Location

Add this setting to specify the path name of the JavaOS for Business operating system code. You can accept the default in the text field or add your own information.

Notes:

1. You must add this setting to a platform or for each machine in the platform, even if you accept the default.
2. Follow the steps in “Platform and machine resources and settings” on page 100 after adding this setting.

You can specify different images for machines or platforms. For example, you might want to test a new version on a limited number of machines.

TFTP AutoMounter setting

Add this setting to specify a mount point for TFTP.

Note: You must add this setting to a platform or for each machine in the platform, even if you accept the default.

Time zone settings

Add this setting to a user, group, machine, profile or platform to specify a local time zone. The available time zones are:

```
GMT       Greenwich Mean Time time zone.
ECT       European Central Time time zone.
EET       Eastern European Time time zone.
ART       (Arabic) Egypt Standard Time time zone.
EAT       Eastern African Time time zone.
MET       Middle East Time time zone.
NET       Near East Time time zone.
PLT       Pakistan Lahore Time time zone.
IST       India Standard Time time zone.
BST       Bangladesh Standard Time time zone.
VST       Vietnam Standard Time time zone.
CTT       China Taiwan Time time zone.
JST       Japan Standard Time time zone.
ACT       Australia Central Time time zone.
AET       Australia Eastern Time time zone.
SST       Solomon Standard Time time zone.
NST       New Zealand Standard Time time zone.
MIT       Midway Islands Time time zone.
HST       Hawaii Standard Time time zone.
AST       Alaska Standard Time time zone.
PST       Pacific Standard Time time zone.
PNT       Phoenix Standard Time time zone.
MST       Mountain Standard Time time zone.
CST       Central Standard Time time zone.
EST       Eastern Standard Time time zone.
IET       Indiana Eastern Standard Time time zone.
PRT       Puerto Rico and US Virgin Islands Time time zone.
CNT       Canada Newfoundland Time time zone.
AGT       Argentina Standard Time time zone.
BET       Brazil Eastern Time time zone.
CAT       Central African Time time zone.
```

Trace tool settings

Add the trace tool settings to configure the trace settings and trace flags. In the pop up panel, check State to turn on the trace tool. The Host Name is the server where the trace daemon is running. . See Chapter 33, “Using the debug trace” on page 153 for a full description of the trace tool.

If you plan to use the trace tools, this information is required by a machine at bootup. Follow the steps in “Platform and machine resources and settings” on page 100 after you add this setting.

Video driver

Use this setting to specify the video resolution. Select an entry from the dropdown list. The parameters are width × height × color depth.

Chapter 22. Managing users

Use the JCT to define and manage users. In a JavaOS for Business installation you control the way each user logs in, the applications used, and the system behavior the user sees.

User's tasks

Users typically do not have access to any management tasks such as changing a password, adding applications to the desktop, or customizing the desktop. You can, through the JCT, give a user access to multimedia and to printers.

Preparing for users

First, use the JCT to add users to your installation. To add users efficiently, consider defining groups first, then adding users to the group.

Later, as needed, you can define attributes for individual groups or individual users. You might want to define one printer that several users can access. Or, you might want to define a server-connected printer available to several groups. The JCT lets you do these tasks.

Your installation might have been configured to ask each user for a password, to insert a smartcard, for both, or for neither. Giving each new user a list of the login procedures and requirements can be helpful. “Groups and users” on page 98 describes the process to create users and groups.

User's problems

A user who receives a message is likely to request help from the administrator, especially if the message is other than the typical “`Printer is out of paper.`” Users might not know how to respond to messages about communications problems or server responses. See Chapter 27, “User messages” on page 135 for a list of JavaOS for Business messages a user might see. (That list contains only operating system messages and does not include messages from applications that have been added.)

Chapter 23. System Maintenance

Select the Maintenance option of the JCT to set or change system settings, to work with the JavaOS System Database, or the master configuration file.

The master configuration file

The master configuration file contains resource information needed to boot JavaOS for Business. It holds the first information a network computer needs before it has access to the full network computer data in the JSD. “Platform and machine resources and settings” on page 100 tells how to update the master configuration file.

System settings

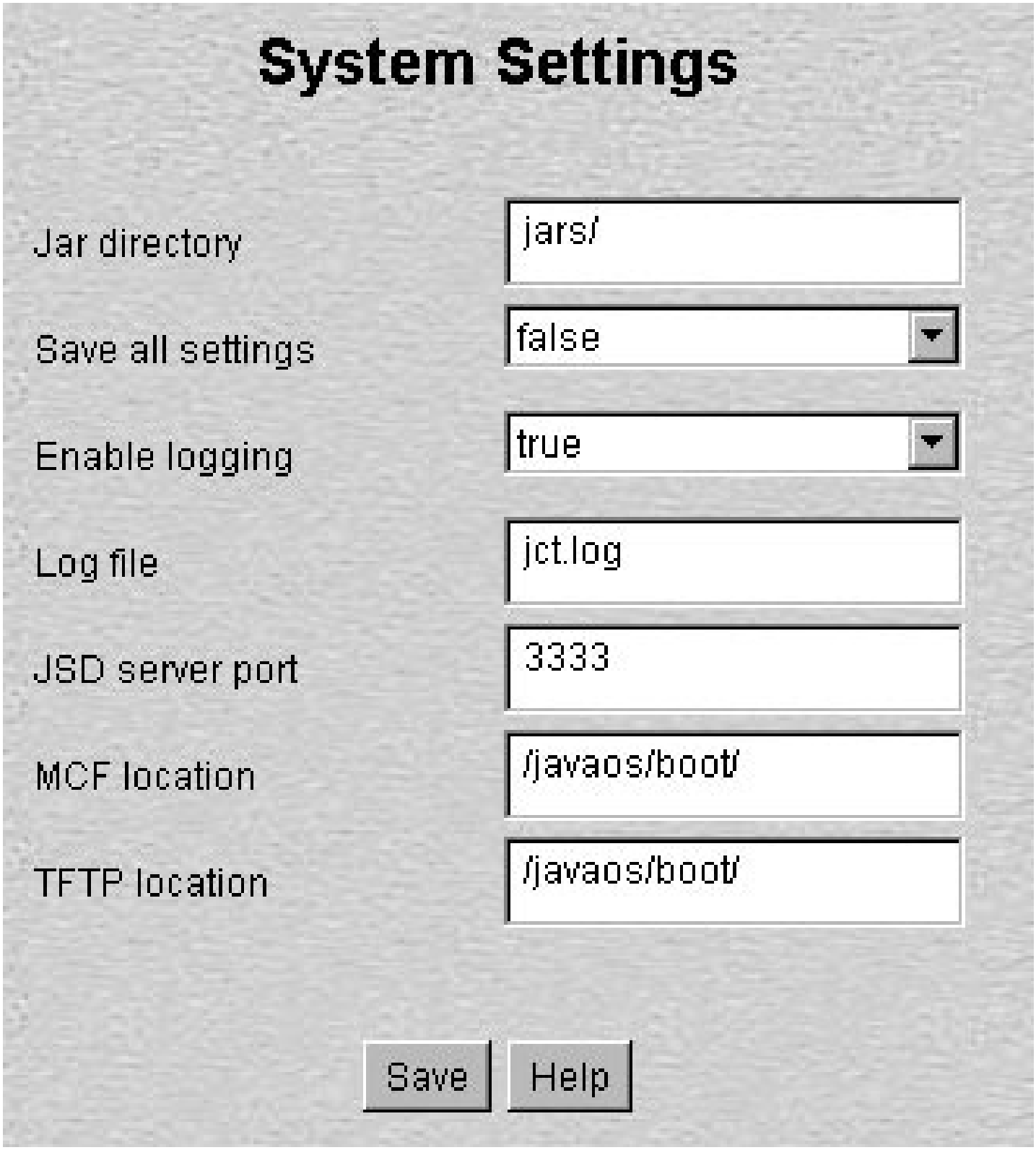

Figure 35. Settings panel for system settings in the JCT

The individual pieces of the JavaOS Configuration Tool are stored in jar files. You can specify a new location for the JavaOS Configuration Tool to find these files. You can accept the default for this setting.

If your installation supports more than one server, you can specify that the JavaOS System Database be stored at a specific server. In the **JSD Server field**, enter the TCP/IP hostname followed immediately by the port number of the connection. You are not required to change this setting.

JavaOS Configuration Tool logging

You can configure the JavaOS Configuration Tool to log runtime exceptions. You can also specify a location for the log file created. See Chapter 36, “The JCT tool log” on page 167 for information about the log.

Working with the JavaOS System Database

The JavaOS System Database (JSD) is the main repository (sometimes called a registry) of information about users, applications, resources, and other data required by the JavaOS for Business operating system. The tools, such as the JavaOS Configuration Tool, for managing system resources are methods for accessing the JSD.

