



Installation Guide

Network User Interface



78-0534C
Printed in the U.S.A.
August 2001

This document contains proprietary information that is protected by copyright. No part of this document may be reproduced, translated, or modified without the prior written consent of QAD Inc. The information contained in this document is subject to change without notice.

QAD Inc. provides this material as is and makes no warranty of any kind, expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. QAD Inc. shall not be liable for errors contained herein or for incidental or consequential damages (including lost profits) in connection with the furnishing, performance, or use of this material whether based on warranty, contract, or other legal theory.

MFG/PRO is a registered trademark of QAD Inc. QAD, QAD eQ, and the QAD logo are trademarks of QAD Inc.

Designations used by other companies to distinguish their products are often claimed as trademarks. In this document, the product names appear in initial capital or all capital letters. Contact the appropriate companies for more information regarding trademarks and registration.

Copyright © 2001 by QAD Inc.
78-0534C

QAD Inc.
6450 Via Real
Carpinteria, California 93013
Phone (805) 684-6614
Fax (805) 684-1890
<http://www.qad.com>

Contents

ABOUT THIS GUIDE	1
What Is in This Guide?	2
Audience	2
Installation Documentation Updates	3
QAD Web Site	3
Related Documentation	4
Progress Documentation	4
MFG/PRO eB Documentation	4
Document Conventions	5
QAD's Global Support Services	6
 SECTION 1 NETUI IMPLEMENTATION.....	7
 CHAPTER 1 SYSTEM OVERVIEW	9
Implementation Overview	10
NetUI Service Packs and JDK Versions	10
QAD eB Desktop	11
NetUI Components	12
Prerequisite Components	13
Third-Party Components	13
QAD-Supplied Components	15
NetUI Architecture	17
Overview	17

Client Components	19
Maintenance and Transaction Components	20
Menu, Navigation, Browse, Inquiry, and Report Components	26
Progress Server Architecture	27
Administrative Framework	28
WebSpeed Transaction Server	29
AppServer	30
Data Security	31

CHAPTER 2 SYSTEM REQUIREMENTS 33

Skills Required for Implementation	34
Prerequisite Requirements	34
Client Requirements	35
Operating Systems	35
Client Software	35
Minimum Client Hardware	35
Web Server Requirements	36
Operating Systems	36
Software	36
Hardware	36
Progress Server Requirements	37
Operating Systems	37
Hardware	38
Servers	38
RMI Server Requirements	38
Java Requirements	39

CHAPTER 3 PLANNING 41

Deployment	42
Deployment Options	42
Deployment Issues	44
Installation Summary	45
Install MFG/PRO eB	46
Install the NetUI Web Server	47

Install the Java Development Kit (JDK)	47
Create a NetUI Parameter File	49
Set Up NetUI Progress Servers	49
Install NetUI Software	51
Configure the Telnet Server	52
Configure the NetUI Environment	53
Run the NetUI Client Setup	53
CHAPTER 4 SETTING UP THE NETUI ENVIRONMENT	55
Installing the NetUI Web Server	56
Install the Web Server	56
Set Up the MIME-Type	56
Verify the Web Server	57
Installing the Java Development Kit	57
Creating a NetUI Parameter File	58
Connection Parameters	59
Create the Parameter File	60
Setting Up the NetUI Progress Servers	63
Setup Prerequisites	64
Setup Environments	64
Set Up on Windows NT	65
Set Up on UNIX	83
CHAPTER 5 INSTALLING THE NETUI SOFTWARE ON UNIX	97
Installation Overview	98
Installing the Language Files	98
Mount the CD-ROM	98
Run the Installation Script	99
Installing the Server Files	101
Mount the CD-ROM	101
Run the Installation Script	101
Installing the NetUI Service Pack	107
Items Installed	107

CHAPTER 6 INSTALLING THE NETUI SOFTWARE ON WINDOWS NT 109

Installation Overview	110
Installing the Language Files	110
Installing the Server Files	113
Installing the NetUI Service Pack	117
Items Installed	117

CHAPTER 7 CONFIGURING TELNET ON UNIX 119

Configuration Overview	120
Configuring the Telnet Environment	120
Restricted Shells	121
Examples of Security Measures	121
Configuring Telnet Options in MFG/PRO	123
Configure Telnet Server Settings	124
Configure Telnet Log-In Sequence	125
Configure Telnet Connection Settings	129

CHAPTER 8 CONFIGURING TELNET ON WINDOWS NT 131

Configuration Overview	132
Download Updated Software	132
Prerequisite Components	132
Installing the Telnet Server	133
Install the Software	133
Register the Software	134
Configuring the Telnet Environment	135
Create a Telnet Directory	135
Create a General Startup File	136
Create User-Specific Startup Files	138
Set Up Local User Accounts	140
Configuring Telnet Options in MFG/PRO	141
Configure Telnet Server Settings	141
Configure the Telnet Log-In Sequence	143
Configure Telnet Connection Settings	146

CHAPTER 9 CONFIGURING THE NETUI ENVIRONMENT 149

Configuration Summary	150
Setting Up the RMI Server	150
RMI Directory Structure	153
Creating the NetUI Help Script	154
Configuring NetUI Constants	155
Configure NetUI Service Constants	156
Map Language Codes	160
Map Code Pages	166
Configure the Auxiliary AppServer	166
Installing Client Files	168
Run Client Setup before Service Pack 2	168
Run Client Setup after Service Pack 2	172
Complete NetUI Setup	176

SECTION 2 REFERENCE 179**CHAPTER 10 CONFIGURING OPTIONAL USER SETTINGS. 181**

Configuration Overview	182
Defining User Settings	183
Defining NetUI URLs	187
Define Menu-Level URL Links	188
Define URL Links for Browsers	189
Monitoring User Sessions	192
Other NetUI Configuration Options	193
Adding New Folders to the Main Navigation Tree	193
Assigning Programs to the User Menu	194
Assigning Buttons to the Tool Bar	194

CHAPTER 11 SYSTEM ADMINISTRATION. 197

Running the NetUI Using a Proxy Server	198
Application Modes	199

NetUI HTML Files	201
Default HTML Files	201
HTML File Parameters	204
WebSpeed Agent Sizing and Behavior	206
Agent Usage for Reports and Inquiries	207
Agent Usage for Browsers and Look-Ups	207
Mechanisms to Optimize Agent Usage	208
Securing Internet Communication	209
Secure Sockets Layer	209
SSL Security in the NetUI	210
SSL Setup Overview	211
Set Up SSL	212
Verify SSL Security	214
Common System Administration Tasks	215
RMI Server Administration	215
NetUI Output Administration	216
Memory Management	216
MFG/PRO Database Set Administration	217
CHAPTER 12 ENVIRONMENT TABLES	219
Prerequisite Component Information	220
NetUI Setup Information	222
NetUI Software Installation Information	224
NetUI Configuration Information	225
GLOSSARY	227
INDEX	235

About This Guide

What Is in This Guide? 2

QAD Web Site 3

Related Documentation 4

Document Conventions 5

QAD's Global Support Services 6



What Is in This Guide?

Use this guide to install and configure QAD's Network User Interface (NetUI). The guide is divided into two sections: NetUI implementation and reference.

The NetUI implementation section contains instructions on how to:

- Set up a Web server for use with the NetUI.
- Install the Java Development Kit for use with the NetUI.
- Set up the Progress WebSpeed and Open AppServer products for use with the NetUI.
- Install the QAD NetUI software.
- Configure a telnet server for use with the NetUI.
- Configure the NetUI environment, define NetUI constants, and run the NetUI client setup.

The reference section contains instructions on performing optional user and system administration activities. It also contains tables that you can use to document your NetUI environment.

If you are using the optional eB Desktop for the NetUI, you received a separate installation guide for this product. Use it to install the eB Desktop after you have completed the NetUI installation.

Audience

These instructions are intended for a system administrator with between three and five years experience configuring and managing hardware and operating system software. Additionally, the administrator should have a good understanding of networking concepts and administration.

Implementing the NetUI requires the installation and configuration of third-party software such as the Java Development Kit, the Progress WebSpeed and AppServer products, and the Web server. These instructions require some installation and configuration experience to implement these components.

Installation Documentation Updates

Installation Errata

In addition to these instructions, you may receive a supplementary errata sheet with changes and additional instructions. Check your product package.

Even when an errata sheet is included with your product package, QAD recommends that you download the most up-to-date errata sheet from the QAD Web site. New information may have been added to the errata sheet since your product was shipped.

Installation Guides

To ensure a successful implementation, QAD installation guides are periodically updated. To determine whether your installation guide has been updated, refer to the QAD Web site. Compare the item number listed on your installation guide with the number listed on the QAD Web site. If your installation guide has been updated, download and use the most recent version.

QAD Web Site

QAD's Web site provides a wide variety of information about the company and its products. You can access the Web site at:

<http://www.qad.com>

For QAD customers with a Web account, MFG/PRO documentation is available for review or downloading at:

<http://support.qad.com/documentation/>

To obtain a QAD Web account, go to:

<http://support.qad.com/>

Most user documentation is available in two formats:

- Portable document format (PDF). PDF files can be downloaded from the QAD Web site to your computer. You can view them with the free Adobe Acrobat Reader. A link for downloading this program is also available on the QAD Web site.
- HTML. You can view user documentation through your Web browser. The documents include search tools for easily locating topics of interest.

Features also include an online solution database to help users answer questions about setting up and using QAD products. Additionally, the QAD Web site has information about training classes and other services.

Related Documentation

Progress Documentation

- For information on installing and configuring WebSpeed, see the *WebSpeed Installation and Configuration Guide*.
- For information on installing and configuring AppServer, see *Building Distributed Applications Using the Progress AppServer*.

MFG/PRO eB Documentation

- For an overview of new features and software updates, see the *Release Bulletin*.
- For information on installing or converting to MFG/PRO eB, refer to the installation guide for your system.
- For technical details, see the *File Relationships* and *Database Definitions*.
- For information on using MFG/PRO eB, refer to the *User Guides*.
- For instructions on navigating the MFG/PRO eB Windows and character environments, refer to *User Guide Volume 1: Introduction*. Navigation information for the NetUI is provided in *User Guide: eB Desktop and Network User Interface*.

- For information on using features that let MFG/PRO eB work with external applications, see the *External Interface Guides*. For example, these guides describe the Warehousing application program interface (API) and Q/LinQ, the tool set for building and using tools that perform complex data exchange between MFG/PRO eB and external systems.
- To view documents online in PDF format, see the *Documents on CD* and *Supplemental Documents on CD*. The CD-ROM media includes complete instructions for loading the documents on a Windows network server and making them accessible to client computers.

Note MFG/PRO eB installation guides are not included on a CD. Printed copies are packaged with your software. Electronic copies of the latest versions are available on the QAD Web site.

Document Conventions

This guide uses the typographic conventions listed in the following table.

If you see:	It means:
monospaced text	A command or file name.
<i>italicized monospaced text</i>	A variable name for a value you enter as part of an operating system command; for example, <i>YourCDROMDir</i> .
indented command line	A long command that you enter as one line, although it appears in the text as two lines.
Note	Alerts the reader to exceptions or special conditions.
Important	Alerts the reader to critical information.
Warning	Used in situations where you can overwrite or corrupt data, unless you follow the instructions.

MFG/PRO eB is available in several interfaces: Windows, character, Web browser, and an interface for object-oriented programs. To standardize presentation, this guide uses the following conventions:

- In Chapter 9 and Chapter 10, screen captures show the Windows interface.
- References to keyboard commands are generic. For example, choose Go refers to F2 in the Windows interface and to F1 in the character interface. When you use the product, the Progress status line at the bottom of the program window lists the main UI-specific keyboard commands available for that program.

For complete keyboard command summaries for each MFG/PRO eB interface, refer to the appropriate chapters of *User Guide Volume 1: Introduction* and *User Guide: eB Desktop and Network User Interface*.

QAD's Global Support Services

QAD's NetUI has a wide variety of configuration possibilities, is highly scalable, and easily customized. While this guide provides basic installation and configuration information, it cannot consider all of the possible computing environments and variations into which the NetUI can be implemented.

To take full advantage of the NetUI's flexibility and potential in your specific environment, contact your QAD Support representative for information on the installation and customization offerings supplied by QAD's Global Support Services. These offerings include performance enhancements as well as technical and administration training.

SECTION 1

NetUI Implementation

This section provides instructions for implementing the NetUI.

System Overview **9**

System Requirements **33**

Planning **41**

Setting Up the NetUI Environment **55**

Installing the NetUI Software on UNIX **97**

Installing the NetUI Software on Windows NT **109**

Configuring Telnet on UNIX **119**

Configuring Telnet on Windows NT **131**

Configuring the NetUI Environment **149**

CHAPTER 1

System Overview

This chapter provides the background information required to implement the NetUI. You should read this chapter *before* you begin any phase of implementation.

Implementation Overview **10**

NetUI Components **12**

NetUI Architecture **17**

Progress Server Architecture **27**

Data Security **31**

Routing Maintenance (Main Screen)

Routing Code: 10-15000 OPERATION: 20 NONAD(75) CODE: 00

Standard Operation: 10:30 INSPECTION: ALL SITE

Description: INSPEC PER PROX-000

Machines per Op: 1 Milestone: 1

Overlap Units: 1

Queue Time: 1.0

Wait Time: 0.0

Setup Time: 0.0

Route by Production Route

Implementation Overview

This guide provides instructions on performing the installation and configuration procedures required to implement the NetUI in an MFG/PRO eB environment.

The overall NetUI implementation process can be broken down into the following major stages:

- Planning. During this stage, you plan the NetUI deployment and installation activities and obtain information about your environment.
- Setup. During this stage, you set up your environment for the NetUI.
- Installation. During this stage, you install the QAD NetUI software.
- Configuration. During this stage, you configure components in the NetUI system to work together.

Installation planning and installation steps for the NetUI are affected by two considerations: the service-pack level you are installing and whether you are planning to use the optional eB Desktop product that works with the NetUI.

NetUI Service Packs and JDK Versions

The NetUI as it was released with MFG/PRO eB in August 2000 has the following Java requirements:

- Sun Microsystems Java Development Kit (JDK) 1.1.8 on the Web server and remote method invocation (RMI) server
- Java Plug-In version 1.1.3 on each client

Beginning with NetUI Service Pack 2, these requirements have changed to take advantage of improvements offered by later versions of the JDK. With Service Pack 2, NetUI requires:

- Sun Microsystems JDK 1.3.0_02 or higher on the Web server and RMI server
- Java Runtime Environment 1.3.0_02 or higher, which includes the Java Plug-In, on each client

Note NetUI with Service Pack 2 has the same Java requirements as the optional eB Desktop product.

Changes in the version of Java also affect the security algorithm used on the client and the client install.

With the more recent version of Java, Service Pack 2 supports users of Netscape Communicator version 6.

This installation guide can be used to install the NetUI as it was originally released, the NetUI with Service Pack 1, and the NetUI with Service Pack 2 or higher. Notes about differences in the installation process are included when necessary.

QAD eB Desktop

QAD provides an additional optional component that replaces the standard NetUI desktop. This product—eB Desktop—is purchased separately and shipped on its own CD with its own installation manual.

QAD eB Desktop adds an HTML-based front end to the NetUI using an expanded version of the NetUI's embedded mode. All MFG/PRO eB programs can be accessed through an easy-to-use HTML navigation system. Full search capability is provided for menu numbers, menu descriptions, and program names.

If you plan to use eB Desktop, how you install the NetUI is affected. If you are using eB Desktop with NetUI and Service Pack 1:

- Do not perform the NetUI client installation, but follow the instructions in *Installation Guide: eB Desktop* for client requirements and installation steps.
- Follow the instructions in *Installation Guide: eB Desktop* for installing updated help scripts using JDK 1.3.0_0.2 or higher.
- Follow the instructions in *Installation Guide: eB Desktop* for setting up the RMI server using JDK 1.3.0_0.2 or higher.