The JSD contains information about each application. A programmer must make sure that each application has the information the JSD requires or that application cannot be made available to users.

The database also contains information about each available device such as printers, network computers (called *machines* in the JCT), modems, sound cards, and so on. Other information defines login procedures, security procedures, and network information. Settings for the system, such as proxy servers, error logging enablement, and some file locations are also stored in the JavaOS System Database.

JavaOS for Business properties

The JavaOS for Business properties are all those entities you can set, or change through the JavaOS Configuration Tool, the netShell, or the command line. You might think of each piece of information in the JavaOS System Database as a property. Properties might also have been defined by application programs, or by the operating system itself when it was installed. This documentation does not list the individual properties; instead it describes tasks you can do by using tools to set or change properties. The most straightforward way to change properties is through the JavaOS Configuration Tool.

Chapter 24. Using the Java command line

The JavaOS for Business command line, at the server, lets you use the following command line options:

-d*javaosProperty=value* Use this command option to set a JavaOS for Business property. For example, to enable the use of the netShell, enter `-djavaos.netShell=true` at the command line. You can also set a property with the netShell **set** command.

-D*SystemProperty=value* Use this command option to set a system property. For example, to specify Central Standard Time as the time zone for your users, enter `-Duser.timezone=CST` at the command line. You can also set a property with the netShell **set** command.

-i*URL* Use this command option to specify a URL. The URL is expanded, the file referred to by the URL is evaluated as if it were, itself, a sequence of JavaOS for Business command lines. The following is a sample URL file.

```
## ## Boot configuration file
## ## Each line contains only one definition
## ## This is a comment line - until the end of the line

## ## Blank lines are ignored
## ## Lines with just whitespace are ignored

-djavaos.apps=http://directory/subdirectory/filename.html
-djavaos.console=true
-djavaos.netShell=true
-ihttp://directory/ subdirectory/filename #includes another URL
```

-u*javaosProperty* Use this command option to undefine a property. For example, to disable the use of the netShell, enter `-ujavaos.netShell=false` at the command line. You can also set a property with the netShell **set** command.

-u*System Property* Use this command option to undefine a system property. For example, to disable Central Standard Time as the time zone for your users, enter `-uuser.timezone=CST` at the command line. You can also set a property with the netShell **set** command.

Chapter 25. Using the netShell commands

This section lists the netShell commands provided with the JavaOS for Business operating system. These commands are typically used to find or change system information at a network computer when a problem has occured or is suspected.

The netShell is available over the serial console and through telnet to port 20000, if it has been configured into JavaOS for Business. Typically you use a terminal emulator on a laptop connected through a null modem on comm port 1.

? | help Display the list of netShell commands.

cdebug *arguments* Use the conditional. Specify **cdebug ?** for help. See Chapter 35, “Using the conditional debug (console output) messages” on page 165 for more information.

checkMemoryAccess true | false Configures AccessableMemory class memory bounds. By default, these bounds are turned off.Use **checkMemoryAccess** when code you are developing that uses this class returns `arrayOutOfBounds` exceptions during execution.

console Toggle console output.

crefs Clear soft references.

debug *classname* **true | false** Enable or disable debug for the specified class.

dump *arguments* Dump information. For example, to view all the current javaos properties, enter **dump props** at the netShell command line. Specify **dump ?** for help.

eatmem Consume memory.

etherstat Get Ethernet statistics.

exit | close | quit Exit from NetShell.

file File test commands.

gc Force synchronous garbage collection.

ipstat Get IP statistics.

java *classname arguments* Run built-in class with arguments.

jitset *threshold* Set an absolute JIT threshold for benchmarks.

jperf *arguments* Use the performance and debug trace. Specify **jperf ?** for help.

keys Show control key equivalents.

logger Start the error logger command line interface. See Chapter 32, “Using the error log” on page 149.

monitorstat Monitor statistics.

mount server:/path/name /LOCALPREFIX Mount an NFS file server.

prom Invoke debugger console.

qreboot Do a quick reboot of a network computer.

read *host port bytes* Test TCP read performance.

reboot Do a full (slow) reboot of a network computer.

redirect Redirect console output here. This is a toggle command.

redirnet *netaddress* Redirect console output to the specified address.

ref *classname* Find all instances of *classname*.

reflect Run reflector tests.

resolve *name* Perform hostname lookup.

run Run a built-in JavaOS for Business test.

runf Run finalization.

set *commandline_arguments* Set system properties. For example, to set NIS to false, enter **set -djavaos.nis=false** at the netShell command line.

statccsw *number_of_seconds* Get stat thread count and context switches.

statThreads Get thread statistics.

steal *size* Create ungarbage-collectable object of *size*.

svga Get or set SVGA register (x86).

sysinfo Get system information.

tcp *host port packets bytes* Ping test with TCP.

tcpstat Get TCP statistics.

top Get UNIX top-like Java thread statistics.

trace *arguments* Trace. Specify **trace ?** for help.

udp *host port packets bytes* Ping test with UDP.

unmount LOCALPREFIX Unmount an NFS file server.

unsteal Return all space taken in a previous **steal**.

url *URL* Read the specified HTML file from the http server.

write *host port bytes* Test TCP write performance.

Part 4. Messages

This information lists and explains the messages that administrators at the server or users at the clients can receive. It lists only those messages sent from the operating system, not messages sent from specific applications. See the information provided with a specific application or applet for its unique messages.

Numbered messages are listed first in numerical order. Unnumbered messages are then listed in alphabetical order. A word in *italics* is a variable that will change depending on the action that displayed the message.

Chapter 26. Server messages

This section lists and explains the operating system messages that administrators at the server can receive. See the information for specific applications and applets for their unique messages.

Numbered server messages

1000.10 Skipping non-displayable Property: *property_name***.**

Explanation: The Java bean is missing a property editor.

User Response: Contact the JAR file owner; settings might not be displayed.

2000.10 No Java Beans found in jar file: *JarFile_name***.**

Explanation: The jar file contains no Java beans.

User Response: Contact the JAR file owner; settings might not be displayed.

2000.20 Missing JarFile BeanContext: *BeanContext_name***.**

Explanation: The jar file is corrupt.

User Response: Contact the JAR file owner; settings might not be displayed.

2000.30 Unable to load JarFile class *class_name***.**

Explanation: The jar file is corrupt.

User Response: Contact the JAR file owner; settings might not be displayed.

2000.40 Error trying to read JarFile: *JarFile_name***.**

Explanation: The jar file is corrupt.

User Response: Contact the JAR file owner; settings might not be displayed.

2000.50 Unable to load any resources for bundle: *bundle_name***.**

Explanation: The help information cannot be loaded.

User Response: Contact the JAR file owner; settings might not be displayed.

2000.60 Warning — File name does not match class name: *File_name***,** *class_name***.**

Explanation: The jar file is corrupt.

User Response: Contact the JAR file owner; settings might not be displayed.

2000.70 Warning — Incorrect class format: *class_format*.

Explanation: The jar file is corrupt.

User Response: Contact the JAR file owner; settings might not be displayed.

2000.80 Error in MANIFEST.JCT at line: *line_number*.

Explanation: The jar file is corrupt.

User Response: Contact the JAR file owner; settings might not be displayed.

2000.90 Error in MANIFEST.MF at line: *line_number*.

Explanation: The jar file is corrupt.

User Response: Contact the jar file owner; settings might not be displayed.

2000.100 Unable to load class: *class_name*.

Explanation: The jar file is corrupt.

User Response: Contact the jar file owner; settings might not be displayed.

2000.110 Unable to introspect class: *class_name*.

Explanation: The jar file is corrupt.

User Response: Contact the jar file owner; settings might not be displayed.

2000.120 Unable to access class: *class_name*.

Explanation: The class is private or protected.

User Response: Contact the jar file owner; settings might not be displayed.

2000.130 Unable to get property value for: *property*.

Explanation: The Java bean property has no get() method.

User Response: Contact the jar file owner; settings might not be displayed.

3000.20 No HTML entries found for: *file_name*.

Explanation: The applet configuration is missing a HTML file.

User Response: Contact the jar file owner; settings might not be displayed.

3000.30 Unsupported JarFile Context: *Context_name*.

Explanation: The manifest.jct file is missing context.

User Response: Contact the jar file owner; settings might not be displayed.

3000.40 Unable to obtain list of Jar Files: *list_name*.

Explanation: The jarfile directory contains no jar files.

User Response: Place all jars in the jars/ direcotry, also check the file permissions.

3000.50 Successful completion.
3000.60 Settings saved successfully.
3000.70 Unable to save settings.

Explanation: There is a problem with the Java system directory, or a Java bean property.

User Response: Check Java system directory errors, also make sure the port settings are correct.

3000.80 Cancel changes and quit.
3000.90 Save changes and exit.
3000.100 You must change a setting before saving.

Explanation: You selected a change settings panel, but made no changes.

User Response: Click Cancel to exit without making a change.

3000.110 Unable to save settings. database or JarFile error.

Explanation: There is a problem either with the Java system directory or the Java bean property.

User Response: Check for Java system directory, also make sure port settings are correct.

4000.10 Invalid JSD entry: *JSD_entry*.

Explanation: The Java bean and the Java system directory entry are incompatible.

User Response: Contact the jar file owner; settings might not be displayed.

4000.20 Error restoring JarFile from JSD entry: *JSD_entry*.

Explanation: The Java bean and the jar file are incompatible.

User Response: Contact the jar file owner; settings might not be displayed.

4000.30 Unable to create JSD entry for: *JSD_entry*.

Explanation: Java beans are not serializable.

User Response: Contact the jar file owner; the settings might not be saved.

4000.40 Unable to create JSD entry for: *JSD_entry*.

Explanation: Java beans are not serializable.

User Response: Contact the jar file owner; the settings might not be saved.

4000.45 Invalid JSD entry: *JSD_entry*.

Explanation: The system database entry is damaged.

User Response: Follow your local procedures for obtaining support.

4000.50 Unable to set property value for: *property_value***.**

Explanation: Java bean property setter is invalid.

User Response: Contact the jar file owner; the settings might not be saved.

4000.60 Factory Method is null: *factory_method***.**

Explanation: The MANIFEST.JCT JSDEntryNameFactory is bad.

User Response: Contact the jar file owner; the settings might not be saved.

4000.70 Unable to locate factory method: *factory_method***.**

Explanation: The MANIFEST.JCT JSDEntryNameFactory lost.

User Response: Contact the jar file owner; the settings might not be saved.

4000.80 Unable to invoke factory method: *factory_method***.**

Explanation: The MANIFEST.JCT JSDEntryNameFactory access.

User Response: Contact the jar file owner; the settings might not be saved.

4000.90 Security: Unable to invoke factory method: *factory_method***.**

Explanation: The MANIFEST.JCT JSDEntryNameFactory access.

User Response: Contact the jar file owner; the settings might not be saved.

4000.100 Factory Method does not exist: *factory_method***.**

Explanation: The MANIFEST.JCT JSDEntryNameFactory access.

User Response: Contact the jar file owner; the settings might not be saved.

4000.110 Unable to access factory method: *factory_method***.**

Explanation: The MANIFEST.JCT JSDEntryNameFactory access.

User Response: Contact the jar file owner; the settings might not be saved.

4000.120 Error during search at: *search_location***.**

Explanation: The jar file is in the Java system directory, but not on disk.

User Response: Contact the jar file owner; the settings might not be saved.

4000.130 Invalid directory: *directory_name***.**

Explanation: The jar file directory is invalid.

User Response: Check the jar directory permissions. Make sure the jar directory exists.

10000.10 Could not load user properties file: *file_name***.**

Explanation: The file does not exist, or the file format is incorrect.

User Response: Make sure *file_name* exists and is in the correct format.

10000.20 Could not load MCT properties file: *file_name***.**

Explanation: The file does not exist, or the file format is incorrect.

User Response: Make sure *file_name* exists and is in the correct format.

10000.30 Could not connect to the JSD server at: *server_name***.**

Explanation: The server might be unavailable, or the server name *server_name* is incorrect.

User Response: Make sure the server is available or correct the server name and make sure the port is correct.

10000.40 Exception: *exception_name***.**

User Response: Follow local procedures for problem determination.

10000.50 The specified entry cannot be found on: *server_name***.**

Explanation: The entry was not found.

User Response: Follow local procedures for problem determination.

10000.60 Invalid ID specified in call to writeMCF: *integer***.**
10000.70 System database cannot be accessed on this JVM; trying to create one.

Explanation: This is an informational message.

10000.80 System database cannot be created in this JVM.

User Response: Follow local procedures for problem determination.

10000.90 Exception getting coalesced tree: *exception_name***.**

Explanation: This is an informational message.

User Response: Follow local procedures for problem determination.

10000.100 Could not obtain a coalesced tree. Returning.

User Response: Follow local procedures for problem determination.

10000.110 Exception while getting bootimage entry.

Explanation: The bootimage entry is not configured correctly in the JSD.

User Response: Check the configuration of the bootimage entry.

10000.120 Exception while writing bootimage entry: *exception_name*.

Explanation: The MCF file cannot be created or written to because of permissions or security settings.

User Response: Check the file and directory permissions and the security settings.

10000.130 Exception while creating boot tree: *exception_name*.

Explanation: The configuration object could not be created.

User Response: Follow local procedures for problem determination.

10000.140 Exception while writing device information: *exception_name*.

Explanation: The device information cannot be created or written because of permissions or security settings.

User Response: Check the file and directory permissions and the security settings.

10000.150 Problem while creating a subdirectory: *subdirectory*.

Explanation: The subdirectory cannot be created because of permissions or security settings.

User Response: Check the file and directory permissions and the security settings.

10000.160 This is a configuration file for: *configuration_file*.

Explanation: This is an informational message.

10000.170 Warning: Bootimage entry not found.

Explanation: The bootimage entry is not configured correctly in the JSD.

User Response: Check the configuration of the bootimage entry.

11000.00 Buliding MCF file for: *machine_or_platform_name*.

Explanation: This is an informational message.

11000.10 Done building MCF file for: *machine_or_platform_name*.

Explanation: This is an informational message.

11000.20 See the log file. Failure building MCF file for: *machine_or_platform_name*.

Explanation: This is an informational message.

User Response: See the log file.

11000.30 Unable to write the MCF file: *mcf_file_name*.

Explanation: The MCF file cannot be created or written because of permissions or security settings.

User Response: Check the file and directory permissions and the security settings.

11000.40 Problem with JSD when writing the MCF file for: *machine_or_platform_name***.**

Explanation: Either the connection with JSD failed or the JSD is not configured correctly.

User Response: Follow local procedures for problem determination.

11000.50 Machine or platform is not defined or selected: *machine_or_platform_name***.**

Explanation: Either you have not defined this machine or platform yet in the JSD, or you have not selected a machine or platform.

User Response: Check the JSD configuration, or select a machine or platform.

11000.60 Selected: *machine_name***.**
11000.70 New MCF location is: *pathname***.**
11000.70 The following exception occurred while accessing file: *exception_name***.**

User Response: Check the file and directory permissions and the security settings.

Unnumbered server messages

Unable to find version information.

Explanation: The **bldlevel** tool was unable to find version information in the file.

Intervention required — printer *printer_name* **not responding.**

Explanation: The specified printer is not responding.

Intervention required — printer *printer_name* **is offline.**

Explanation: The specified printer is offline.

Intervention required — printer *printer_name* **is out of paper.**

Explanation: The specified printer is out of paper.

Intervention required — printer *printer_name* **reports an error.**

Explanation: The specified printer reports an error.

LocalContentTransportOutputStream - abortFlag set.

Explanation: See one of the following messages about the printer problem.

- Only one data source allowed.
- Spooling problem.
- Cannot get hostname.
- Unforseen error.

lpd: Job sent to printer *printer_name* **at host** *host_name***.**
lpd: Printing cancelled.
lpd: Sending to printer *printer_name* **at host** *host_name***.**
lpd: Remote printer refused job.
lpd: Unexpected problem.

Chapter 27. User messages

This information lists and explains the JavaOS for Business operating system messages that users might receive. Messages might be displayed by:

- Multimedia components
- Printers
- Communications components
- Security

Users also might click on a help button to receive context-sensitive help. Help is provided by many specific applications.

Users do not see the messages in the format shown here; a user sees the message in a dialog box. The user can typically click an OK button to respond to a message. See the information provided with specific applications and applets for their unique messages.

IBM PCI TokenRing Message

Explanation: This initial message identifies a TokenRing Message.

User Response: Read the following message and follow its instructions.

Adapter error. Click OK to reset. EISR =

Explanation: Click OK to reset.

User Response: Click OK. If the message is displayed again, ask your system administrator for help.

Adapter error. Check connectors, then click OK to reset. LAN error =

Explanation: Click OK to reset.

User Response: Click OK. If the message is displayed again, ask your system administrator for help.

Inactivity alert. Move the mouse or press a key to prevent an automatic logoff.

Explanation: The user's profile is set to warn the user when a specified time has passed with no network computer activity.

User Response: The user can move the mouse or press a key to remain active. After a continued period of non-activity, the user might be automatically logged out.

Failed login.

Explanation: The user was not allowed to log in.

User Response: The user should make sure the correct user ID, password, and PIN were entered.

The attempted default server login has failed. Press Next to retry or contact the system administrator.

Explanation: The user was not allowed to log in.

User Response: The user should make sure the correct user ID, password, and PIN were entered. The user should then retry or call the system administrator.

Login unsuccessful. The user ID and password information are incorrect.

Explanation: The user was not allowed to log in.

User Response: The user should make sure the correct user ID, password, and PIN were entered. The user should then retry or call the system administrator.