If you are using eB Desktop with NetUI and Service Pack 2, some of these steps are different.

- Do not perform the NetUI client installation, but follow the instructions in *Installation Guide: eB Desktop* for client requirements and installation steps.

- Skip the instructions in *Installation Guide: eB Desktop* for installing updated help scripts using JDK 1.3.0_0.2 or higher. Your version of the NetUI with Service Pack 2 will already be using JDK 1.3.0_0.2.
- Skip the instructions in *Installation Guide: eB Desktop* for setting up the RMI server using JDK 1.3.0_0.2 or higher since you will already have updated it during the service-pack install.

NetUI Components

The NetUI relies on the interaction of several different software components. There are three basic types of components:

- Prerequisite components. These components must exist in your environment before you begin implementing the NetUI. This guide assumes their existence and does not provide installation and configuration instructions for them.
- Third-party components. These components are supplied by software vendors other than QAD. Vendors can include Progress Software Corporation, Netscape Communications Corporation, Microsoft Corporation, and Sun Microsystems, Inc. This guide provides information about installing and configuring these components for use in a NetUI environment.
- QAD supplied components. These components are supplied by QAD on the NetUI media. This guide provides installation and configuration information for these components.

The following sections provide brief descriptions of the NetUI components. These descriptions are intended to provide a framework for understanding the NetUI architecture before you begin your implementation. Specific details on the different aspects of the NetUI components can be found in the appropriate chapters of this guide.

For example, for information on the system requirements for a specific component—such as version, disk space, and memory—see Chapter 2, “System Requirements,” on page 33.

Prerequisite Components

Before you begin implementing the NetUI, make sure you have the following prerequisite components. This guide assumes that these components are installed and configured.

- The required Progress software components for MFG/PRO
- MFG/PRO eB database server
- MFG/PRO eB character client code
- Internet access

◆ For more information, see the MFG/PRO installation guide for your system.

Third-Party Components

These components are supplied by vendors other than QAD. This guide includes instructions for installing and configuring these components for use in a NetUI environment.

Progress Server Components

In addition to the Progress software required for MFG/PRO, the NetUI uses the following Progress server products.

- NameServer
- AppServer
- WebSpeed

Java Component

The NetUI uses the Java Development Kit (JDK) to enable the remote method invocation (RMI) server and the NetUI online help. On Windows NT systems, the JDK may have been installed when you installed Progress, depending on which Progress components you selected. If you are using a UNIX server, you must download the JDK from the Sun Microsystems Web site.

The version of the JDK required depends on the service pack level you are installing.

◆ See “NetUI Service Packs and JDK Versions” on page 10.

Web Server Component

The NetUI uses a Web server to enable user access to the NetUI. You must obtain Web server software from the vendor of your choice. The NetUI can operate with any common gateway interface (CGI) compliant Web server. Examples include Netscape Enterprise Server and Microsoft Internet Information Server (IIS).

Web Browser Component

QAD's NetUI is a browser-based interface to MFG/PRO. Each client machine must have a browser to access the NetUI. QAD supports the NetUI on Microsoft Internet Explorer and Netscape Communicator.

Important If you plan to use the eB Desktop, you must use Internet Explorer 5.5 or above; Netscape Communicator is not supported.

You must use a supported browser and version. Otherwise, there are no special installation or configuration issues involving the Web browser. Therefore, this guide does not include instructions for installing and configuring the browser software. QAD suggests you follow the browser vendor's instructions to install and test your Web browser.

Telnet Server Component

The NetUI uses a telnet server to enable the telnet interface for MFG/PRO maintenance and transaction programs. Depending on the operating system of the machine on which your telnet server will run, you may need to install additional software.

- If you plan to use a UNIX machine for the telnet server, you can use the default telnet service provided with the operating system.
- If you plan to use a Windows NT machine for the telnet server, you must install the Georgia SoftWorks telnet server software. This software is supplied by QAD.

▶ See "Windows NT Telnet Server Software" on page 17.

QAD-Supplied Components

The following components are supplied by QAD and are included on the NetUI media.

QAD Security Files

The NetUI Java code operates in conjunction with security files that authenticate the code. The specific files involved vary depending on the version of Java being used.

- Before Service Pack 2, a file called `identitydb.obj` is loaded on each client machine by the NetUI client setup process.
- Beginning with Service Pack 2, a `qadkeystore` file is copied to each client. In addition, the `java.policy` file is modified so that users can run `.jar` files signed by QAD.

▶ See “NetUI Service Packs and JDK Versions” on page 10.

Java Virtual Machine

The Java Virtual Machine (JVM) is responsible for executing the platform-independent Java bytecode on a specific operating system. Some JVMs contain just-in-time (JIT) compilers that increase Java performance.

The most widely used Web browsers include JVMs. However, the NetUI uses a more recent version of the JVM than the browser default. To enable this JVM, the NetUI client setup process installs a Sun-developed JVM called the Java Plug-In.

Note Beginning with Service Pack 2, the client setup process installs the Java Runtime Environment (JRE) 1.3.0_0.2, which includes the Java Plug-In.

Using Sun’s JVM has the following advantages:

- It is a known, stable JVM that follows all language standards.
- It works with most Web browsers.
- It can be installed automatically on Windows clients.
- It bypasses the Web browser’s existing JVM only when explicitly required by the NetUI HTML files.

- It supports new versions of the Java language earlier than the browser JVMs.

The Java Plug-In requires less than 10 MB of disk space. The JRE requires a little more than 20 MB of disk space.

NetUI Java Code

The NetUI Java code is contained in Java archive files (extension .jar) on the NetUI media. During the NetUI media installation, these files are copied to the Web server's document directory.

Table 1.1 lists the NetUI Java archive files and provides a brief description of the Java programs (classes) in each.

Table 1.1
NetUI Java Archive
Files

File	Description
multinet.jar	Contains the NetUI Java classes
o4glrt.jar	Contains the Progress open client classes

How the .jar files are managed on the client depends on the version of Java you are using.

Java 1.1.8

With Java 1.1.8, the NetUI Java files are downloaded to each client machine during the NetUI client setup. After client setup, the NetUI Java files are located in the `applet` subdirectory below the `lib` directory in the Java Plug-In installation directory.

Java 1.3

With Java 1.3, .jar files are managed by the JRE and are automatically downloaded when the version on the client is older than the version on the server. The client setup does not download the files, since they are automatically downloaded the first time a client needs them. Archive files are located in the user's Windows profile directory in a subdirectory called `java_plugin_AppletStore`.

NetUI HTML Files

The NetUI HTML files provide the access point for client Web browsers when they first launch the NetUI. These files can be customized for different business purposes.

▶ See “NetUI HTML Files” on page 201.

Windows NT Telnet Server Software

The NetUI installation media contains Georgia SoftWorks Windows NT Telnet Server software. The Windows NT telnet server enables NetUI client access to MFG/PRO telnet maintenance and transaction programs on a Windows NT server. The Georgia SoftWorks server is fully integrated with MFG/PRO telnet session functionality and works seamlessly in the NetUI environment. The server lets you monitor client sessions as well as detect and eliminate abnormally terminated client sessions.

For more information, access the Georgia SoftWorks Web site for a copy of their *Windows NT Telnet Server User Guide*.

<http://www.georgiasoftware.com>

NetUI Architecture

Overview

The NetUI architecture diagram (Figure 1.1) illustrates the components that make up the NetUI and how they interact.

Fig. 1.1
NetUI Architecture

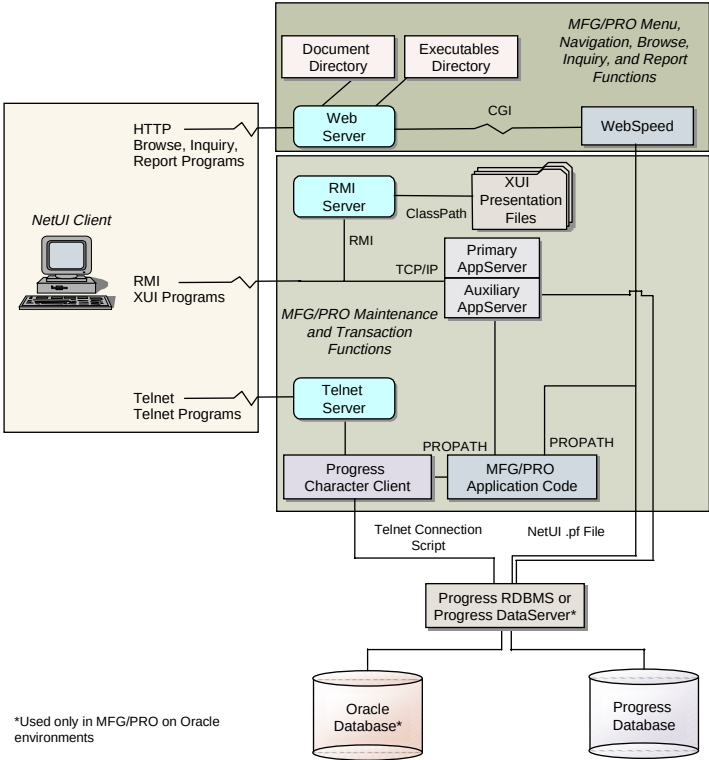


Figure 1.1 uses the acronyms and abbreviations in Table 1.2.

Table 1.2
NetUI Acronyms
and Abbreviations

Acronym or Abbreviation	Definition
RMI	Remote Method Invocation
TCP/IP	Transmission Control Protocol/Internet Protocol
XUI	The NetUI's extended user interface (XUI)
XML	Extensible Markup Language
CGI	Common Gateway Interface
HTTP	Hypertext Transfer Protocol
RDBMS	Relational Database Management System

See page 227.

See the glossary for an explanation of these terms.

The following sections describe how the NetUI components interact to perform different MFG/PRO functions. Using the NetUI architecture

diagram as a guide, the components in each of the logical areas are discussed.

Client Components

These components are located on the client machine and enable access to the NetUI system. Figure 1.2 illustrates the NetUI client components.



Fig. 1.2
NetUI Client
Components

Components

Java Plug-In or Java Runtime Environment (JRE). This component is installed on the Web server from the NetUI media. (JRE, which includes the Java Plug-In, is installed starting with Service Pack 2.)

It is then downloaded to each client machine during the client setup process. The Java Plug-In runs the JVM that is responsible for executing the platform-independent Java bytecode on a specific operating system.

QAD security files. This component is installed on the Web server from the NetUI media. It is downloaded to each client machine during the NetUI client setup. The QAD security files verify the authenticity of the client to run the NetUI Java files.

Web browser. This component is installed on each client machine using the browser vendor's installation instructions. It performs the following functions:

- Provides access to the NetUI HTML pages that launch a NetUI session
- Checks the security permissions and the QAD security files before executing the NetUI Java code

- Calls the Java Plug-In, which launches the JVM that executes the NetUI Java code
- Performs basic Web functions, such as displaying the NetUI program windows

NetUI Java Code. This component is installed on the Web server from the NetUI media. It is downloaded to each client machine during the NetUI client setup. These files contain the Java classes that comprise the NetUI Java code.

Interactions

- 1 A user launches the Web browser on the NetUI client machine and accesses a NetUI HTML file on the Web server.
- 2 The NetUI HTML file specifies the Java Plug-In, which bypasses the browser's default JVM.
- 3 The Java Plug-In accesses the QAD security files and runs the NetUI Java code.

Maintenance and Transaction Components

► For more details on these components, see *User Guide: eB Desktop and Network User Interface*.

These components can be located on one or more machines and enable MFG/PRO maintenance and transaction programs accessed by NetUI clients.

There are two ways to access maintenance and transaction programs in the NetUI:

- A telnet interface. All of the maintenance and transactions programs are available using telnet. Figure 1.3 illustrates the telnet interface.

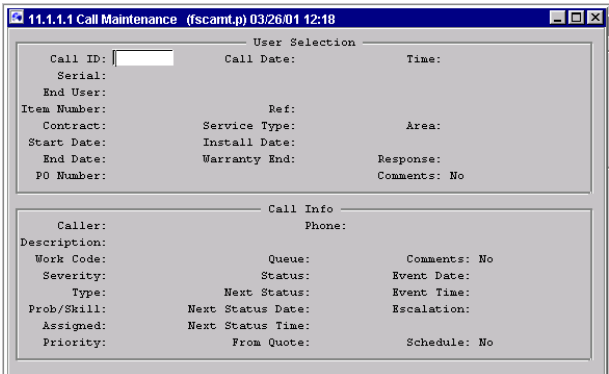


Fig. 1.3
NetUI Telnet
Interface

- A configurable graphical user interface (GUI). This interface is referred to as the extended user interface (XUI), and is only available for a subset of programs.

Note You can also access programs using the optional eB Desktop, which presents screens with HTML navigation.

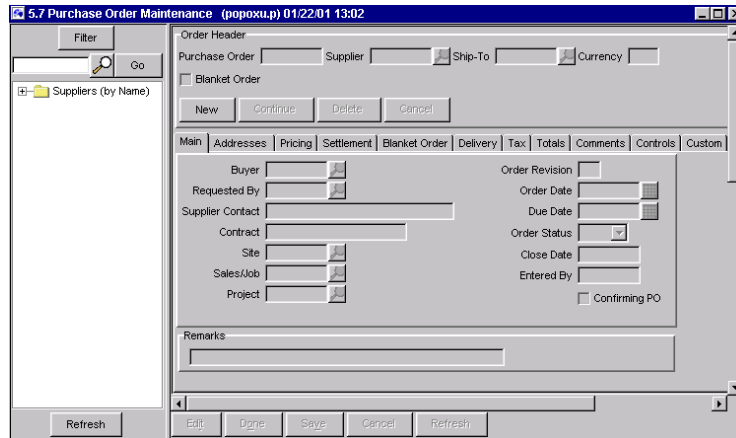
Table 1.3 lists the programs that are available in the XUI.

Menu	Description	Program
1.4.1	Item Master Maintenance	ppptxu.p
2.1.1	Customer Maintenance	adcuxu.p
2.3.1	Supplier Maintenance	adsuxu.p
5.3.1	Blanket Order Maintenance	popoxu1.p
5.7	Purchase Order Maintenance	popoxu.p
5.13.1	Purchase Order Receipts	porcxu.p
7.1.4	Sales Order Read	sosoxu.p
13.5	Product Structure Maintenance	bmpsxu.p
14.13.1	Routing Maintenance	rwroxu.p
14.13.2	Routing Maintenance (Rate Based)	rwroxu1.p
15.5	Formula Maintenance	fmpsxu.p
15.13	Process Definition Maintenance	rwroxu2.p
27.10	Customer Credit Review/Update	arcsxu.p

Table 1.3
XUI Programs

Figure 1.4 is an example of a program using the extended user interface.

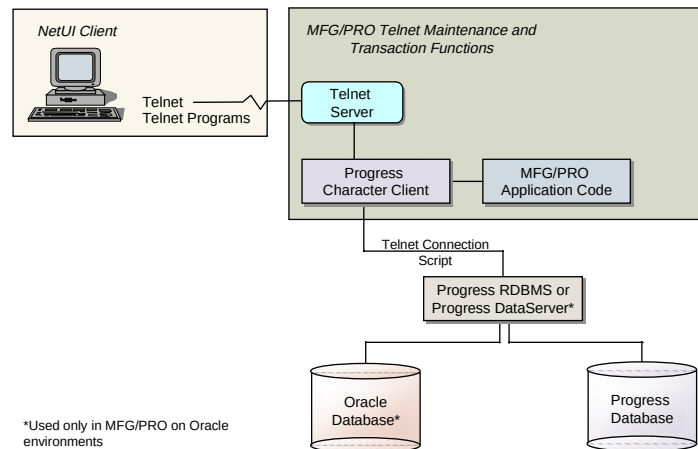
Fig. 1.4
Sample XUI
Program



Telnet Interface

The following components enable access to MFG/PRO maintenance and transaction programs using the NetUI's telnet interface. Figure 1.5 illustrates the telnet components.

Fig. 1.5
Telnet Interface
Components



Components

Telnet server. Depending on the operating system you are using, this component is either already present or installed from the NetUI media.

- For UNIX servers, use the telnet server provided with your operating system.
- For Windows NT servers, use the Georgia SoftWorks telnet server included on the NetUI media.

This component provides the telnet interface for MFG/PRO maintenance and transaction programs accessed through the NetUI.

Progress character client. This component was installed with the Progress software when you implemented MFG/PRO. It provides the client executable for MFG/PRO maintenance and transaction programs accessed using the NetUI's telnet interface.

MFG/PRO application code. This component was installed during the MFG/PRO implementation. The application code for NetUI clients is part of the MFG/PRO character client code.

Progress RDBMS. This component was installed during the MFG/PRO implementation. It provides access to the MFG/PRO databases.

MFG/PRO Progress or Oracle databases. These components were created during the MFG/PRO implementation. The MFG/PRO databases contains the MFG/PRO system data.

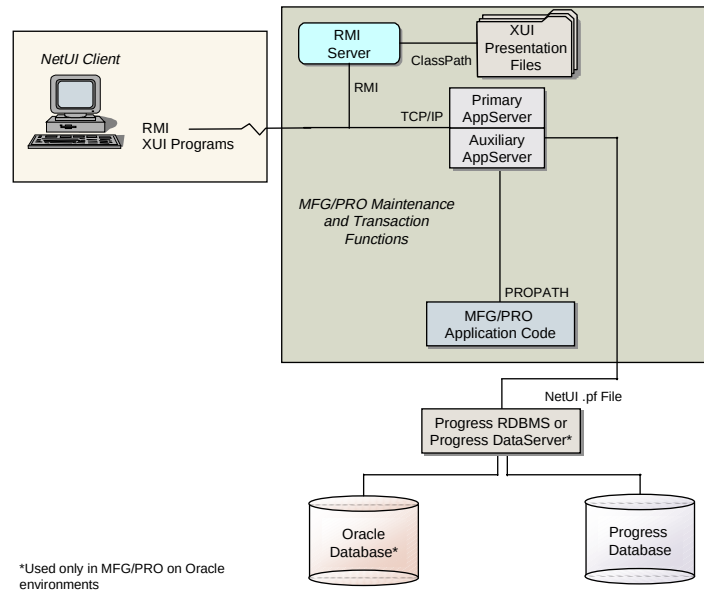
Interactions

- 1 A NetUI user requests a telnet maintenance or transaction program.
- 2 The telnet server launches a Progress character client session.
- 3 The client sessions access the MFG/PRO character client code to run the requested program.
- 4 The client session accesses the MFG/PRO databases using a connection script.

Extended Interface

The following components enable access to the MFG/PRO maintenance and transaction programs using the XUI. Figure 1.6 illustrates the XUI components.

Fig. 1.6
XUI Components



Components

RMI server. This component is created using the NetUI Java code and the JDK. It enables the NetUI Java code on the client machine to make remote procedure calls (RPCs) and access the NetUI files on the Web server.

XUI presentation files. This component is installed on the RMI server from the NetUI media. The presentation files define the screen presentation for MFG/PRO maintenance and transaction programs using the XUI.

Primary and auxiliary AppServers. These components are included with the Progress software. They provide the client executables for XUI maintenance and transaction programs.

MFG/PRO application code. This component was installed during the MFG/PRO implementation. The application code for NetUI clients is part of the MFG/PRO character client code.

Progress RDBMS. This component was installed during the MFG/PRO implementation. It provides access to the MFG/PRO databases.

MFG/PRO Progress or Oracle databases. These components were created during the MFG/PRO implementation. The MFG/PRO databases contain the MFG/PRO system data.

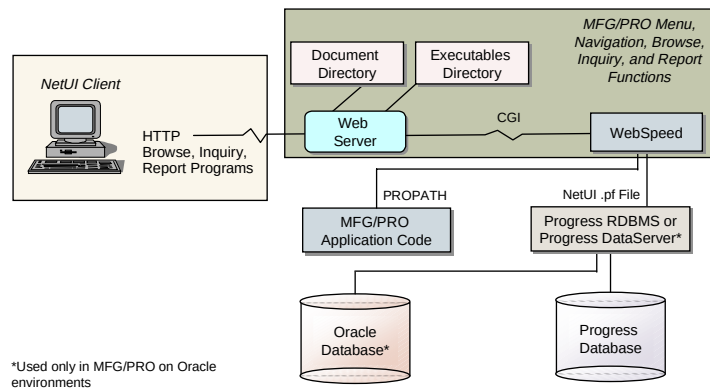
Interactions

- 1 A NetUI user requests an XUI maintenance or transaction program.
- 2 An AppServer instance launches an application server process that accesses the MFG/PRO character client code for the requested program.
- 3 The application server process accesses the MFG/PRO databases using a parameter file.
- 4 The RMI server accesses the NetUI presentation files to provide the XUI presentation to the NetUI.

Menu, Navigation, Browse, Inquiry, and Report Components

These components can be located on one or more machines and enable the MFG/PRO menu and navigation functions for NetUI clients. Additionally, they enable MFG/PRO browse, inquiry, and report programs accessed by NetUI clients. Figure 1.7 illustrates the menu, navigation, report, browse, and inquiry components.

Fig. 1.7
Menu, Navigation,
Browse, Inquiry,
and Report
Components



Components

Web server. This component is installed using the Web server vendor's installation instructions. It provides access to the WebSpeed and HTML components of the NetUI.

Document directory. This component was created during the Web server installation. This directory contains the NetUI HTML files that provide the initial access point for the NetUI. It is also the directory into which the NetUI software is initially installed.

Executables directory. This component was created during the Web server installation. This directory contains the NetUI executables—such as the NetUI online help executable and WebSpeed messenger executable.

WebSpeed. This component is included with the Progress software. It provides the client executables for MFG/PRO browse, report, and inquiry programs accessed through the NetUI.

MFG/PRO application code. This component was installed during the MFG/PRO implementation. The application code for NetUI clients is part of the MFG/PRO character client code.

Progress RDBMS. This component was installed during the MFG/PRO implementation. It provides access to the MFG/PRO databases.

MFG/PRO Progress or Oracle databases. These components were created during the MFG/PRO implementation. The MFG/PRO databases contain the MFG/PRO system data.

Interaction

- 1 A NetUI user requests a browse, report, or inquiry program.
- 2 The WebSpeed messenger contacts the WebSpeed broker.
- 3 The WebSpeed broker assigns an agent process to the client request.
- 4 The agent process accesses the MFG/PRO character client code to run the requested browse, inquiry, or report program.
- 5 The agent process accesses the MFG/PRO databases using a parameter file.

Progress Server Architecture

The NetUI uses Progress WebSpeed Transaction Server and AppServer instances to manage NetUI client requests for MFG/PRO programs and database records. Before you begin setting up your WebSpeed server and AppServer instances, you should understand the components that make up these instances.

Additionally, all of the Progress server products—including the WebSpeed and AppServer products—use the Progress administration framework to provide consistent administration and management functionality. A basic understanding of this framework facilitates setting up and managing the WebSpeed server and AppServer instances.

These sections provide only the basic information you need to set up your NetUI environment. For more detailed information about any of the following components, see the Progress documentation for your system.

Administrative Framework

The Progress administrative framework consists of the following components:

- The AdminServer
- The unified broker properties file
- The NameServer
- The Progress Explorer
- Server management utilities

AdminServer

The AdminServer provides administration functions for all of the Progress server products. The Progress Explorer and the server management utilities run as clients of the AdminServer.

Unified Broker Properties File

The unified broker properties file (`ubroker.properties`) defines the server instances in your environment. These include the NameServer, WebSpeed server, and AppServer instances used by the NetUI.

NameServer

The NameServer directs NetUI client requests to the appropriate WebSpeed and AppServer instances. The instructions in this guide set up the WebSpeed and AppServer instances to use the default NameServer (NS1) supplied by Progress.

Progress Explorer

The Progress Explorer is a graphical user interface tool for the `ubroker.properties` file. Using this tool, you can create and configure the WebSpeed server and AppServer instances for your NetUI environment.

The Progress Explorer offers an easier and less error-prone method of modifying the `ubroker.properties` file. You can also modify the properties file using a text editor if you cannot use the Progress Explorer in your environment. This guide contains configuration instructions for both situations.

Server Management Utilities

The server management utilities let you stop, start, and check the status of the Progress server instances in your environment. Table 1.4 lists the management utility for each server product.

Server	Utility
AdminServer	proadsv
NameServer	nsman
WebSpeed server	wsbman
AppServer	asbman

Table 1.4
Server
Management
Utilities

WebSpeed Transaction Server

The WebSpeed Transaction Server provides the Progress client executables for NetUI menu and navigation functions and for MFG/PRO report, browse, and inquiry programs accessed through the NetUI. The NetUI requires at least one WebSpeed server instance.

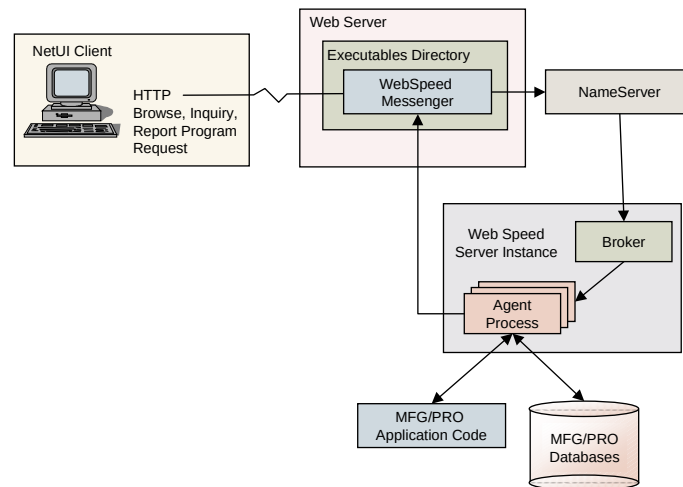
A WebSpeed server instance consists of:

- A pool of WebSpeed client executables called agents
- A broker that manages the agents

In addition to agents and brokers, WebSpeed server instances use a WebSpeed messenger executable. The messenger handles the transfer of data between the Web server and the WebSpeed agent during a single Web transaction. It is installed in the Web server's executables directory during the WebSpeed installation process.

Figure 1.8 illustrates a WebSpeed server instance and its interactions with the other NetUI components.

Fig. 1.8
WebSpeed Server Instance



AppServer

The AppServer provides the Progress client executable for the XUI programs in the NetUI. The XUI requires at least two AppServer instances: a primary instance and an auxiliary instance.

An AppServer instance consists of:

- A pool of AppServer client executables called server processes
- A broker that manages the server processes

Figure 1.9 illustrates a single AppServer instance and its interactions with the other NetUI components.

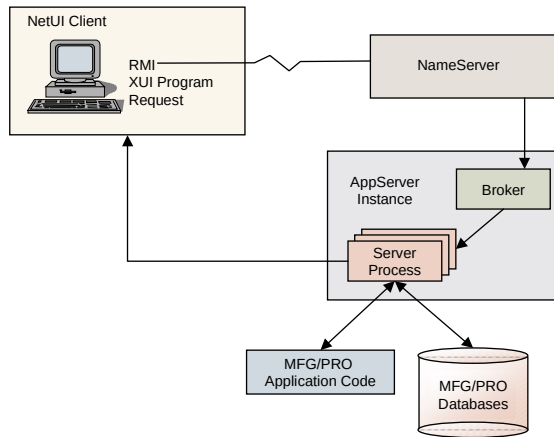


Fig. 1.9
AppServer Instance

Data Security

In this release of the NetUI, all traffic between the client and the server is initially insecure and, with the use of a network sniffer, security could be compromised.

Various additional system security options are available. See the following topics:

- For UNIX telnet servers, “Configuring the Telnet Environment” on page 120
- “Securing Internet Communication” on page 209

System Requirements

A NetUI implementation requires planning for adequate system resources. This chapter provides guidelines you can use to generate estimates of the hardware and software requirements.

Skills Required for Implementation **34**

Prerequisite Requirements **34**

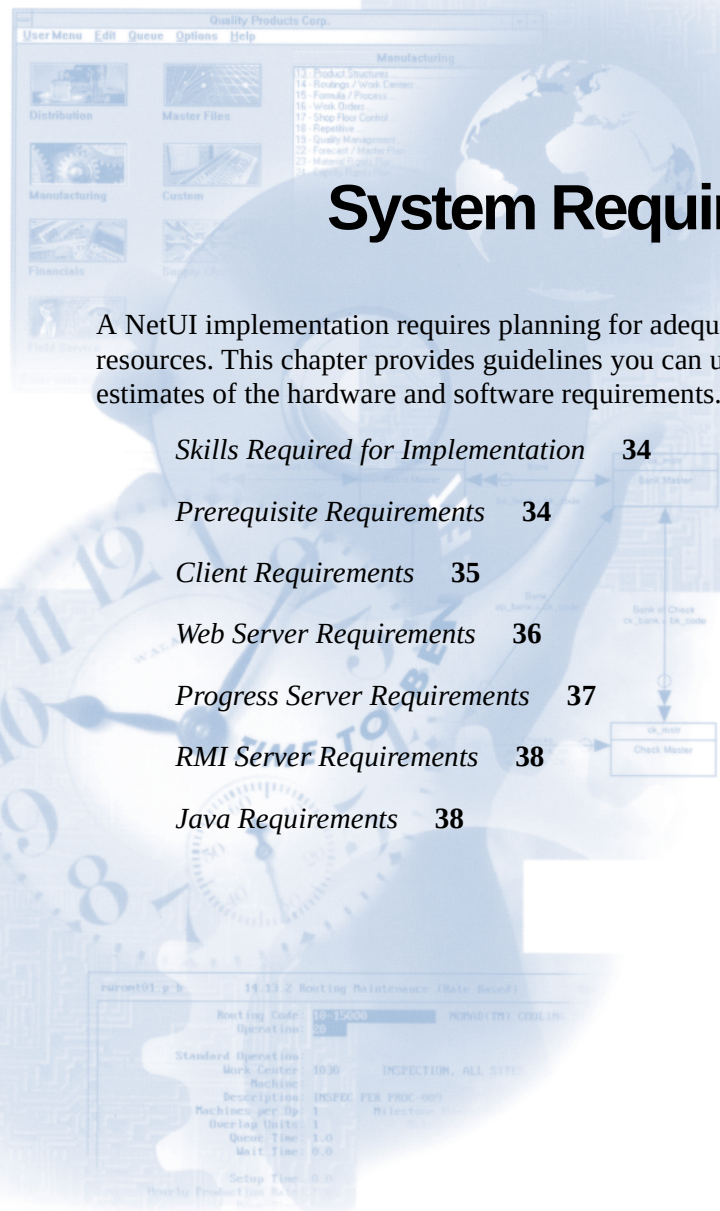
Client Requirements **35**

Web Server Requirements **36**

Progress Server Requirements **37**

RMI Server Requirements **38**

Java Requirements **38**



The background image shows a NetUI application window titled "Quality Products Corp." with a menu bar (User Menu, Edit, Queue, Options, Help) and a "Manufacturing" section. Below the menu are icons for Distribution, Master Files, Manufacturing, Custom, Financials, and Inventory. A list of manufacturing options is visible: 13 Product Structures, 14 Routings / Work Centers, 15 Formula / Process, 16 Work Orders, 17 Shop Floor Control, 18 Repetitive, 19 Quality Management, 20 Forecast / Master Plan, 21 Material Requirements Planning, and 22 Engineering Change Order.