Login unsuccessful. The user ID and password information are incorrect. This is the last try.

Explanation: The user was not allowed to log in.

User Response: The user has tried to log in the specified number of tries. The user should make sure the correct user ID, password, and PIN were entered. The user should then retry or call the system administrator.

Unable to connect to server The server information is incorrect, the server is inactive, or there is a network problem. Contact your system administrator.

Explanation: The user was not allowed to log in.

User Response: The user should retry or call the system administrator.

Network login. The login was cancelled. Enter your network login information.

Explanation: The user was logged out by the system.

User Response: The user should log in again, if desired.

Part 5. Troubleshooting JavaOS for Business

The following sections describe troubleshooting JavaOS for Business, the tools available, and how to get help, if necessary. These sections describe only JavaOS for Business; for problems with applications, see the documentation provided with the specific application.

Chapter 28. Determining the source of a problem

Typically, the first step in problem determination is deciding whether the problem is caused by hardware or software.

This section describes troubleshooting, ways to approach problems, some paths through troubleshooting, and leads you to the section on troubleshooting tools.

The troubleshooting process

The goal of troubleshooting is a continuously available system with productive users. To meet this goal, you need procedures that help you isolate and solve problems quickly and efficiently. You want to clear up user errors swiftly, isolate failing hardware, and solve system-wide problems. Figure 36 shows a problem handling flow. Most problems can be handled quickly and locally.

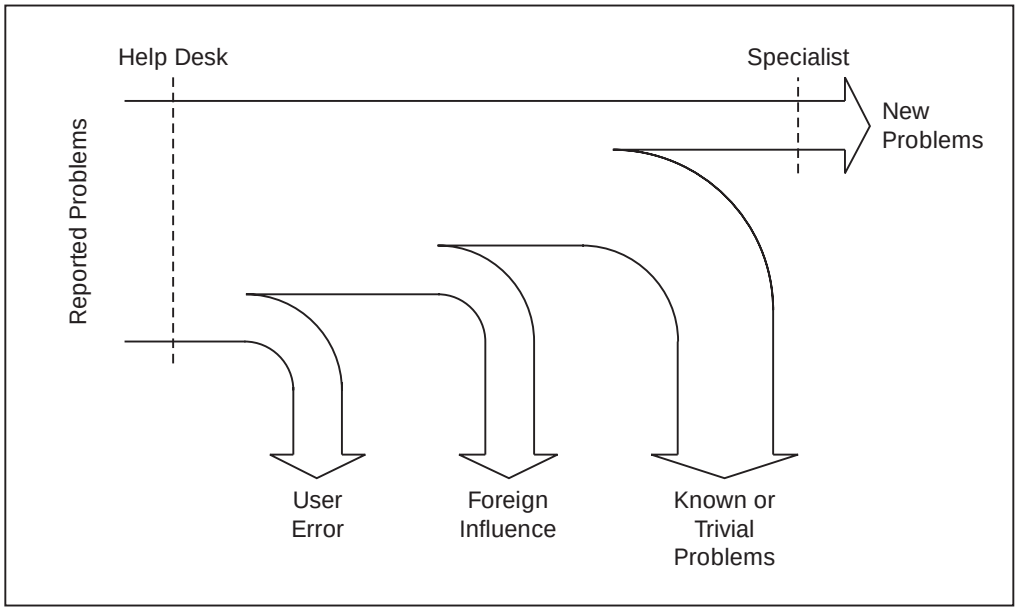

Figure 36. Flow of eliminating problems by solving small ones quickly

Many user problems occur when users do not understand how to use the devices or the applications. Prevent these problems by giving users clear instructions and training. A help desk might be able to filter out most user errors, temporary network outages, and trivial problems, such as expired passwords or out-of-space situations.

An approach to a problem situation is to list the symptoms and the reactions that the system shows as problems.

Checking the problem environment

Sometimes the severity of a problem is different for different people. For example, a user might be unable to login because of a simple forgotten password or because of a complex network failure. And, the reported severity often depends on the discoverer. To a user forgetting a password is a tragedy. To an administrator, that's a minor annoyance.

Ask these questions when a problem is reported:

- Is it really a problem or might the user be confused?
- Did it ever run correctly before?
- When did the problem appear the first time?
- How did the problem become visible?
- Is more than one user affected?
- Can the problem be re-created?
- What has been changed recently?
- Is the foundation (the operating system, the communications programs, the network) operating correctly?

These questions let you know the environment and the participants in the problem.

Selecting a JavaOS for Business tool

JavaOS for Business provides a set of tools to help solve problems, examine performance, and verify system information.

Error logs

The error log on the server can log operating system events both from the server and from network computers. See Chapter 32, “Using the error log” on page 149 for a description of the error log capability. If you suspect that errors are occuring at a network computer, check the error log at the server for that device. “Using the error log viewer” on page 150 describes the error log viewer.

Conditional debug messages

The conditional debug messages can let you know what is happening in operating system code. The conditional debug is a control for C language programs `printf()` and Java `System.out.println` messages to the serial console output device.

See Chapter 35, “Using the conditional debug (console output) messages” on page 165. You can use the netShell **cdebug** commands to specify major and minor numbers to turn on and off the sending of these messages; the code need not be recompiled.

Debug trace

The debug trace is another mechanism for tracing an application. The debug trace sends messages to a network-resident daemon that catches the messages and writes them to a file. The debug trace messages are binary data describing system events. The jstrace tool formats the trace messages so that you can read them. You can use the netShell **jperf** command to start or stop the debug trace.

ICAT debugger

The ICAT debugger is a source-level debugger for C, assembler code, Java, and JVM applets and applications. The debugger runs outside of a JVM on either Windows NT or on OS/2 Warp 4. The debugger uses a Java debug probe to manipulate the Java application, and uses TCP/IP to communicate with the Java debug probe that runs inside a JVM. This design allows for remote debugging; that is, the target application and Java debug probe can run on a separate machine from the host machine of the debugger itself. You can launch a Java application with the debugger. The debugger presents windows (Source, Storage, Call Stack, Variable Monitors, and so on) that display the current state of the program being debugged. You can step through the application, run the application, catch breakpoints, and so on. The supported debug file formats for the native-method DLLs that you may choose to debug include HLL (IBM VisualAge) and CodeView. See *ICAT for JavaOS* for full instructions.

The ICAT debugger requires:

- A debug build
- Source code access (usually through NFS)
- Local javaos.debug file
- A serial or null-modem connection

Isolating the Source of a Problem

Try to re-create the problem, if you can. If not, try to ensure that the suspected cause of the problem can be isolated or at least tested in such a way that you can confirm that you have found the real problem. And, when you've solved the problem, you can be sure that your solution is valid.

Examine the log files to find any reported errors. The log files maintained by the components often show the regular activity, as well as errors that occurred. See Chapter 32, “Using the error log” on page 149 for information about error logs created by JavaOS for Business.

Check for any other automatic or manual process that might affect your problem. Check to see if the application or task that is in trouble depends on other application data, termination, or any other dependency that might cause the problem.

Hardware or firmware

For a hardware problem, refer to the documentation that came with the device.

Chapter 29. Checking your software level

Use the NetShell **sysver** command to check the level of each package in the JavaOS for Business system. This information is commonly required when you make a service call. **sysver** returns:

- Package-Title: Title of the package
- Package-Version: Version number
- Package-Vendor: Vendor company or organization
- Specification-Title: Title of the specification
- Specification-Version: Version number
- Specification-Vendor: Vendor company or organization

A sample output of **sysver** is:

```
Manifest-version: 1.0

Name: ibm/abc/
Specification-Title: IBM ABC package
Specification-Version: 1.1.3
Specification-Vendor: IBM
Package-Title: ibm.ABC
Package-Version: @#IBM:9.53#@##1##Mon May 19 13:32:47 1998 bld003:
                 :en:us:::@@(C) Copyright IBM Corp. 1998
Package vendor: International Business Machines Corp.
```

Use the **bldlevel** tool to check the level of JavaOS for Business files at your fileserver. This information might be required when you make a service call. Invoke **bldlevel** on javaos.load and all jar files for downloadable device drivers or other loadable system components. **bldlevel** returns information similar to the following:

```
#
#version control infomation
language         en                #ISO 639 (2 char) language code
country          en                #ISO 639 (2 char) country code
vendor_short_nm  IBM               #Vendor short name
vendor_long_nm   International_Business_Machines_Corp  #Vendor Long name
build_level      0.0               #build Level string (m.n) [numeric]
build_sub_lvl    0                 #build sub-level
driver_name      g17               #driver name
cmvc_release     javaos.gem        #CMVC release
#
```

If the command cannot find the information in a file, an error message is displayed. See Chapter 26, “Server messages” on page 127.

Chapter 30. Reporting problems

If you cannot solve a problem locally at your installation, you might make a service call to your service provider. Before you make that call, you can establish procedures that make problem resolution more efficient.