At the bottom, a "Routing Maintenance (Main Screen)" window is open, displaying the following data:

Routing Code: 10-15000		DESCRIPTION: CORR. INS.
Operation:	20	
Standard Operation:	10.00	INSPECTION, ALL SITES
Work Center:		
Description:	INSPEC PER PROC-005	
Machines per Op:	1	Allocation:
Overlap Units:	1	
Queue Time:	1.0	
Wait Time:	0.0	
Setup Time:	0.0	
Route by Production Route		

Skills Required for Implementation

NetUI implementation and maintenance requires experience with the following:

- System administration, including:
 - Either UNIX or Windows NT server operating system
 - User management
 - Network administration, specifically TCP/IP network
 - Device administration
 - File system administration
 - Windows client administration
- Progress software experience, including basic WebSpeed and AppServer installation and administration
- MFG/PRO administration
- Progress or Oracle database administration
- Web server installation and administration
- Basic HTML
- Basic telnet
- Basic Java and RMI

Prerequisite Requirements

▶ For more information, see the MFG/PRO installation guide for your system.

- The required Progress software components for MFG/PRO eB
- MFG/PRO eB database server
- MFG/PRO eB character client code
- Internet access

Client Requirements

Operating Systems

The following operating systems are supported:

- Microsoft Windows 95 and 98
- Microsoft Windows NT 4.0
- Microsoft Windows 2000

Client Software

- QAD's NetUI client environment, which includes:
 - Sun's Java Plug-In 1.1.3 for NetUI installations prior to Service Pack 2 or JRE 1.3.0_0.2 or higher starting with Service Pack 2.
For non-English installations, the internationalized version of the Java Plug-In is required. This is supplied with the NetUI software.
 - QAD security files
 - NetUI Java code
- Web browser software. Supported browsers are:
 - Microsoft Internet Explorer 5.0 or later.
 - Netscape Communicator 4.6 and 4.7 for NetUI installations prior to Service Pack 2. Starting with Service Pack 2, Netscape 6.0 is also supported.

▶ See "Installing Client Files" on page 168.

Important NetUI uses cookies to store information on the client. Make sure that the browser is set up to allow cookies.

Note If you plan to use the optional eB Desktop with the NetUI, different client requirements exist. You *must* use Microsoft Internet Explorer 5.5 or later and install JRE 1.3.0_0.2 or higher.

Minimum Client Hardware

- Depending on the client operating system, the following memory is required:
 - For Windows 95 and 98 clients, 64 MB of RAM

- For Window NT and 2000 clients, 128 MB of RAM
- P400 or faster processor
- At least 15 to 25 MB of disk space, depending on the version of Java
- Ability to display 65,536 or more colors

Web Server Requirements

Operating Systems

NetUI clients communicate with the Web server using network protocols and do not require a specific operating system on the server. Use the operating system that best suites your needs. You should choose the operating system based on criteria such as your experience or familiarity with the operating system, performance needs, and available support.

Software

- You can use any Web server software that supports a common gateway interface (CGI). Examples include:
 - Netscape Enterprise Server, Version 2.01 or higher
 - Microsoft Internet Information Server (IIS)
- The Progress WebSpeed messenger

Hardware

Minimum hardware requirements differ depending on the Web server software and operating system you implement. The performance and scalability of the NetUI is directly related to proper server sizing. If you are unsure or inexperienced in performance and capacity planning, you should obtain professional assistance before purchasing any hardware.

Software and Hardware Sizing

The following components of the NetUI are most affected by Web server hardware resources:

- The Web server software. See the software vendor’s sizing guidelines for your Web server’s operating system.
- The Progress WebSpeed software. See the Progress WebSpeed sizing guidelines for your Web server’s operating system.

CPU Requirements

The required number of central processing units (CPUs) and their speed depends on the type of hardware and operating system you are implementing. See the hardware vendor’s product documentation and the operating system documentation for sizing guidelines.

Disk Space Requirements

The Web server must have enough disk space for the following NetUI components.

NetUI Component	Disk Space
Web server software	Required disk space depends on the server software you implement
WebSpeed software	100 MB
NetUI software	150 MB for the NetUI Java code and HTML files

Table 2.1.
Disk Space
Requirements

Progress Server Requirements

Operating Systems

NetUI clients communicate with the Progress servers using network protocols and do not require that the machines on which these servers run use a specific operating system.

The NetUI supports the following platforms:

- Windows NT (Intel)
- Sun Solaris (SPARC)
- IBM AIX
- Digital UNIX (Tru64)
- HP-UX
- SCO UnixWare
- Linux

Hardware

▶ See the System Requirements chapter in the MFG/PRO installation guide for your system.

The Progress servers are responsible for executing the MFG/PRO code for the NetUI clients. The machine on which these servers run should be sized in a manner similar to a server running MFG/PRO for an equivalent number of MFG/PRO character clients.

Servers

▶ See “WebSpeed Agent Sizing and Behavior” on page 206.

- Progress NameServer Enterprise Edition, Version 9.1
- Progress WebSpeed Transaction Server, Version 3.1 with sufficient WebSpeed agent licenses to support expected transaction volume
- Progress AppServer, Version 9.1

RMI Server Requirements

▶ See “NetUI Service Packs and JDK Versions” on page 10.

- Software: Sun Microsystems Java Development Kit (JDK). The version needed depends on your service pack level.
- Network: An open TCP/IP network port.

Note If you plan to use the eB Desktop, JDK 1.3.0_02 or greater is always required. See *Installation Guide: eB Desktop* for details.

Java Requirements

Sun Microsystems JDK is required for the RMI server and NetUI online help. The version needed depends on your service pack level:

- Prior to Service Pack 2, JDK 1.1.8 is required on the Web server and RMI server.
- Starting with Service Pack 2, JDK 1.3.0_0.2 or higher is required on the Web server and RMI server.

▶ See “Installing the Java Development Kit” on page 57.

On Windows NT systems, JDK 1.1.8 may be installed as part of installing Progress, depending on which Progress components you install. If you are using a UNIX server or have Service Pack 2 or higher, you must download the JDK from the Sun Microsystems Java Web site.

Note If you plan to use the eB Desktop, JDK 1.3.0_02 or greater is always required.

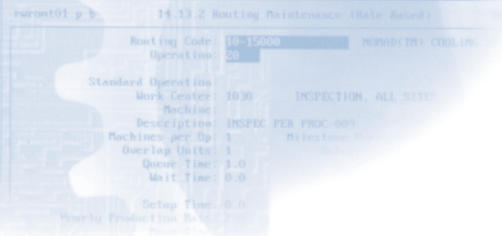
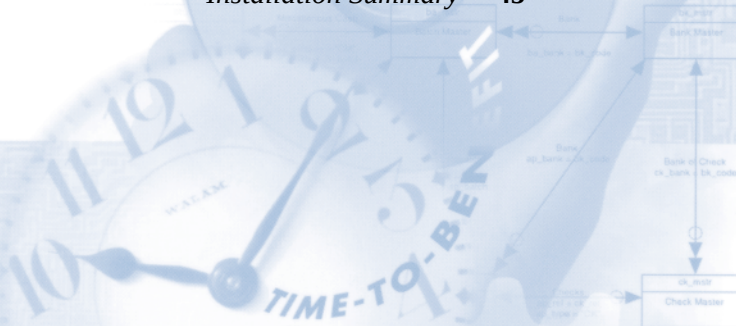
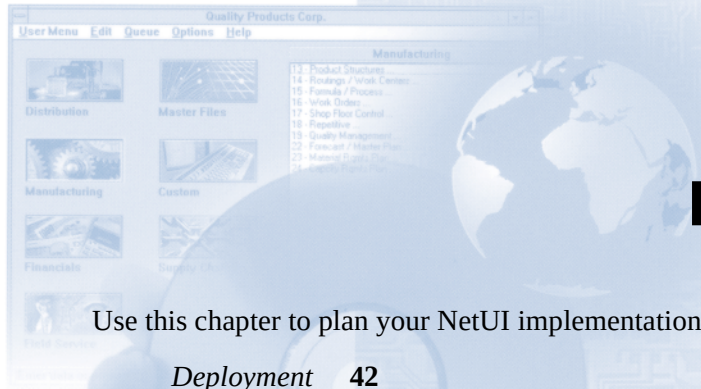
CHAPTER 3

Planning

Use this chapter to plan your NetUI implementation.

Deployment 42

Installation Summary 45



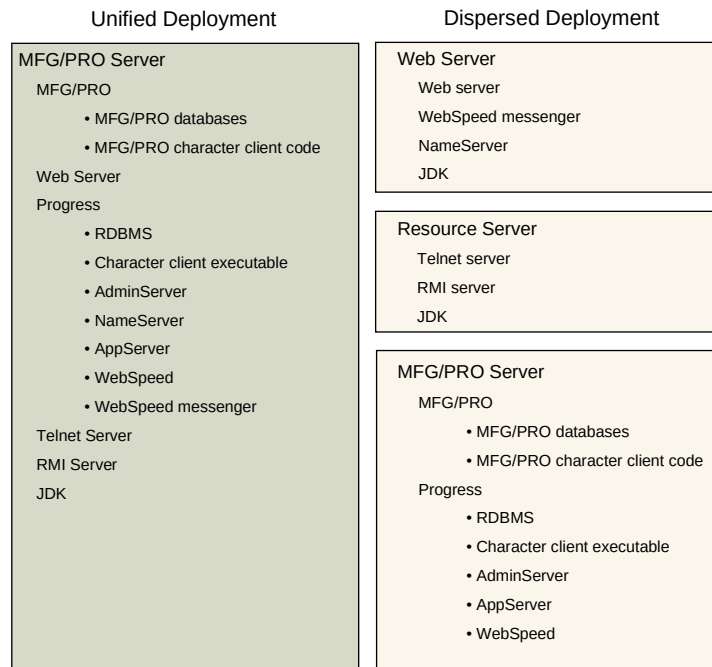
Deployment

Deployment Options

The various components of MFG/PRO and the NetUI can be deployed in several configurations. You can install all of the components on a single adequately sized server in a unified deployment or on multiple servers in a dispersed deployment. Both deployments offer advantages for different MFG/PRO environments.

Figure 3.1 illustrates examples of both unified and dispersed deployments.

Fig. 3.1
NetUI
Deployments



Unified Deployment

In a unified deployment, all of the NetUI components are implemented on the same physical machine as the MFG/PRO components. With an adequately sized server, this option can provide the following benefits:

- Increased system performance because of reduced network traffic and the absence of network dependencies
- Simplified system administration because all of the components are on the same physical machine

Dispersed Deployment

In a dispersed deployment, the Web, resource, and MFG/PRO server components of the NetUI are implemented on two or more machines. With adequately sized servers and appropriate network bandwidth, this option can provide the following benefits:

- Increased system performance for remote users
- Division of NetUI computing requirements between server machines

A dispersed deployment may require you to implement NetUI components on more than one machine. This type of deployment also causes increased network overhead for NetUI component communication.

QAD recommends that you group the NetUI components as they appear in Figure 3.1. Although it is possible to separate the NetUI components further, the dispersed deployment in Figure 3.1 contains logical groupings that consider component interdependencies, overall system performance, and ease of administration.

Important You should not attempt a dispersed deployment without the help of a network engineer.

Deployment Issues

When planning the NetUI deployment, be sure to consider each of the following issues.

System Characteristics

Consider your expectations for system characteristics such as accessibility, reliability, scalability, performance, and maintenance. Different NetUI deployments may improve one characteristic while degrading another. You should implement the NetUI in the deployment that best achieves your expectations.

For example, a unified deployment may ease administration because all of the components are located on the same server. However, to provide better performance for remote users, you may want to deploy a dispersed configuration with components located on the remote user's local-area network (LAN).

Location of Prerequisite Components

The location of prerequisite components—such as the MFG/PRO databases, the MFG/PRO character client code, and their related Progress software—affect the deployment you implement. Keep in mind that spreading components across multiple machines increases network delays and may reduce overall system performance.

Existing Infrastructure

If you decide to deploy the NetUI in a dispersed configuration, the overall system performance is directly affected by:

- Network topology, bandwidth, and throughput
- The processing power of the machines involved in the deployment

For example, you may find that while a component runs faster on a dedicated server—removing CPU contention with other components—the network delays that result from moving the item actually decrease system performance.

NetUI Architecture

Although NetUI components can be spread across multiple machines to suit your needs, certain components rely on each other and must be present on the same machine. When considering deployment options, keep NetUI dependencies in mind and be aware that copies of key components may need to exist on multiple machines.

▶ See Figure 3.1 on page 42.

Installation Summary

Figure 3.2 provides a summary of the major tasks required to install the NetUI.

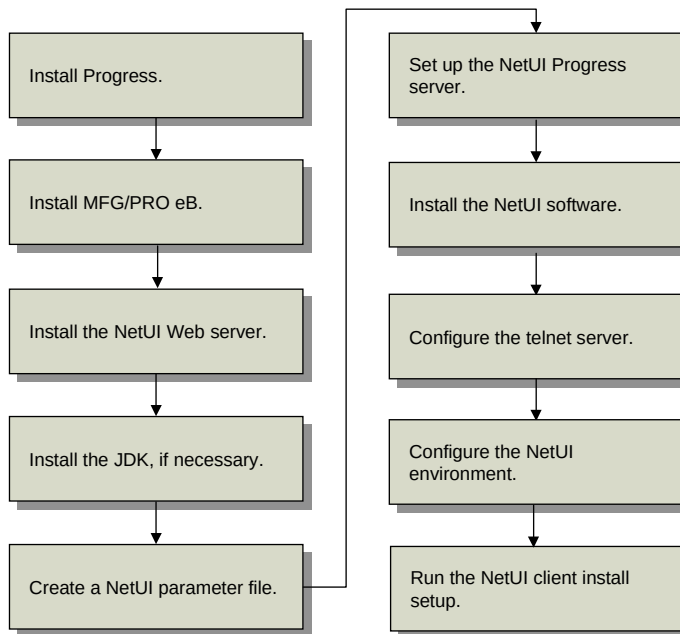


Fig. 3.2
NetUI Installation
Summary

The following sections break each of the major installation task into activities. Use this information as a guide to planning and performing each task.

Install MFG/PRO eB

Before you begin installing the NetUI, you must install and configure an MFG/PRO eB environment. In general, this task involves completing the following activities. Instructions for completing these activities are not included in this guide.

▶ See the Progress installation and configuration documentation.

- Perform a complete installation of the Progress components required by MFG/PRO eB.

Important Do not use the typical or custom options when you install Progress. You must perform a *complete* installation.

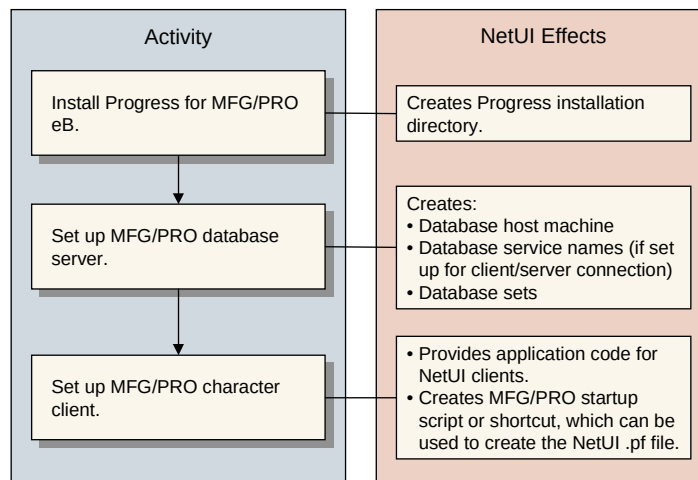
▶ See the MFG/PRO installation guide for your system.

- Implement MFG/PRO eB including MFG/PRO character clients. The MFG/PRO application has two sets of code: one for graphical user interface (GUI) clients and the other for character clients. NetUI clients use the MFG/PRO character client code.

Because NetUI clients use the character client code, you may want to install the MFG/PRO code on multiple servers. You can install the character client code from the character client media at any time.

Figure 3.3 illustrates the MFG/PRO eB installation activities and some of the effects these activities have on your NetUI environment.

Fig. 3.3
Installing
MFG/PRO eB



Important Before you begin implementing the NetUI, fill in Table 12.1 on page 220 with the prerequisite component information.

Install the NetUI Web Server

Figure 3.4 illustrates the Web server installation activities and some of the effects these activities have on your NetUI environment. Specific instructions for these activities are included in Chapter 4, “Setting Up the NetUI Environment,” on page 55.

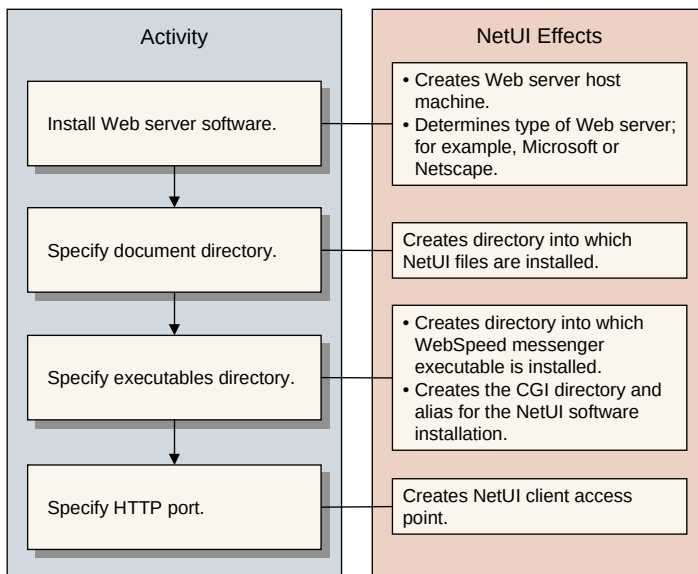


Fig. 3.4
Installing the Web Server

Install the Java Development Kit (JDK)

Sun Microsystems JDK is required for the RMI server and NetUI online help. The version needed depends on your service pack level:

- Prior to Service Pack 2, JDK 1.1.8 is required on the Web server and RMI server.
- Starting with Service Pack 2, JDK 1.3.0_0.2 or higher is required on the Web server and RMI server.

Note The optional eB Desktop product always requires at least JDK 1.3.0_02. The requirements for eB Desktop are discussed in *Installation Guide: eB Desktop*.

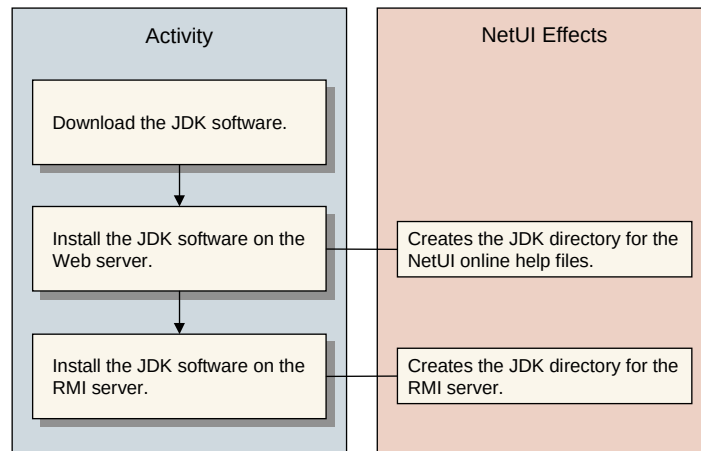
You must always complete this step on UNIX systems by downloading the JDK from the Sun Microsystems Java Web site. On Windows NT

systems, JDK 1.1.8 may have been installed when you installed Progress, depending on which Progress components you selected.

Important If installed with Progress on a Windows NT system, JDK 1.1.8 is located in the \$DLC/jdk directory. Installing it again in a different location may prevent the WebSpeed agents from starting. Check to ensure that this directory does not exist before downloading and installing the JDK.

Figure 3.5 illustrates JDK installation activities and some of the effects these activities have on your NetUI environment. Specific instructions for these activities are included in Chapter 4, “Setting Up the NetUI Environment,” on page 55.

Fig. 3.5
Installing the JDK



Important The RMI server and Web server can be located on the same machine. In this case, you can install the JDK once in a single directory and use this directory for both the RMI server and NetUI online help.

Create a NetUI Parameter File

Figure 3.6 illustrates creating a NetUI parameter file and the effect this has on your NetUI environment. Specific instructions this activity are included in Chapter 4, “Setting Up the NetUI Environment,” on page 55.

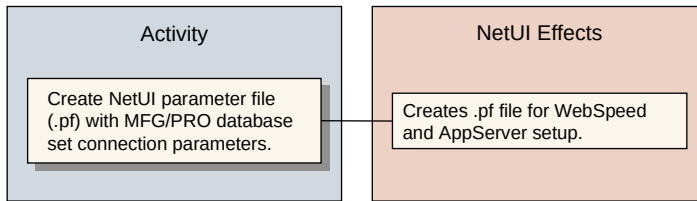


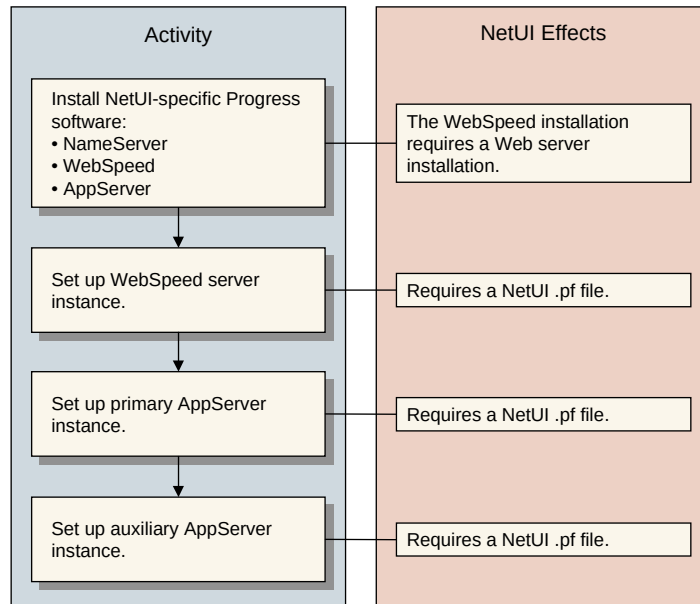
Fig. 3.6
Creating a NetUI
Parameter File

Important At this point, you must decide whether the NetUI WebSpeed agents and AppServer application processes will access the MFG/PRO databases using local host or client/server connections.

Set Up NetUI Progress Servers

Figure 3.7 illustrates the activities involved in setting up the NetUI Progress servers and some of the effects these activities have on your NetUI environment. Specific instructions for these activities are included in Chapter 4, “Setting Up the NetUI Environment,” on page 55.

Fig. 3.7
Setting Up the
NetUI Progress
Software



Important You must install the NetUI Progress software based on how you decided to access the MFG/PRO databases.

- If you decided to use a client/server connection, you can install the WebSpeed and AppServer software anywhere.
- If you decided to use a local host connection, you must install the WebSpeed and AppServer software on the MFG/PRO database server.

Install NetUI Software

Figure 3.8 illustrates the NetUI software installation activities and some of the effects these activities have on your NetUI environment. Specific instructions for these activities are included in the appropriate chapter for the Web server's operating system.

- For UNIX Web servers, see Chapter 5, “Installing the NetUI Software on UNIX,” on page 97.
- For Windows NT Web servers, see Chapter 6, “Installing the NetUI Software on Windows NT,” on page 109.

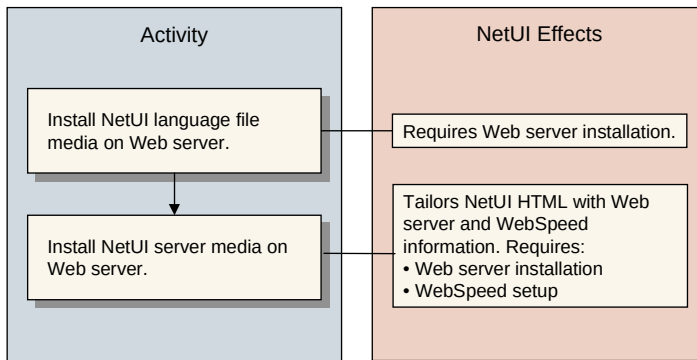


Fig. 3.8
Installing the NetUI
Software

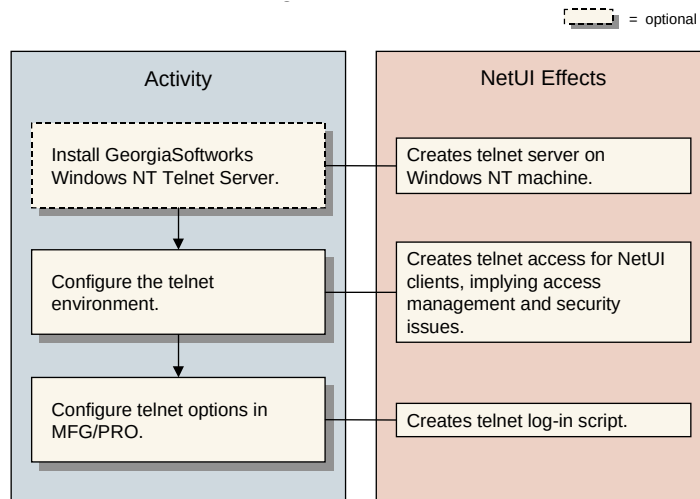
Important The NetUI installation process copies files to the Web server's primary document directory only. If your NetUI deployment requires NetUI files on additional machines, you must move them yourself.

Configure the Telnet Server

Figure 3.9 illustrates the telnet server configuration activities and some of the effects these activities have on your NetUI environment. Specific instructions for these activities are included in the appropriate chapter for the telnet server's operating system.

- For UNIX telnet servers, see Chapter 7, “Configuring Telnet on UNIX,” on page 119.
- For Windows NT telnet servers, see Chapter 8, “Configuring Telnet on Windows NT,” on page 131.

Fig. 3.9
Configuring the
Telnet Server



Configure the NetUI Environment

Figure 3.10 illustrates the NetUI configuration activities and some of the effects these activities have on your NetUI environment. Specific instructions for these activities are included in Chapter 9, “Configuring the NetUI Environment,” on page 149.

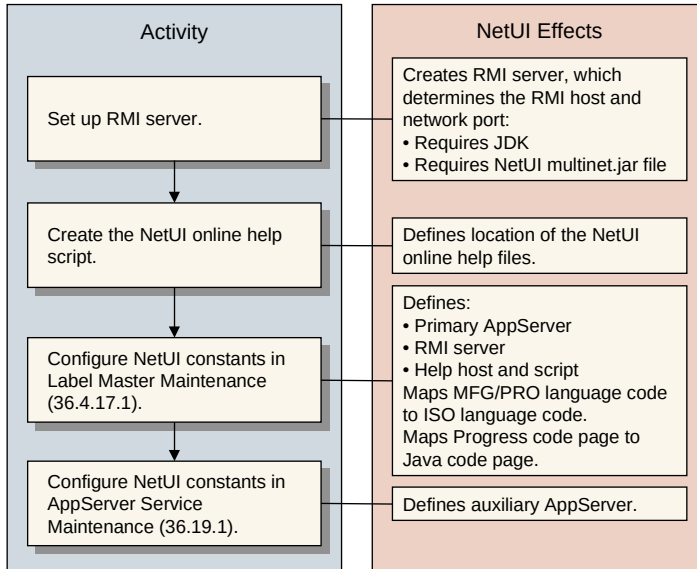
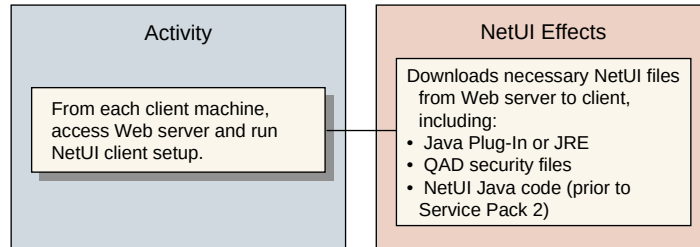


Fig. 3.10
Configuring the
NetUI Environment

Run the NetUI Client Setup

Figure 3.11 illustrates the NetUI client setup activities and some of the effects these activities have on your NetUI environment. Specific instructions for these activities are included in Chapter 9, “Configuring the NetUI Environment,” on page 149. A different client install script is used before and after you install Service Pack 2.

Fig. 3.11
Running the NetUI
Client Install



Note If you plan to use the optional eB Desktop product, you do not need to complete the NetUI client installation. The eB Desktop installation replaces the NetUI client installation.

CHAPTER 4

Setting Up the NetUI Environment

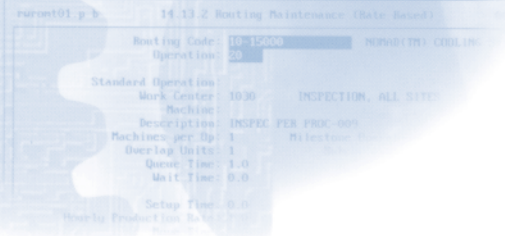
Use this chapter to set up your environment before installing the NetUI software.

Installing the NetUI Web Server 56

Installing the Java Development Kit 57

Creating a NetUI Parameter File 58

Setting Up the NetUI Progress Servers 63



Installing the NetUI Web Server

Use the following instructions to install and set up your NetUI Web server.

Install the Web Server

Tip

For required versions, refer to “Web Server Requirements” on page 36.

- 1 Use the installation instructions provided by the Web server software vendor to install your Web server.

For example, to install Netscape Enterprise Server, download it from the Netscape Web site:

<http://www.netscape.com/download/>

- 2 Locate your Web server’s document directory. If one does not exist, you must create one.

Record this information in Table 12.2 on page 222. You must supply this directory during the NetUI software installation.

♦ For instructions, refer to your Web server’s documentation.

- 3 Locate your Web server’s executables directory and alias. Although there is no default executables directory, it is recommended that you use the default executables alias, `cgi-bin`. If an executables directory and alias do not exist, you must create them.

Record this information in Table 12.2 on page 222. You must specify this information during the WebSpeed and NetUI software installations.

Set Up the MIME-Type

♦ See “Installing Client Files” on page 168.

Web servers insert Multi-Purpose Internet Mail Extension (MIME) types in a header at the beginning of any Web transmission. Clients use this header to select an appropriate application for the type of data transmitted.

During client setup, you use the Web browser on the client machine to download the NetUI client files. If you have a UNIX Web server, use the following instructions to enable clients to associate the proper application with the client setup files.

- 1 Remove .exe from magnus-internal and any other MIME-type.

type=magnus-internal/cgi exts=cgi,bat

- 2 Add .exe to the application/octet-stream MIME-type.

type=application/octet-stream exts=bin,exe

Verify the Web Server

To verify that the Web server is functioning, start the Web browser on a client machine. Then, enter a URL referencing only the Web server name. For example, enter the following and verify that there is a response.