Designate specific personnel who are to make service calls to eliminate multiple calls and confusion that can result if several people call about one problem.

Have standard information available in an easily accessed format. Your provider is likely to ask you for your customer or order number, any license number on the product, the release level, date of installation, and about any modifications you may have made. It's a good idea to store this information separately from the equipment it is describing.

You are also expected to have a description of the problem, the circumstances leading up to the problem, and any other supporting information you can gather.

A description of your system's normal operation is useful for comparison when you make a service call.

Chapter 31. Booting the network computer

This information describes the JavaOS for Business server and network computer bootup process and tells what to watch for. All your hardware should be installed, machines should be connected, and addresses should be assigned before you boot up a network computer.

When you first start a network computer, it typically goes through hardware and communications test procedures specific to that network computer's manufacturer before it loads the JavaOS for Business operating system. If hardware or communications problems occur at the start of bootup, you see indications before the TFTP message displays. In that case, consult the documentation that accompanied your hardware.

Users of network computers can roam from machine to machine. Bootup information is thus separated, at a simplistic level, into machine information and user information. All information that identifies a machine and a user is stored in fields in the JavaOS System Database. When a network computer boots up, it first loads machine information from the server's machine information. Then when a user starts to log in, information is needed from the user data.

Typical machine information

Typical machine information includes the machine type and MAC address, the available servers for that machine, and the platform (the group the machine was assigned to with the JavaOS Configuration Tool.

Typical user information

Typical user information includes the user's ID, group memberships, application availability, desktop settings, and security levels available.

Network computer bootup

During bootup, JavaOS for Business running at the server loads the operating system from the server to the network computer's memory, initializes that individual network computer, and prepares the system for use. When the bootup process displays the a login window, the bootup process has completed all hardware and communication checks. This process is described in more detail below.

Starting the bootup process

The bootup process consists of several components:

- Boot server, a typical TCP/IP machine providing the DHCP protocol.
- Boot image server
- Machine configuration server
- User configuration server
- Authentification server

These components might reside on separate computers and communicate, or might reside all on the same computer.

network computer bootup details

A network computer is booted only when a user turns on the device. At that time, data is transmitted from the server. Any bootup failure results in error messages displayed at the server.

The network computer uses the appropriate network broadcast protocols to start a DHCP conversation with the boot server. The DHCP conversation provides an IP address, the name of a server from which to retrieve the master configuration file (MCF), and the name of the MCF.

The MCF has the information required to boot the network computer to the point where it has a working TCP/IP protocol stack.

Once the network computer knows the boot image server address and the name of the MCF, it can contact that server and download the MCF, typically using TFTP. Then, using the URLs in the MCF, the network computer downloads the appropriate kernel image and loadable services from the boot image server. The loadable services and the MCF are placed in the network computer's memory in a filesystem called ROM by the boot image server:

MCF /ROM/BootConfigFile

services /ROM/services/*service_name*.zip

Machine configuration URL /ROM/MachineConfig.html

The boot monitor now begins to execute the JavaOS for Business kernel image. The loadable services files are loaded and initialized using the MCF in the network computer. The network computer uses the IP address provided, and gets enough information to allow a user login.

The boot sequence is now essentially complete.

Chapter 32. Using the error log

This section describes the logs available with the operating system, tells how to start and stop logging, and tells how to use the error log viewer, described in “Using the error log viewer” on page 150.

Error messages provide valuable information for determining the source of a system failure by documenting error conditions that might have led up to a system failure.

JavaOS for Business can transmit error messages across the network from a network computer to an error log daemon active at a server. The error log daemon then spools the messages to error log files. The operating system creates error logs to tell system administrators and service personnel that there may be system errors. The server accepts the messages and places them in appropriately named directories and files to keep information from each network computer separate.

By default, the error log is turned on when the system is booted up. To turn off error logging, see “Error log commands” on page 151. If an error log does not already exist for a specific network computer, one is created when an error message is received. Subsequent messages are appended to existing error log files.

The error log is designed for use after the communications stack has been initialized. Errors that occur before the communications stack is up should be handled with conditional debug messages (see Chapter 35, “Using the conditional debug (console output) messages” on page 165) or the debug trace (see Chapter 33, “Using the debug trace” on page 153).

Errors encountered during normal system operation are recorded in the error log. Such errors include null pointers, low memory situations, file opening errors, and similar errors. (Out of memory errors might not be logged because creating a log entry itself requires allocating some memory.)

There is assumed to be no local disk storage on the network computers and memory on the network computers might be constrained: therefore system error logs are transmitted over the network to a server. The goal of the error log entry is to supply sufficient information to make problem determination and locating the source of the problem easier.

The resource bundles contain language-specific messages and information. The error log viewer needs access to the JavaOS for Business resource bundles in order to display error log messages in the currently specificed locale. The error log viewer displays an error message if it cannot find a resource bundle. Your installation must have installed the resource bundles where they can be found by the error log viewer.

Enabling error logging

You can enable or disable error logging through the JavaOS Configuration Tool. To enable error logging:

1. Click on Maintenance.
2. Click on System settings.
3. Click on Enable logging.
4. In the box below the check box, enter the name to store error log information in, or leave the default file name.
5. Click the Save button to save your changes.

To disable logging, follow the procedure above, but clear the check mark from Enable logging.

The error log daemon

The error log daemon runs at the server. It waits to catch error messages from the active network computers. You must have the error log daemon active at the server in order to catch error messages from the network computers.

The TCP/IP packets are sent to the error log daemon, at the IP address and port number provided in the Java System Database or specified with netShell. The server from which the system got its load image is the default IP Address; the default port number is 7786. The error log daemon accepts error log packets on a port designated when the daemon is started. The packets are stored in the directory you designate, in a file named with the MAC address of the client originating the error message. Thus, error logs from different clients on the network are kept separate. Subsequent error log messages are appended to existing error log files. A new error log file is created if one does not exist.

The LogSrv.cfg file is supplied zipped as part of a group of system files. Before you start the error log daemon, unzip the file from

`com/ibm/jossrv/elsrvdemon/LogSrv.cfg`

and edit the file as desired. Save the LogSrv.cfg file in the directory from which you plan to start the error log daemon. (In the zipped file name, `elsrvdemon` is the correct spelling although it does not correspond with the usual spelling of *daemon*.)

Type the command:

`java LogSrv [ LogSrv.cfg ]`

to start the daemon. LogSrv.cfg is a configuration file with the following contents:

```
dir=Absolute_directory_path_name_of_log_file
port=port
filesize=filesize
```

If you do not specify a configuration file, or if you specify the default LogSrv.cfg, the defaults are:

```
dir=the current directory
port=7786
filesize=50 KB
```

Using the error log viewer

The error log viewer, at the server, formats and displays error log records. The error log viewer is a server-based Java application that formats the error log record by record by taking the error message number and displaying (in the language of the current locale) the error message and the rest of the supporting information for that record and all other records in the error log.

If the optional bundleId is included in the record, the error log viewer attempts to find a file named in the bundleId string, appended with the current country and locale codes, in its CLASSPATH. If the information is found, the viewer uses that file to display the errors. Otherwise, the error number is displayed without the error message text. See Figure 37 on page 151 for sample error log viewer output.

Starting the error log viewer

Type the command:

`java LogViewer`

align-center NAME='ERRLOG'.Sample error log output screen capture.

Figure 37. Sample error log output

at the server to start the error log viewer. Then, select the log file you wish to view. If you select other than a log file for viewing, a blank window and an error message are displayed.

Error log commands

Use netShell commands to control creation of error logs and the entry of information in the logs.

The NetShell **logger** command and its operands control the error log actions:

logger *?* Displays the valid logger commands

logger host *host_port* Sets or changes the host and port number for the error log facility to reach the error log daemon. Use this command to specify where the log records are stored.

logger ignore *num id mth* Tells error log to ignore a specific request for an error log message. You can specify to ignore up to 3 dump triggers.

Be careful to prevent a JavaOS for Business system from flooding the local network with inconsequential error log messages. Consider, after you have established what is normal for your installation, setting logger ignore specifications for events you know are inconsequential.

The operands are:

num The error number

id The error ID (-1 for any ID)

mth The method name, including the class name and module name separated by dots. For example, sun.javaos.Shell.logger.

logger ignore reset *num* Deletes the specified ignore trigger. Specify **99** to reset all triggers.

logger on Turns on the error log. This is the default.

logger off Turns off the error log.

logger info Displays the error log on or off state, the host name and port, and lists the dump and ignore triggers.

Error log entry contents

Error log entries are ASCII formatted strings. Each entry in the error log has the following contents. Each entry ends with **\n**.

Sequence number Number of error messages issued since boot up (generated by Logger).

Error number Corresponds to an entry in the translated Error Message Resource Bundle (input parameter).