`http://www.MyWebServerURL.com`

If this test fails, follow the troubleshooting steps outlined in your Web server's documentation.

Installing the Java Development Kit

The JDK is required for the RMI server and NetUI online help. The version needed depends on your service pack level:

- Prior to Service Pack 2, JDK 1.1.8
- Starting with Service Pack 2, JDK 1.3.0_0.2

Note The optional eB Desktop product always requires at least JDK 1.3.0_02. The requirements for eB Desktop are discussed in *Installation Guide: eB Desktop*.

You must always complete this step on UNIX systems by downloading the JDK from the Sun Microsystems Java Web site. On Windows NT systems, JDK 1.1.8 may have been installed when you installed Progress, depending on which Progress components you selected.

Important If installed with Progress on a Windows NT system, JDK 1.1.8 is located in the \$DLC/jdk directory. Installing it again in a different location may prevent the WebSpeed agents from starting. Check to ensure that this directory does not exist before downloading and installing the JDK.

Depending on your NetUI deployment, you may need to install the JDK on more than one machine. Use the following list to determine the machines on which you must install the JDK.

- The JDK must be installed on the same machine as the RMI server.
- The JDK must be installed on the Web server to enable NetUI help.

Important You must have internet access to download the JDK software.

To download the JDK 1.1.8 software, access the Sun Microsystems Java Web site:

www.java.sun.com/products/jdk/1.1/

To download the JDK 1.3.0_0.2 software, access the following site:

www.java.sun.com/products/archive/j2se/1.3.0_02/index.html

Install the JDK on each machine where it is required. Record the location of the JDK in Table 12.2 on page 222. You will need to supply this information when you configure the NetUI environment.

Creating a NetUI Parameter File

◆ For details, see the Database Sets section of the Implementation Overview chapter in the MFG/PRO installation guide for your system.

Complete this section to create a parameter file (extension .pf) for the NetUI environment. When you set up the WebSpeed server and the AppServer, you assign a NetUI parameter file to the WebSpeed server and AppServer brokers. The agents and application server processes controlled by the broker use the parameter file to connect to the MFG/PRO databases in a database set.

This section provides instructions for creating a NetUI parameter file for the MFG/PRO Production database set. In subsequent sections, you set up WebSpeed server and AppServer instances for the Production database set using this parameter file.

Depending upon the implementation of other interfaces in your MFG/PRO environment, you may already have files available that you can use to create a NetUI parameter file.

- If you implemented Windows MFG/PRO clients—either graphical user interface (GUI) or Windows character clients—parameter files exist for the database sets for which you generated client startup shortcuts. You can modify these files to create a NetUI parameter file.

- If you implemented UNIX character clients only, you can modify the MFG/PRO client startup scripts to create a NetUI parameter file.

Connection Parameters

The connection parameters contained in the NetUI parameter file depend on whether the WebSpeed and AppServer processes connect to the MFG/PRO databases using local host or client/server connections.

Local Host Connection Parameters

If the WebSpeed and AppServer brokers are on the same server as the MFG/PRO databases, you can use local host or shared memory connections to access the databases. This can improve NetUI performance by eliminating network overhead between the processes and the databases.

To enable local host connections, the NetUI parameter files must contain the connection parameters described in Table 4.1.

Parameter	Description
-db	The physical database name parameter. Follow this parameter with the path and physical name of an MFG/PRO database.
-trig	The triggers parameter. Follow this parameter with the name of the directory containing the database triggers for the main MFG/PRO database. For MFG/PRO this is the triggers subdirectory. The parameter value is: -trig triggers. This parameter is only used with main databases.
-ld	The logical database name parameter. Follow this parameter with the logical name of an MFG/PRO database. This parameter is only used with support databases.

Table 4.1
Local Host
Connection
Parameters

Client/Server Connection Parameters

If the WebSpeed and AppServer brokers are on a different machine than the MFG/PRO databases, you must use client/server connections to access the databases. This can improve NetUI system performance by spreading the MFG/PRO and NetUI resource requirements between two servers.

To enable the server processes to make client/server connections, the NetUI parameter files must contain the connection parameters described in Table 4.2.

Table 4.2
Client/Server
Connection
Parameters

Parameter	Description
-db	The physical database parameter name. Follow this parameter with the physical name of the MFG/PRO database. You do not need to include the path to the database file.
-trig	The triggers parameter. Follow this parameter with the name of the directory containing the database triggers for the main MFG/PRO database. For MFG/PRO this is the <code>triggers</code> subdirectory. The parameter value is: <code>-trig triggers</code> . This parameter is only used with main databases.
-ld	The logical database name parameter. Follow this parameter with the logical name of an MFG/PRO database. This parameter is only used with support databases.
-S	The database service name parameter. Follow this parameter with an MFG/PRO database service name. You can use the Database Set Maintenance utility in MFG/UTIL to find the service name for a database.
-N	The network parameter. Follow this parameter with the network protocol used to connect to the MFG/PRO databases. For MFG/PRO, this is TCP/IP. The parameter value is: <code>-N TCP</code> .
-H	The host name parameter. Follow this parameter with the machine name or IP address of the MFG/PRO database server. This guide uses <i>DBServer</i> as an example in place of an actual machine name or IP address.

Create the Parameter File

For UNIX character client environments, you create the NetUI parameter file by copying and editing the client startup script for the Production database set.

For all other environments, create the NetUI parameter file by copying and editing the MFG/PRO parameter file for the Production database set.

Use the instructions appropriate for your environment to create the NetUI parameter file.

UNIX Character Client Environments

- 1 Locate the MFG/PRO client startup script for the Production database set. In the default MFG/PRO configuration, the character client startup scripts are located in the MFG/PRO database server administration directory.

- 2 Make a copy of the client startup script and rename it something meaningful to your installation. Use the file extension `.pf`:

```
cp client.Production Production.pf
```

Record the location of the NetUI parameter file in Table 12.2 on page 222. You must supply the location of this file when you set up the Progress servers.

- 3 Open the parameter file in a text editor:

```
vi Production.pf
```

- 4 Use the following instructions to edit the parameter file:

- a Remove all of the lines up to the first database in the set. Leave only the database connection parameters.

- b Remove any back slashes that might appear at the end of a line:

```
-db /qad/mfgpro/eB/db/mfgprod.db -trig triggers \
```

- c Remove the `-p mf.p` parameter from the last line:

```
-d mdy -yy 1920 -Bt 350 -c 30 -D 100 -mmax 8192 -nb 200  
-s 63 -p mf.p
```

- 5 Save your changes and close the text editor.

The following examples are the edited versions of the `Production.pf` file for both Progress and Oracle database environments.

Progress Database Environments

Example Local Host Connections: This example is for local host connections and contains only the `-db`, `-ld`, and `-trig` parameters. Remember when configuring local host connections that you must include the full path and physical name of the database after the `-db` parameter:

```
-db /qad/mfgpro/eB/db/mfgprod.db -trig triggers
-db /qad/mfgpro/eB/db/hlpprod.db -ld qadhelp
-db /qad/mfgpro/eB/db/admprod.db -ld qadadm
-d mdy -yy 1920 -Bt 350 -c 30 -D 100 -mmax 8192 -nb 200
-s 63
```

Example Client/Server Connections: This example is for client/server connections and contains the `-H`, `-S`, and `-N` parameters:

```
-db mfgprod -H uxsvr1 -S mfgprod-srv -N tcp
-trig triggers

-db mfghelp -ld qadhelp -H uxsvr1 -S mfghelp-srv
-N tcp

-db admin -ld qadadm -H uxsvr1 -S admin-srv
-N tcp

-d mdy -yy 1920 -Bt 350 -c 30 -D 100 -mmax 8192 -nb 200
-s 63
```

Oracle Database Environment

Example Oracle Database Connections:

```
-db SchemaHolder -ld qad -R0 -znotrim -trig triggers
-db ORACLE_SID -ld qaddb -dt ORACLE -U qad@ORACLE_SID
-P qad -c 255

-cpinternal iso8859-1 -cpstream iso8859-1 -cpcoll basic
-d mdy -yy 1920 -Bt 350 -D 100 -mmax 8192 -nb 200 -s 63
```

GUI Client or Windows NT Server Environments

- 1 Locate the MFG/PRO parameter file for the Production database set; for example, `product.i.pf`. This file was generated by MFG/UTIL when you configured the MFG/PRO client startup shortcut and is located by default in the client working directory.
- 2 Make a copy of this file and rename it something meaningful to your NetUI installation; for example, `Production.pf`.
Record the location of the NetUI parameter file in Table 12.2 on page 222. You must supply the location of this file when you set up the Progress servers.
- 3 Open the parameter file in a text editor and modify it for your NetUI environment using the following instructions:
 - a Verify that the connection parameters in the parameter file are appropriate for your NetUI environment.
If you are creating a NetUI parameter file for local host connections, make sure the file contains the connection parameters listed in Table 4.1.
If you are creating a NetUI parameter file for client/server connections, make sure the file contains the connection parameters listed in Table 4.2.
 - b Remove the `-p mfg.p` parameter from the last line of the file.
- 4 Save your changes and close the text editor.

Setting Up the NetUI Progress Servers

Use this section to set up the Progress servers for the NetUI. These include the:

- WebSpeed server, which the NetUI uses for menus and navigation, and to run MFG/PRO browse, inquiry, and report programs. This server is required for the NetUI.
- AppServer, which the NetUI uses for extended user interface (XUI) MFG/PRO transaction and maintenance programs. This server is required to use the XUI.

Setup Prerequisites

Before using these instructions, you must install the Progress software using the Progress installation and configuration guide appropriate to your system. In addition to the Progress software required by MFG/PRO, you must also install:

- NameServer Enterprise Edition
- AppServer
- WebSpeed Enterprise Transaction Server

Important When installing Progress software, select the *complete* installation option. This ensures that all of the components and subcomponents of each product are installed.

Setup Environments

Since the Progress software can be installed on either a Windows NT or UNIX server, this section contains instructions for both operating systems. Use the instructions appropriate to your environment.

- If you installed the Progress software on a Windows NT server, use the instructions in “Set Up on Windows NT” on page 65.

You can also use these instructions if you have a Windows machine network-connected to a UNIX server. The Windows machine must have the Progress Explorer installed and you must be able to log into the UNIX system as a valid user. QAD recommends this option, since the Progress Explorer—available only for Windows—provides an easy-to-use configuration interface.

- If you installed the Progress software on a UNIX server, use the instructions in “Set Up on UNIX” on page 83.

Set Up on Windows NT

This section provides instructions for setting up the WebSpeed Server and AppServer instances using the Progress Explorer, which runs on Windows machines only. You can use these instructions to set up:

- A Windows NT server
- A UNIX server using a network-connected Windows machine

In the following instructions, you use the Progress Explorer to modify the WebSpeed and AppServer sections of the `ubroker.properties` configuration file. This file is located in the `properties` subdirectory below the Progress installation directory.

Launch the Progress Explorer

Use these instructions to start the AdminServer and launch the Progress Explorer.

- 1 Verify that the AdminServer process is running. Use the instructions appropriate to the server where the Progress software is located.

- If the software is located on a Windows NT server, enter the following command from the system command line:

```
Progress\bin\proadsv -query
```

If the AdminServer is not running, open Settings|Control Panel|Services from the Windows Start button. Then, select the AdminService for Progress 9.1 and click Start.

- If the software is located on a UNIX server, run the following command on that server:

```
Progress/bin/proadsv -query
```

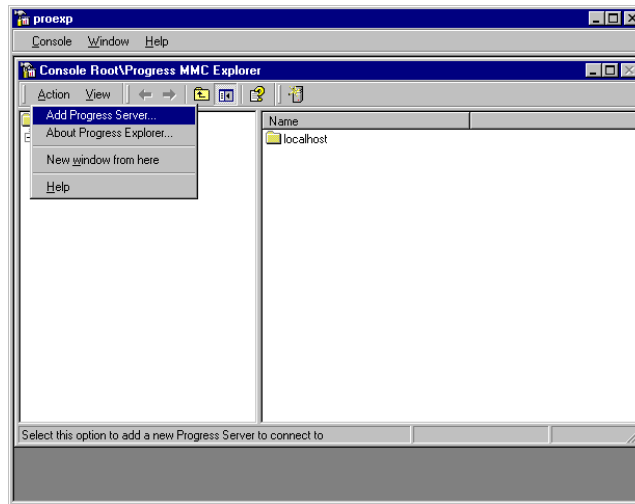
If the AdminServer is not running, start it using the following command:

```
Progress/bin/proadsv -start
```

- 2 Once the AdminServer is started, open Programs|Progress 91|Progress Explorer Tool from the Windows Start button. The Progress Explorer window displays.

- 3 If you are configuring a remote machine, use the following instructions to add an icon for the remote machine to the Progress Explorer. If you are configuring the Progress servers for this machine, use the localhost icon.

- a Open Action|Add Progress Server.

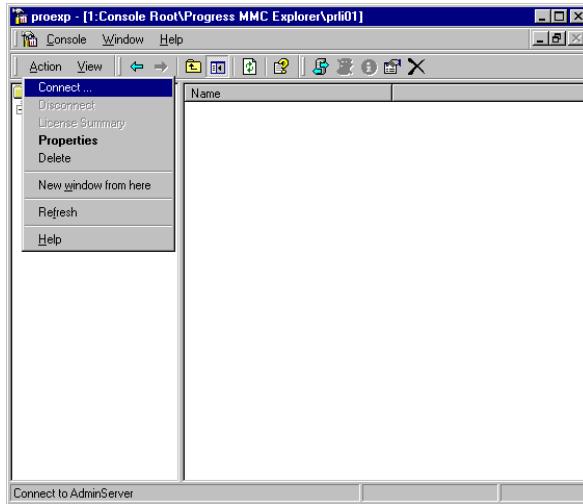


- b Complete the General tab in the Server Properties window using the following table as a guide.

Field	Description
Server	Enter the machine name of the server you want to configure.
User	Enter the user name under which you want to log in to the remote machine. If you are configuring a remote UNIX machine, you log in as a user with root-level permissions. If you are configuring a remote Windows NT machine, log is as a user with administrator permissions.
Password	Enter the password for the user which you are using to log in to the remote machine.

- c Click OK to continue.

- 4 In the Progress Explorer window, right-click the server icon for the machine on which you want to configure the WebSpeed server and AppServer instances. Select Connect.



- 5 If you are configuring a remote machine, you are prompted for a user name and password. Use the same name and password that you specified in step 3.

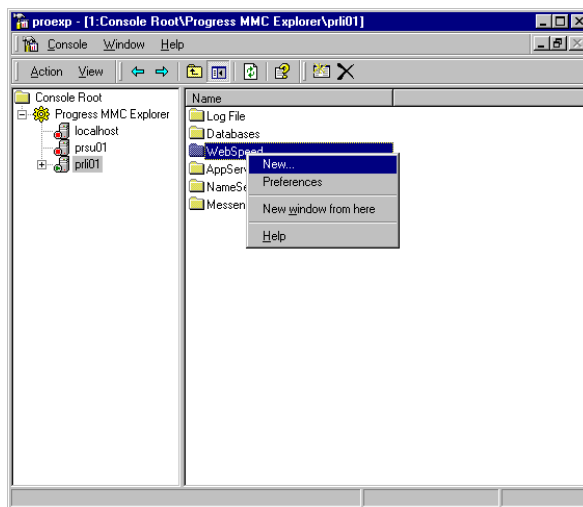
The Progress Explorer connects to the AdminServer process running on the machine where you are configuring WebSpeed and AppServer instances. A connection is indicated by a green arrow in the machine icon.

Set Up the WebSpeed Server

Use these instructions to set up a WebSpeed server instance. At least one Webspeed server instance is required for the NetUI.

Setting Up

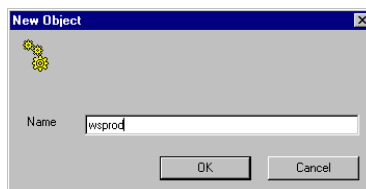
- 1 Once you have connected to the server machine, right-click the WebSpeed folder and select New.



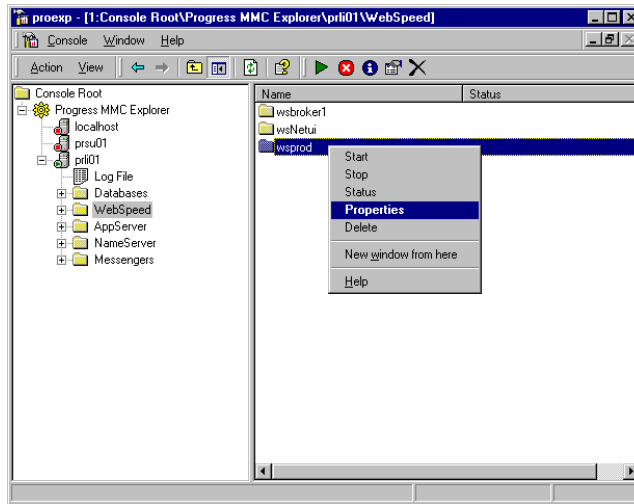
- 2 In the New Object window, enter the name you want to use for the WebSpeed broker. The following instructions use wsprod as an example.

Record the broker name in Table 12.2 on page 222. You must supply this name during the NetUI software installation.

When ready, click OK to continue.

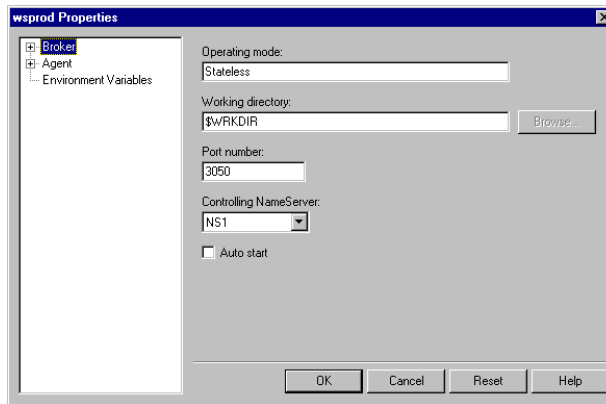


- 3 Right-click the new WebSpeed server instance and select Properties.



- 4 In the Properties window, select Broker and use the following table to complete the broker fields.

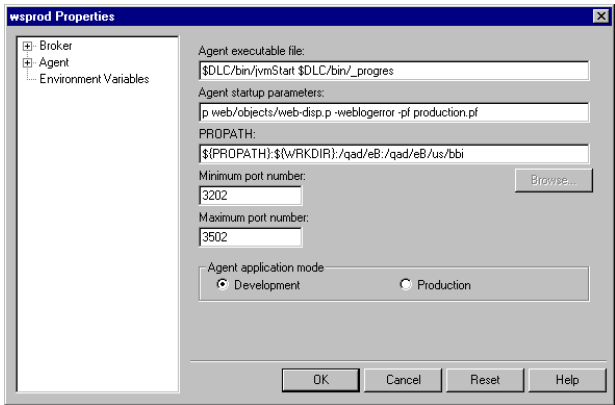
Field	Description
Operating mode	Accept the default in this field.
Working directory	The working directory this WebSpeed server instance will use. You can accept the default in this field or enter an alternate working directory.
Port number	An available network port for this broker. Record this information in Table 12.2 on page 222. You must supply this port during the NetUI software installation.
Controlling NameServer	Accept the default in this field.
Auto start	Select this option if you want this broker and the agents it controls to start when the server on which they run is started.



5 Select Agent and use the following table to complete the agent fields.

Field	Description
Agent executable file	Accept the default in this field.
Agent startup parameters	Enter the <code>-pf</code> parameter followed by the path and name of the NetUI parameter file. See “Creating a NetUI Parameter File” on page 58.
PROPATH	Enter the Progress search path (PROPATH) for this broker’s processes. The PROPATH must contain the following directories: • The current working directory (.). • The directory containing the MFG/PRO character client code. • The <code>bbi</code> subdirectory below the language-code directory of the language you are implementing. The <code>bbi</code> subdirectory is located below the MFG/PRO character client code directory. • Any directories containing custom code that you want NetUI clients to use.
Minimum port number	Enter the lowest port this broker’s agents can use.

Field	Description
Maximum port number	Enter the highest port this broker’s agents can use.
Agent application mode	Accept the default, Development. This mode provides enhanced troubleshooting capabilities, which you can use for initial installation and tuning. Once you have configured and tuned the NetUI environment to your satisfaction, select Production to reduce the number of agents each NetUI client session uses. For more information, see the Progress <i>WebSpeed Installation and Configuration Guide</i> .



- 6
- Click OK in the Properties window to complete the WebSpeed server setup.

Verifying

To verify that the WebSpeed server instance is set up correctly, you should start it and check its status. QAD recommends that you manage the WebSpeed server instance using the command line rather than the Progress Explorer. Use the commands in Table 4.3 to verify your WebSpeed instance.

For additional commands, see the Progress *WebSpeed Installation and Configuration Guide*.

In Table 4.3, *Progress* refers to the Progress installation directory and *BrokerName* refers to the name of your WebSpeed broker.

Important The commands in Table 4.3 are case sensitive. Be sure to enter *BrokerName* with the correct upper-case and lower-case letters.

Table 4.3
Broker Start, Stop,
and Status
Commands

Action	Command
Start Broker	<i>Progress\bin\wtbman -name BrokerName -start</i>
Stop Broker	<i>Progress\bin\wtbman -name BrokerName -stop</i>
Check Broker Status	<i>Progress\bin\wtbman -name BrokerName -query</i>

1 Start the MFG/PRO database servers for the Production database set.

Depending on the operating system of your MFG/PRO database server, you can use either of the following methods to start the MFG/PRO database servers:

- For UNIX servers, use the database startup script for the Production database set; for example, `start.Production`.
- For Windows NT servers, use the startup shortcut for the Production database set.

When you start the WebSpeed broker, the broker's agents attempt to connect to the databases specified in the NetUI .pf file. This connection fails if the servers for the databases are not running.

Important Always start the appropriate MFG/PRO database servers before starting WebSpeed server or AppServer brokers. Additionally, if you need to stop the WebSpeed or AppServer brokers, you should stop the associated database servers.

2 Use the startup command in Table 4.3 to start the WebSpeed server instance.

3 Use the query command in Table 4.3 to check the status of the WebSpeed instance.

4 After you check the status, leave the instance running.

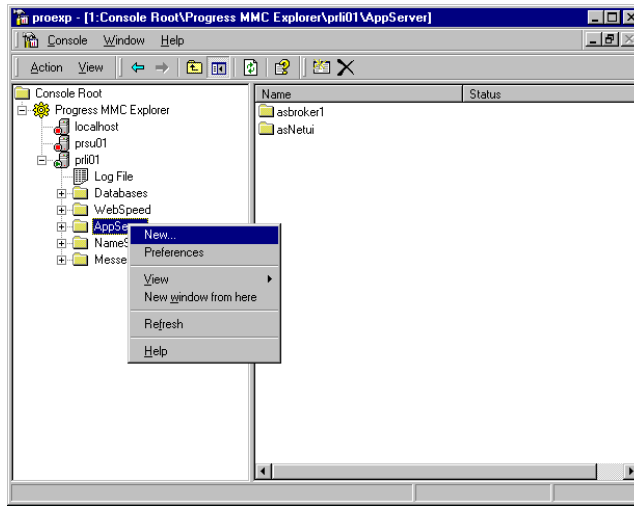
Note In addition to the wtbman management utility, you may also need to refer to the log files for the WebSpeed server instance. To view the log files, expand the instance folder in the left-hand Progress Explorer pane and click the appropriate log file. The log displays in the right-hand Explorer pane.

Set Up the Primary AppServer

Use these instructions to set up the primary AppServer instance. You must complete this setup to use XUI-enabled programs.

Setting Up

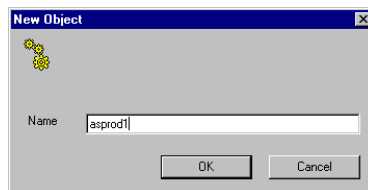
- 1 In the left-hand pane of the Explorer window, right-click AppServer and select New.



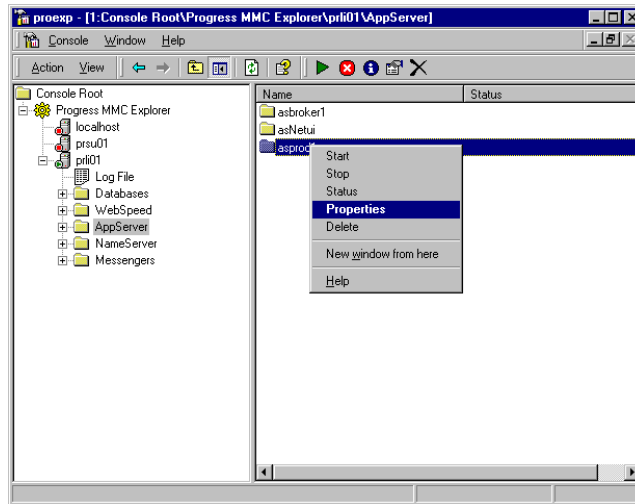
- 2 In the New Object window, enter the name you want to use for the primary AppServer broker. The following instructions use asprod1 as an example.

Record the broker name in Table 12.2 on page 222. You must supply this name when you configure the NetUI constants.

When ready, click OK to continue.

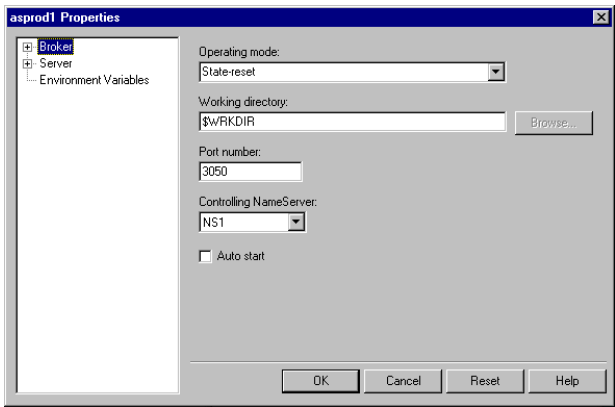


- 3 Right-click the new primary AppServer instance and select Properties.



- 4 In the Properties window, select Broker and use the following table to complete the broker fields.

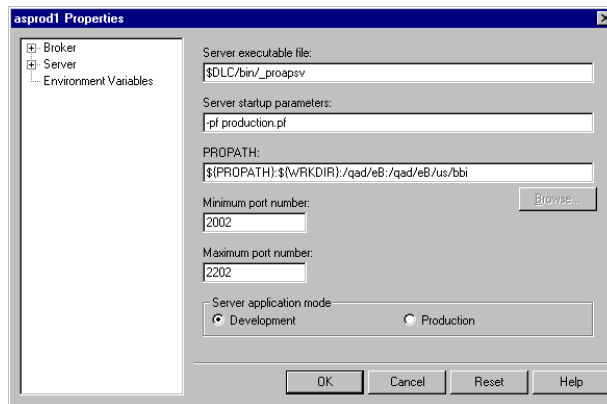
Field	Description
Operating mode	Accept the default in this field.
Working directory	The working directory this server instance will use. You can accept the default in this field or enter an alternate working directory.
Port number	An available network port for this broker. Record this information in Table 12.2 on page 222. You must supply this port number when you configure the NetUI constants.
Controlling NameServer	Accept the default in this field.
Auto start	Select this option if you want this broker and the agents it controls to start when the server on which they run is started.



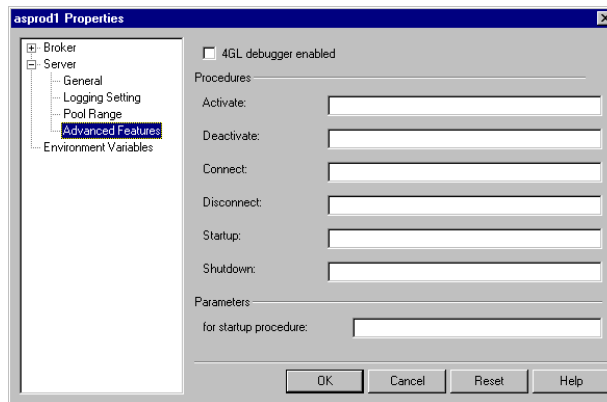
5 Select Server and use the following table to complete the server fields.

Field	Description
Server executable file	Accept the default in this field.
Server startup parameters	Enter the -pf parameter followed by the path and name of the NetUI parameter file. See “Creating a NetUI Parameter File” on page 58.
PROPATH	Enter the Progress search path (PROPATH) for this broker’s processes. The PROPATH must contain the following directories: • The current working directory (.). • The directory containing the MFG/PRO character client code. • The bbi subdirectory below the language-code directory of the language you are implementing. The bbi subdirectory is located below the MFG/PRO character client code directory. • Any directories containing custom code that you want NetUI clients to use.
Minimum port number	Enter the lowest port this broker’s processes can use.

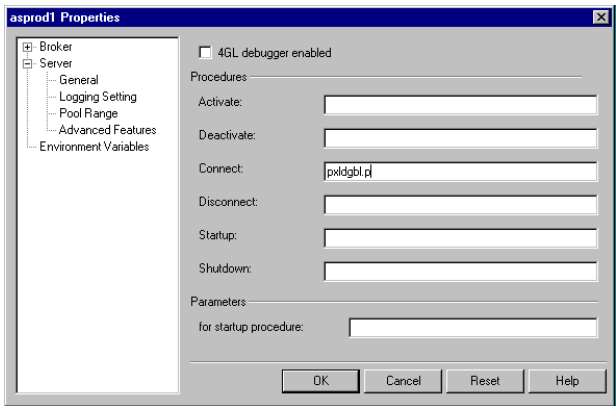
Field	Description
Maximum port number	Enter the highest port this broker's processes can use.
Server application mode	Accept the default, Development. This mode provides enhanced troubleshooting capabilities, which you can use for initial installation and tuning. Once you have configured and tuned the NetUI environment to your satisfaction, select Production to reduce the number of processes each NetUI client session uses. For more information, see the Progress <i>Building Distributed Applications Using the Progress AppServer</i> documentation.



- 6 In the left-hand pane of the Properties window, expand the Server options and select Advanced Features.



- 7 In the Connect field, enter `pxldgbl.p`. This program loads the run-time parameters required by the XUI.



- 8 When ready, click OK in the Properties window to complete the primary AppServer setup.

Verifying

To verify that the primary AppServer instance is set up correctly, you should start it and check its status. QAD recommends that you manage AppServer instances using the command line rather than the Progress Explorer. Use the commands in Table 4.4 to verify an AppServer instance.

In Table 4.4, *Progress* refers to the Progress installation directory and *BrokerName* refers to the name of the broker for the AppServer instance.

For additional commands, see the Progress *Building Distributed Applications with the Progress AppServer* documentation.

Action	Command
Start Broker	<code>Progress\bin\asbman -name BrokerName -start</code>
Stop Broker	<code>Progress\bin\asbman -name BrokerName -stop</code>
Check Broker Status	<code>Progress\bin\asbman -name BrokerName -query</code>

Table 4.4
Broker Start, Stop,
and Status
Commands

Tip

If the database servers are not running, use the instructions in “Verifying” on page 71 to start them.