Error ID Unique identifier for each error message (input parameter).

User ID Name of user logged into network computer (retrieved by Logger).

Bundle ID Unique resource bundle identifier (will have the current country and locale appended to it to create a file name that will be searched for in the error log viewer's CLASSPATH) for displaying the translated error message.

Method name Name of the method and class name (class.method) and the module name where Logger was called from (retrieved from the Java stack by Logger).

Time stamp When the error occurred.

Thread name JavaOS for Business thread name.

Thread priority JavaOS for Business thread priority number.

Thread group JavaOS for Business thread group name.

Severity Error severity: Infomational, Fatal, or Non-fatal. This is an input parameter.

Supplemental Message Optional supplemental message string, not translated. This is an input parameter.

Chapter 33. Using the debug trace

Using the proper tools helps you to find the cause of problems. A JavaOS for Business debug trace shows the sequence of methods called and exited from during an applications execution in a system. Debug trace outputs data to a buffer, and subsequently to a server-resident file so that you can examine what was happening in a system in an arbitrary period before a failure.

The debug trace is triggered through flags that you can turn on and off. When the flags are on and the code is executed, a call to fire a trace hook, or output a debug message, is activated. The performance trace hooks are inserted into key areas of JavaOS for Business to determine performance numbers for a particular function such as idle time or task switching.

Debug trace hooks document:

- The entry into a method
- The method's entry parameters
- The exit from a method
- The method's return value
- Any other intermediate information showing state or other changes

Performance and trace hooks *fire* or write into a trace buffer in JavaOS for Business memory and are copied out later for processing and analysis.

You can use one of the following mechanisms to control the traces:

- C Language APIs and Java methods for programmatic control
- The JavaOS System Database
- netShell for manual control while the system is running

The debug trace daemon

The debug trace daemon runs at the server. You must have the error log daemon active at the server in order to catch error messages from the network computers.

The TCP/IP packets are sent to the debug trace daemon, at the IP address and port number provided in the JavaOS System Database or specified with netShell. The server from which the system got its load image is the default IP address; the default port number is 7786. The daemon accepts packets on a port designated when the daemon is started and stores them in a file named with the MAC address of the client originating the debug trace message. The file is stored in a directory that is also designated at daemon startup. Thus, trace events from different clients on the network are kept separate. Subsequent messages are appended onto already existing files, a new file is created if one does not exist.

Starting the debug trace daemon

The command line input parameters for the daemon are the directory on the server in which to create and append to files, and the port number for it to listen on. To start the daemon, type:

```
java traceServer port
```

The trace daemon must be active to accept trace buffer writes from a network computer through TCP/IP sockets on a port. The trace daemon can handle multiple trace buffer writes, creating a new file for each trace buffer and appending consecutive numbers to the file name. For example a sequence of file names might be: `josa21_1.raw`, `josa21_2.raw`, and `josa21_3.raw`.

If you attempt tracing but the trace daemon is not running, the socket is not successfully opened, and no trace buffer is transferred. The trace sends a message to the JavaOS for Business console indicating that the trace buffer could not be sent.

Using the debug trace formatter

Use the debug trace formatter to format the trace buffer information into readable format. A buffer formatted with **jstrace** is sufficient for viewing debug trace data. You can use the

```
java jsformat
```

command to format the trace buffer into a readable format. The complete syntax is:

`java com.ibm.jossrv.perf.trace.jsformat` *`tracefile hdffile  formatted file`*

where:

tracefile The name of the .raw binary file collected by the trace daemon.

hdffile The name of the .hdf file describing trace formatting guidelines.

formatted_file The name of the formatted output ASCII file.

The sample .hdf file provides formatting guidelines for all trace hooks.

Turning on and off the traces

You can turn off individual performance, debug trace, and conditional debug messages dynamically by turning off their respective major code flags. When you build JavaOS for Business you can include **-NOTRACE**. If you this option, stub modules are included in the source path. The files then contain public final static boolean values set to false. You can specify **CDEBUG = TRUE**. Java code that uses performance, debug trace, and conditional debug messages have a conditional statement that the Java compiler can evaluate at compile time, which, in this case, causes the body of the conditional statement not to be compiled. This effectively turns off all references to the traces.

Note: Including the **-NO***x* flag in the build when it previously was not, or removing it when it previously was specified in the build requires a clean rebuild. If not, unpredictable references may result at runtime.

Major and minor code name space for performance trace

The major code numbers range from 0 to 2048. Performance trace major codes are assigned upwards from 0 to 255. Debug trace codes are assigned downward from 2048 to 1987. The code numbers between 256 and 1986 are reserved.

Although debug trace and conditional debug major codes are the same for any module or subsystem, turning on debug trace for a module does not turn on conditional debug for the same module and vice versa. The physical storage of conditional debug minor code flags is in a 32-bit word array separate from the major code numbers for trace.

Major and minor code declarations are in **mmCodes.h** and in **mmCodes.java.** The required #include and import statements are in the JPerf and CondDebug files in C and Java; explicit user actions are not required.

System resources used by traces

The performance and debug traces allocate a buffer in which to temporarily store trace data. The trace buffer size is configurable, but if you allocate too large a trace buffer, it can affect system performance.

Trace configuration parameters

You can configure:

```
Trace daemon IP address
Trace daemon port number
Trace on/off
Trace flags on (major,major,...)
Debug flags on ([major]min[major]min,min...)
Trace buffer size
Trace buffer management type (fill|wrap|ping-pong)
Debug trace filter thread list
```

Using debug tools in the boot cycle

How early in the boot cycle can these tools be used? It depends. None of the Java tools can be used until the Java Virtual Machine (JVM) and the JavaOS System Database have been initialized. As a result, no error log messages and no trace buffer can be sent to a daemon until the Java communications stack is initialized. Conditional debug messages in C can be made as soon as the serial console is initialized (mach_io.c), which is early in the boot sequence.

jperf commands

Use netShell commands to control the performance trace and the storage of information in the buffer.

jperftraceFlush Flushes the trace buffer at the end of a trace run.

jperftraceInit *size mode host port* Allocates a trace buffer where:

size	Specifies the size of the buffer in bytes.
mode	Specifies the mode of collection; normal, wrap, or continuous.
host	Specifies the host name on which the traceServer is to gather the data.
entry	Specifies the TCP port address used by the traceServer.

jperftraceOn Starts the trace.

jperftraceOff Turns off tracing, but keeps the buffer.

jperftraceReset Resets the buffer pointer to the beginning but keeps the buffer.

jperftraceFree Releases the trace buffer.

jperfai Reports CPU utilization at 1 second intervals.

jperfhk *command argument* Sets or disables trace hooks. See “Trace hooks” on page 156 for the trace hooks.

command Can be enable, disable, or query.

argument The major code in decimal.

jperftime For debug, is a test invocation of `Jperf.GetCycles( )`.

jperfq Displays the jperf buffer status.

jperfpsc *command a1 a2 a3* For debug, a test invocation of`JPerf.PerfSysCall(command,a1,a2,a3).`

jperfflagsOn *f1, f2,...* Turns on specified trace flags.

jperfflagsOff *f1, f2,...* Turns off specified trace flags.

jperfflagsOn all Turns on all trace flags.

jperfflagsOff all Turns off all trace flags.

jperfthOn *threadnum* Enables debug trace for a thread.

jperfthOn *threadname* Enables debug trace for a thread.

jperfthOff *threadnum* Disables debug trace for a thread.

jperfthOff *threadname* Disables debug trace for a thread.

jperfthOn all Enables debug trace for all threads.

jperfthOff all Disables debug trace for all threads.

Trace hooks

Each JavaOS for Business trace hook has an associated group code and each is identified by a major code and a minor code. When a specific major code is enabled, all hooks sharing that major code are also enabled. The list below identifies hooks that fire unconditionally if there is an active trace buffer. It also identifies hooks that communicate data.

The following list, by major code, does not show data items that contain no data.

Major code 0xa8

Minor code 0xf

1. Lost hook count

Indicates the number of hooks lost. Used for continuous recording; fires unconditionally.

Major code 0x4

Minor code level+0x1

1. Depth

Begins interrupt at level specified.

Major code 0x4

Minor code level+0x80000001

1. Depth

Ends interrupt at level specified.

Major code 0x10

Minor code 0x1c

1. Global system PC.
2. Method block address. The method block access flags are significant only for interpreted code.
3. Method base address. This is the base address of the first byte of code in the method.
4. Access flags: See tree.h for the key to access flags values.
5. java pc. This is usually the program counter in the interpreted method.

The tprof hook fires at each timer interrupt.

Major code 0x10

Minor code 0xaa

1. System tid
2. java tid
3. Threadname

Thread active at traceOn time. One hook for each thread found at traceOn.

Major code 0x10

Minor code 0xab

1. System tid
2. java tid
3. is System thread (Boolean)

Thread created while trace active.

Major code 0x10

Minor code 0xae

1. System tid
2. java tid
3. threadname.

Thread initialized. Follows the thread create hook 10/ab.

Major code 0x10

Minor code 0xac

1. System tid

Identifies the idle thread.

Major code 0x10

Minor code 0xad

1. System tid

Identifies the garbage collection thread.

Major code 0x10

Minor code 0xf

Indicates that the currently active thread has yielded. All threads yield prior to the dispatcher handing control to another thread; this may not be very interesting.

Major code 0x10

Minor code 0xc

1. System tid
2. Priority

Indicates ID of the thread whose priority is being changes. It also indicates the value of the new priority.

Major code 0x10

Minor code 0xc+0x80000000

1. System tid
2. Priority

Indicates the ID of the thread issuing a set priority call and its current priority. This always follows, after some delay, the 10/0xc hook.

Major code 0x11

Minor code 0xb0

1. Class block address
2. Class loader address
3. Number of methods
4. Classname

Identifies loaded classes at traceOn time. The class is either a ROM class or not. There is one record for each class encountered.

Major code 0x21

Minor code 0x01

1. Method block addres.
2. Class block address
3. Access flags:

   ```
   & 0x4000 (JIT)
   & 0x100 (native)
   ```

4. Method compiled code address (if JIT, see access flags)
5. Methodname (string)
6. Method signature (string)

Identifies methods associated with the class indicated in the prior 11/b0 hook. The class is not a ROM class.

Major code 0x22

Minor code 0x01

1. Method block address

2. Class block address
3. Access flags:

   ```
   && 0x4000 (JIT)
   && 0x100 (native)
   ```

4. Method compiled code address (if JIT, see access flags)
5. Methodname (string)
6. Method signature (string)

Identifies methods associated with the class indicated in the prior 11/b0 hook. The class is a ROM class.

Major code 0x11

Minor code 0xc0

1. Class block address
2. Class loader address
3. Number of methods
4. Class name

Identifies classes as they are loaded by the classloader into the binclasses structure. It is immediately followed by a series of 0x23/0x01 hooks, if enabled.

Major code 0x23

Minor code 0x01

1. Method block address
2. Class block address
3. Access flags:

   ```
   && 0x4000 (JIT)
   && 0x100 (native)
   ```

4. Method compiled code address (if JIT, see access flags)
5. Methodname (string)
6. Method signature (string)

Identifies methods associated with the newly loaded class indicated in the prior 11/c0 hook. The class is not a ROM class.

Major code 0x11

Minor code 0xc8

1. Class block address

Identifies classes as they are unloaded from the binclasses structure. There is one record for each class.

Major code 0x11

Minor code 0xb3

1. jit threshold

Identifies the value of jitThreshold at trace startup. Also, this is the final trace Startup hook to fire. This hook fires unconditionally.

Major code 0x12

Minor code 0x01

1. Current system tid
2. Next system tid

Indicates a thread switch (dispatch).

Major code 0x12

Minor code 0x03

1. Idle tid

Indicates a thread switch (dispatch) to the idle thread.

Major code 0x13

Minor code 0x01

1. Monitor ID
2. Nest count
3. Flags

Indicates successful acquisition of a monitor. This hook is not part of the retail build.

Major code 0x13

Minor code 0x02

1. Monitor ID
2. Monitor owner (system tid)

Indicates a blocked attempt to acquire a monitor already held by another thread. This hook is not part of the retail build.

Major code 0x13

Minor code 0x04

1. Monitor ID

Indicates that a monitor has been ititialized. This hook is not part of the retail build.

Major code 0x13

Minor code 0x03

1. Monitor ID

Indicates that a monitor has been released. This hook is not part of the retail build.

Major code 0x30

Minor code 0x10

1. Object ID
2. Method block

Method invocation. CLOOP=TRUE hook only.

Major code 0x30

Minor code 0x10+0x80000000

1. Object ID
2. Method block

Method exit. CLOOP=TRUE hook only.

Major code 0x32

Minor code 0x11

1. Object ID

Object free.

Major code 0x31

Minor code 0x12

1. Object ID
2. Class block pointer
3. Length

Class object allocate.

Major code 0x31

Minor code 0x13

1. Object ID
2. Array type:

00	Unassigned
01	Array
02	Class
03	Unassigned
04	Boolean (1–byte)
05	char
06	Float, single
07	float, double
08	integer byte signed
09	integer, short signed
0a	integer, int signed
0b	integer, long signed
0c	integer, byte unsigned
0d	integer, short unsigned
0e	integer, int unsigned
0f	integer, long unsigned

Array object allocate.

Major code 0x31

Minor code 0x14

1. Object ID
2. Class block pointer
3. Length

Normal object allocate.

If cmd `=query`, then enabled trace hooks (the trace major codes) are identified on the console. If cmd `=enable`, then arg1 should identify, in decimal, a trace major you wish to enable. If cmd `=disable`, then arg1 should identify, in decimal, a trace major code you wish to disable.

Chapter 34. Using the ICAT kernel debugger

The ICAT kernel debugger is documented in *ICAT Debugger JavaOS for Business - OS/2 Warp 4*.

Chapter 35. Using the conditional debug (console output) messages

The conditional debug messages can let you know what is happening in the operating system code. When activated, messages are displayed at the serial console output device indicating activity at a network computer.

Conditional debug commands

Use netShell commands to control the debug trace and the storage of information in the buffer.

cdebug *?* Displays a help text.

cdebug flagsOn [*maj***]***min* Turns on a single debug flag.

cdebug flagsOn [*maj***]***min,min,min* Turns on more than one debug flag.

cdebug flagsOn [*maj***]***min* **[***maj***]***min* Turns on more than one debug flag.

cdebug flagsOn [*maj***] all** Turns on all the debug flags in the group.

cdebug flagsOn all Turns on all the debug flags.

cdebug flagsOff [*maj***]***min* Turns off a debug flag.

cdebug flagsOff [*maj***]***min,min,min* Turns off some debug flags.

cdebug flagsOff [*maj***] all** Turns off all the debug flags in the group.

cdebug flagsOff all Turns off all the debug flags.

cdebug info Turns off all the debug flags.

cdebug jvars [*var***]** *val* Sets a jVar.

cdebug jvars [*var***]** *val* **[***var2***]** *val2* Sets several jVars.

cdebug jvars [all] *val* Sets all jVars to the same value.

Major and minor code name space for conditional debug

The major code numbers range from 0 to 2048. Debug trace codes are assigned downward from 2048 to 1987. Performance trace major codes are assigned upwards from 0 to 256. The code numbers between 256 and 1987 are reserved.

Although debug trace and conditional debug major codes are the same for any module or subsystem, turning on debug trace for a module does not turn on conditional debug for the same module and vice versa. The physical storage of conditional debug minor code flags is in a 32-bit word array separate from the major code numbers for trace.

Major and minor code declarations are in **mmCodes.h** and in **mmCodes.java.** The required #include and import statements are in the JPerf and CondDebug files in C and Java; explicit user actions are not required.

Chapter 36. The JCT tool log

The JCT can be configured to log runtime exceptions. If the jct.properties file contains a jct.log.dir property, logs are stored in that directory. If no properties have been defined for a log file, no logging is done.

Part 6. Appendixes

Appendix A. Tested Network Computer Hardware Configurations

The following section lists the hardware configurations and adapters known to work with this release of JavaOS for Business. This is not intended as an exhaustive list of all acceptable configurations.

Basic network computer configuration

All platforms tested were equipped with a standard US keyboard, a PS/2 mouse, and a UART 16550 serial port. Firmware levels of 3/97 and 4/98 worked consistently, but earlier versions did not. Configurations with 32 MB of RAM were also tested, but better video and overall performance was achieved with 64 MB or higher. A summary of the processors tested is found in Table 1.

Table 1. Processor and RAM information for tested configurations.

Processor Type	**Processor Speed (MHz)**	**RAM (MB)**
Pentium	166	64
Pentium	166	128
Pentium Pro	180	128
Pentium Pro	200	64
Pentium Pro	200	128
Pentium II	233	64

Video configurations

The network computers tested used one of the following adapters:

- Matrox Millennium MGA (4 MB)
- S3 Trio 64V+ (2 MB)
- S3 Trio 64V2/DX (2 MB)

Audio configurations

The network computers tested support the Soundblaster 16 series of audio adapters. The following models were specifically tested:

- CT1740
- CT1750
- CT1770

The following SoundBlaster 16 models should also work, though they were not explicitly tested:

- CT1730
- CT1759
- CT1779
- CT2230

- CT2230C
- CT2230S
- CT2239
- CT2239C
- CT2239S
- CT2290
- CT2299
- CT2291
- CT2740
- CT2749
- CT2750
- CT2759
- CT2770
- CT2830
- CT2839
- CT2910
- CT2911
- CT2919
- CT2260
- CT2800
- CT2810
- CT2840
- CT2860

JavaOS for Business does not support any Plug and Play versions of SoundBlaster 16 adapters, original 8-bit SoundBlaster or SoundBlaster Pro adapters, or the AWE 32 or AWE 64 adapters.

Network adapters

The following network adapters have been tested:

- IBM Etherjet with 82558 (PXE capable)
- Intel® EtherExpress™ Pro 100B with 82558 (PXE capable)

Printers tested

The following printers have been tested using the IEEE-standard SPP mode parallel printer port on the network computers:

- IBM LaserPrinter 4029-040 10L
- IBM LaserPrinter 4019-001
- IBM 4317
- Lexmark Optra E+
- Lexmark Optra S2450
- Lexmark 4049-12R LaserPrinter OptraR
- Lexmark LaserPrinter 4039-16L
- Hewlett Packard LaserJet 5L
- Hewlett Packard Color LaserJet 1200C

Appendix B. Uninstalling the IBM Intermediate Support Driver

After the IBM Intermediate Support Driver is installed, the properties notebook for the TCP/IP protocol is not accessible. If you need access to this notebook, you must uninstall the driver, access the notebook, and then reinstall the driver as previously outlined. To uninstall the IBM Intermediate Support Driver, do the following:

1. Select **Start**, **Programs**, and then **Command Prompt**.
2. Change to the temporary directory containing the extracted files in step 4 on page 16 of installing the IBM Intermediate Support Driver.

   ```
   d:
   cd \temp\wedge
   ```

3. Run the **wejcfgex** command twice, once with the **-disable** option and once with the **-remove** option.

   ```
   wejcfgex -disable
   wejcfgex -remove
   ```

4. Select **Start**, **Settings**, **Control Panel**, **Network**, and then **Protocols**.
5. Select **IBM Intermediate Support Driver** and select **Remove**. The network properties notebook is now accessible.

Index

A

adding a setting 102
administration 93
application
  load settings 103
  main or startup 71
artwork, replacing 65
authentication
  setting up local 60
  setting up remote 62

B

base smart card setting 104
bldlevel NetShell command 143
boot time device drivers 73
BOOTP (Bootstrap Protocol) 8
bootup
  network computer 148
  problems 147
  process start 147
  successful 147

C

cdebug, netShell command 165
changing a priority 103
changing a setting 102
command
  netShell 123
command line 121
conditional debug
  major and minor codes 165
conditional debug settings 104
conditional debug, NetShell command 165
configuration planning 55
configuring
  keyboard language 75
  multimedia 85
  printers 87
  sound card 85
  time zone 76
  video 83
creating or adding a font 78
customizing
  artwork 65
  login processing 59
  smart card 59

D

daily operations 93, 101
debug trace 153
debug trace formatter 154
device driver
  adding 73
  boot time 73
DHCP (Dynamic Host Configuration Protocol) 8
disable error logging 149
drivers
  specifying keyboard and mouse 81
Dynamic Host Configuration Protocol (DHCP) 8

E

edit panels
  settings 101
enable error logging 149
error logger settings 104
error logging
  enable or disable 149
error logs 149

F

firmware problem 141
fonts
  creating or adding 78
  included 77
  managing 77
French considerations for the HotJava browser 67

G

German considerations for the HotJava browser 67
getting started 95
groups
  JavaOS Configuration Tool 95
groups, planning for 98
guest ID 63

H

hardware problem 141
help information 135
help options
  HotJava browser 67
HotJava browser
  changing bug submission address 69
  changing comments address 68
  changing help options 67
  configuration settings 104

HotJava browser *(continued)*
deleting options 69
load settings 104
non-US English server 67
HTTP 9
Hypertext Transfer Protocol (HTTP) 9

I

iiimp settings 105
Intel EtherExpress Pro 100 setting 105
Italian considerations for the HotJava browser 67

J

JavaOS Configuration Tool 95
main panel 95
terminology 95
using 101
JavaOS for Business
properties 119
JavaOS System Database 119
JCT, using 101
JIT setting 105
jperf, netShell command 155
just in time compiler setting 105

K

keyboard
driver setting 106
format settings 106
language 75
mapping settings 106
specifying a driver 81
keyboard and mouse base settings 105

L

language
customization 75
setting keyboard 106
language format settings 106
Line Printer Daemon (LPD) 8
local authentication 60
locale format settings 106
locale settings 75
logger, netShell command 151
login customizing 59
login settings 107
logs, error 149
LPD (Line Printer Daemon) 8

M

MAC address 7
machine information 147
machine settings sequence 100
machines
JavaOS Configuration Tool 96
main application 71
main application settings 103
major code name space
conditional debug 154, 165
management
users 115
mapping
keyboard 106
master configuration file 117
Media Access Control (MAC) address 7
messages
server 127
user 135
messages, general description 125
minor code name space
conditional debug 154, 165
mouse
specifying a driver 81
mouse driver setting 107
multimedia
configuring 85

N

NC PCI token ring driver setting 107
netShell 123
NetShell commands
bldlevel 143
cdebug 165
jperf 155
logger 151
sysver 143
network computer
bootup 148
Network File System (NFS) 8
network file system settings 108
new installation 95
NFS (Network File System) 8
NFS AutoMounter settings 108
NFS settings 108
NIS printer setting 108
NS1000 smart card driver 108
NSL authenticator login settings 108

O

OpenCard settings 109
operations 93, 101
overriding a setting 103

P

parallel port driver setting 109
parallel printer
  configuring 87
parallel printer setting 109
PCI token ring driver setting 110
performance trace
  major and minor codes 154
platform settings sequence 100
platforms
  JavaOS Configuration Tool 97
PowerPC audio driver settings 110
preparing for users 115
printers
  configuring 87
priority, changing 103
problem
  at bootup 147
  determining the source 139
  reporting 145
problem discovery 140
problem isolation 141
profiles
  JavaOS Configuration Tool 96
properties 119
proxy settings 110

R

reading from a smart card 62
region format settings 107
remote authentication 62
remote printer setting 111
removing a setting 102

S

S3 Trio 64 driver setting 111
serial port driver setting 111
serial printer
  configuring 87
serial printer setting 111
server
  messages 127
setting, adding 102
setting, changing 102
setting, removing 102
settings 101
  application load 103
  base smart card 104
  conditional debug 104
  error logger 104
  HotJava browser 104
  HotJava browser configuration 104
  iiimp 105
  Intel EtherExpress Pro 100 105
settings *(continued)*
  JIT 105
  just in time compiler 105
  keyboard and mouse 105
  keyboard driver 106
  keyboard format 106
  language format 106
  locale format 106
  login 107
  mouse driver 107
  NC PCI token ring driver 107
  NC token ring driver 107
  network file system 108
  NFS AutoMounter 108
  NIS printers 108
  NS1000 for smart card 108
  NSL authenticator login 108
  OpenCard 109
  parallel port driver 109
  parallel printer 109
  PCI token ring driver 110
  PowerPC audio driver 110
  proxy 110
  region format 107
  remote printer 111
  S3 Trio 64 driver 111
  serial port driver 111
  serial printer 111
  smart card 104
  smart card authenticator 112
  SNMP manager address 112
  SOCKS 110
  sound blaster 16 audio driver 112
  SVGA driver 113
  system database location 113
  system image location 113
  TFTP AutoMounter 113
  time zone 113
  token ring driver 110
  trace tool 114
  video driver 114
smart card
  customizing 59
  reading from 62
  storing data on 62
smart card authenticator format settings 112
smart card NS1000 setting 108
smart card setting 104
SNMP manager address setting 112
SOCKS settings 110
sound blaster 16 audio driver 112
sound card
  configuring 85
Spanish considerations for the HotJava browser 67
startup application 71
  user-supplied 71

storing data on a smart card 62
supplier
  reporting problems to 145
SVGA driver setting 113
system administration 93
system database location 113
system image location settings 113
system maintenance 117
  JavaOS Configuration Tool 98
system settings
  error logging 117
  jar directory 118
  JavaOS System Database server 118
  save all settings 119
sysver 143
sysver NetShell command 143

T

tasks
  user 115
TFTP (Trivial File Transfer Protocol) 8
TFTP AutoMounter setting 113
TFTP message 147
time zone
  selecting 76
  settings 113
token ring driver setting 110
token ring driver setting for NC 107
tools
  JavaOS for Business 140
tools, problem solving
  Logs 149
trace tool settings 114
trace, debug 153
Trivial File Transfer Protocol (TFTP) 8
troubleshooting 139
  process 139

U

user
  messages 135
user ID
  not required 63
user inactivity warning time 64
user information 147
user management 115
user problems 115
user tasks 115
users
  JavaOS Configuration Tool 95
users, planning for 98
using the JavaOS Configuration Tool 101

V

video driver
  configuring 83
  settings 114
Video S3 Trio 64 driver setting 111

W

warning time, user inactivity 64

Printed in U.S.A.