- 1 Verify that the MFG/PRO database servers for the Production database set are running.
- 2 Use the startup command in Table 4.4 to start the primary AppServer.
- 3 Use the query command in Table 4.4 to check the status of the primary AppServer.
- 4 After you check the status, leave the instance running.

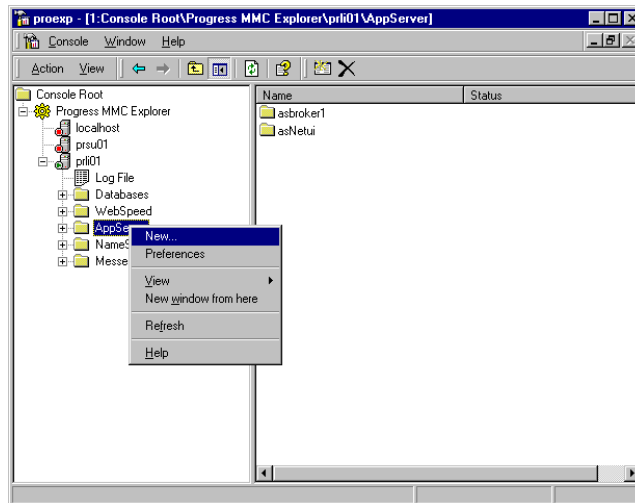
Note In addition to the asbman management utility, you may also need to refer to the log files for an AppServer instance. To view the log files, expand the instance folder in the left-hand Progress Explorer pane, and click the appropriate log file. The log displays in the right-hand Explorer pane.

Set Up the Auxiliary AppServer

Use these instructions to set up the auxiliary AppServer instance. You must complete this setup to use XUI-enabled programs.

Setting Up

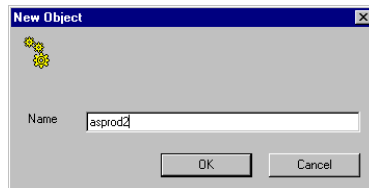
- 1 In the left-hand pane of the Explorer window, right-click AppServer and select New.



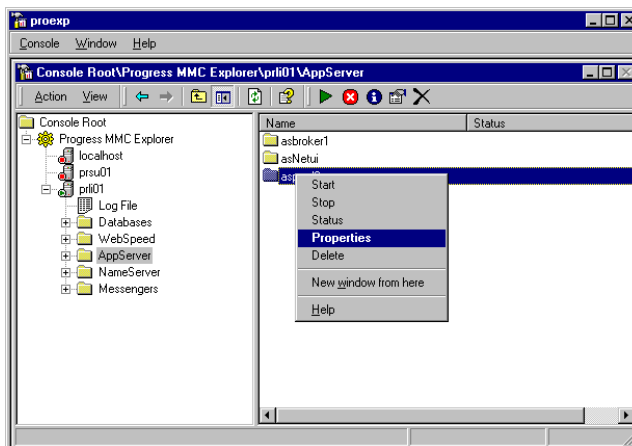
- 2 In the New Object window, enter the name you want to use for the auxiliary AppServer broker. The following instructions use asprod2 as an example.

Record the broker name in Table 12.2 on page 222. You must supply this name when you configure the NetUI constants.

When ready, click OK to continue.

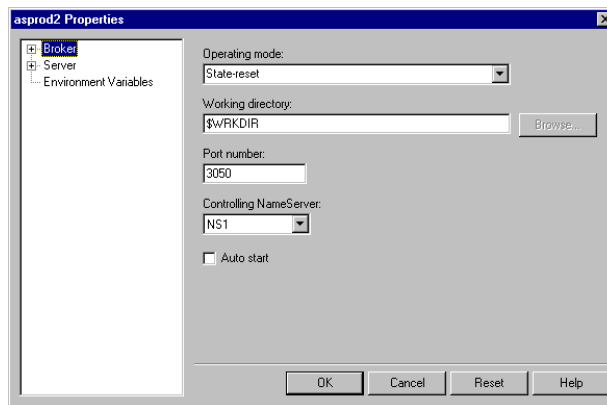


- 3 Right-click the new auxiliary AppServer instance and select Properties.



- 4 In the Properties window, select Broker and use the following table to complete the broker fields.

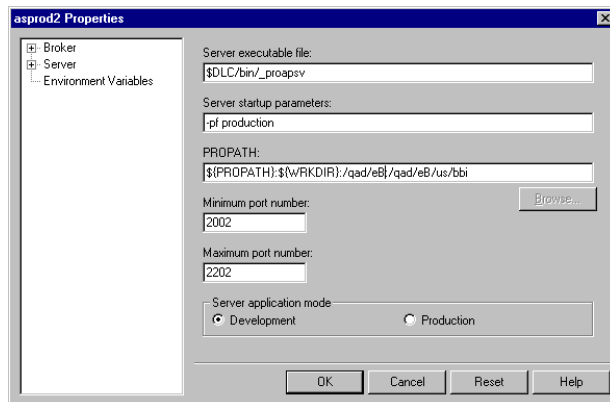
Field	Description
Operating mode	Accept the default in this field.
Working directory	The working directory this server instance will use. You can accept the default in this field or enter an alternate working directory.
Port number	An available network port for this broker. Record this information in Table 12.2 on page 222. You must supply this port number when you configure the NetUI constants.
Controlling NameServer	Accept the default in this field.
Auto start	Select this option if you want this broker and the agents it controls to start when the server on which they run is started.



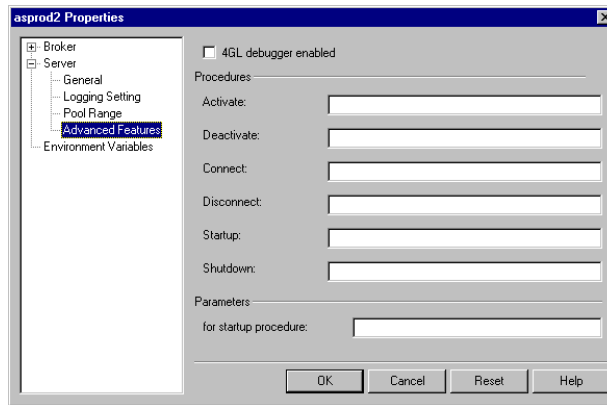
- 5 Select Server and use the following table to complete the server fields.

Field	Description
Server executable file	Accept the default in this field.
Server startup parameters	Enter the <code>-pf</code> parameter followed by the path and name of the NetUI parameter file. See “Creating a NetUI Parameter File” on page 58.

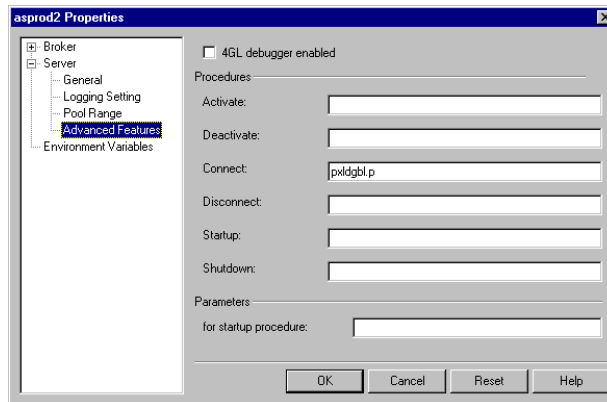
Field	Description
PROPATH	Enter the Progress search path (PROPATH) for this broker's processes. The PROPATH must contain the following directories: • The current working directory (.). • The directory containing the MFG/PRO character client code. • The bbi subdirectory below the language-code directory of the language you are implementing. The bbi subdirectory is located below the MFG/PRO character client code directory. • Any directories containing custom code that you want NetUI clients to use.
Minimum port number	Enter the lowest port this broker's processes can use.
Maximum port number	Enter the highest port this broker's processes can use.
Server application mode	Accept the default, Development. This mode provides enhanced troubleshooting capabilities, which you can use for initial installation and tuning. Once you have configured and tuned the NetUI environment to your satisfaction, select Production to reduce the number of processes each NetUI client session uses. For more information, see the Progress <i>Building Distributed Applications Using the Progress AppServer</i> documentation.



- 6 In the left-hand pane of the Properties window, expand the Server options and select Advanced Features.



- 7 In the Connect field, enter `pxldgbl.p`. This program loads the run-time parameters required by the XUI.



- 8 When ready, click OK in the Properties window to complete the auxiliary AppServer setup.

Verifying

To verify that the auxiliary AppServer instance is set up correctly, you should start it and check its status. Use the commands in Table 4.4 on page 77—replacing the primary AppServer broker name with the auxiliary AppServer broker name—to verify the auxiliary AppServer.

Set Up on UNIX

This section provides instructions for setting up WebSpeed server and AppServer instances on a UNIX server.

In the following instructions, you modify the `ubroker.properties` configuration file. This file is located in the `properties` subdirectory of your Progress installation directory. Review the comments sections of the `ubroker.properties` file for details on the entries in the file and how they affect your environment.

Set Up the WebSpeed Server

Use these instructions to set up a WebSpeed server instance. At least one Webspeed server instance is required for the NetUI.

Setting Up

- 1 Change directories to the `properties` subdirectory of your Progress installation directory. In the following example, *Progress* refers to the directory containing the Progress software.

```
cd Progress/properties
```

- 2 Make a backup copy of the `ubroker.properties` file and name it `ubroker.orig`.

```
cp ubroker.properties ubroker.orig
```

- 3 Open the `ubroker.properties` file in a text editor:

```
vi ubroker.properties
```

```
#####
##          Copyright (c) 1998 by Progress Software Corporation      ##
##          ##                                                     ##
## All rights reserved. No part of this program or document may be  ##
## reproduced in any form or by any means without permission in writing ##
## from Progress Software Corporation.                               ##
##          ##                                                     ##
#####
# WebSpeed/AppServer/DataServer properties definition file for UNIX
#
# ----- ubroker.properties -----
#
# Property File for the Unified Broker and NameServer
#
#
# This definition file includes the properties that you can define in
# a Unified Broker and NameServer configuration. It defines and configures
# four Brokers, one serving a WebSpeed Transaction Server, one serving
# an AppServer, one serving an Oracle DataServer, and one serving an
# ODBC DataServer, and a NameServer.
#
```

- 4 Locate the Sample Definitions section in the `ubrokers.properties` file.

```
#####
## Sample Definitions                                             ##
##          ##                                                  ##
## The following sections provide sample property settings that  ##
## you can use as templates for your own definitions.           ##
##          ##                                                  ##
#####
```

- 5 Copy the sample WebSpeed broker definition and make it the first entry after the configuration comments section.

Figure 4.1 is the sample WebSpeed broker definition.

Fig. 4.1
Sample WebSpeed
Broker

```
# Sample WebSpeed Broker definition
#
[UBroker.WS.wsbroker1]
  srvrLogFile=$WRKDIR/wsbroker1.server.log
  brokerLogFile=$WRKDIR/wsbroker1.broker.log
  portNumber=3055
  userName=
  groupName=
  appserviceNameList=wsbroker1
  controllingNameServer=NS1
  environment=wsbroker1
  uuid=932.99.000.ZZZ:1ee77e:cf3bbe04fd:-8000
  description=A sample WebSpeed Transaction Server
```

Figure 4.2 is the beginning of the configuration comments section in the `ubroker.properties` file. You should review the comments to become familiar with the different WebSpeed entries.

```
#####
##
## The following comments document in detail how to use this configuration ##
## file and how to set the various properties for the NameServer, Unified ##
## Brokers, WebSpeed Transaction Servers, and AppServers. ##
## ##
## ##
#####
```

Fig. 4.2
Configuration
Comments

Figure 4.3 shows the copied sample broker definition as the first entry after the comments section.

```
# Specifies whether the Admin Server automatically starts the
# NameServer when the Admin Server is started.
# Possible values: 0 - no auto start; 1 - auto start
#
# Default: 0
#
#####
# Sample WebSpeed Broker definition
#
[UBroker.WS.wsbroker1]
  srvrLogFile=$WRKDIR/wsbroker1.server.log
  brokerLogFile=$WRKDIR/wsbroker1.broker.log
  portNumber=3055
  userName=
  groupName=
  appserviceNameList=wsbroker1
  controllingNameServer=NS1
  environment=wsbroker1
  uuid=932.99.000.ZZZ:1ee77e:cf3bbe04fd:-8000
  description=A sample WebSpeed Transaction Server
```

Fig. 4.3
Copied Sample
Broker Definition

- 6 Change the comment line to indicate that this definition is for a NetUI WebSpeed broker:

```
# NetUI WebSpeed Broker for Production
```

- 7 Change the value after the second period in the `[UBroker.WS.wsbroker1]` entry to give the broker a meaningful name:

```
[UBroker.WS.wsprod]
```

Record the broker name in Table 12.2 on page 222. You must supply this name during the NetUI software installation.

- 8 Use the following table to configure the default entries in the sample broker definition.

Entry	Description
srvrLogFile=	Enter the path and name of a log file for the broker's agents; for example: <code>/qad/mfgpro/logs/wsprod.server.log</code>
brokerLogFile=	Enter the path and name of a log file for the broker; for example: <code>/qad/mfgpro/logs/wsprod.broker.log</code>
portNumber=	Enter an available port with which the broker can communicate. Record the broker port in Table 12.2 on page 222. You must supply this port during the NetUI software installation.
userName=	Leave this entry blank.
groupName=	Leave this entry blank.
appserviceNameList=	Enter the name you gave this broker; for example, <code>wsprod</code> .
controllingNameServer=	Accept the default for this entry.
environment=	Enter the name you gave this broker; for example, <code>wsprod</code> .
uuid=	Accept the default for this entry. In step 11, you will generate a new universal unique identifier (UUID) for the broker.
description=	Enter a description of the broker if you want.

- 9 Add an entry for `srvrStartupParam=` to the broker definition. This entry defines the startup parameters for the agents this broker controls. You must include the following elements in the startup parameters entry. Table 4.5 contains definitions of each of the elements.

```
srvrStartupParam= -p web/objects/web-disp.p
                  -pf /Path/Production.pf
```

Table 4.5
Agent Startup
Parameters

Element	Description
-p web/objects/web-disp.p	The WebSpeed broker program.
-pf /Path/Production.pf	Enter the path and name of the NetUI parameter file. For more information, see “Creating a NetUI Parameter File” on page 58.

- 10** Add the entries described in the following table to your broker definition.

Entry	Description
svrMinPort=	Enter the lowest port number for the agents this broker controls.
svrMaxPort=	Enter the highest port number for the agents this broker controls.
PROPATH=	Enter the Progress search path (PROPATH) you want the agents to use. You must include the following directories: • The current working directory (.) • The directory containing the MFG/PRO character client code. • The bbi subdirectory below the language-code directory of the language you are implementing. The language-code subdirectory is located below the MFG/PRO character client code directory. • You can add directories for custom code. Example PROPATH= . : /qad/eB/ : /qad/mfgpro/eB/us/bbi

- 11** Generate a new universal unique identifier (UUID) for the broker using the following command. In this command, *Progress* refers to the directory containing the Progress software.

```
Progress/bin/genuuid
```

- 12** When the UUID is generated, paste it into the `uuid=` entry in your broker definition.

Verifying

To verify that the WebSpeed server instance is set up correctly, you should start it and check its status. Use the commands in Table 4.6 to verify your WebSpeed server instance. In Table 4.6, *Progress* refers to the Progress installation directory and *BrokerName* refers to the name of the broker for your WebSpeed server instance.

♦ For additional commands, see the Progress *WebSpeed Installation and Configuration Guide*.

Table 4.6Broker Start, Stop,
and Status
Commands

Action	Command
Start Broker	<i>Progress/bin/wtbman -name BrokerName -start</i>
Stop Broker	<i>Progress/bin/wtbman -name BrokerName -stop</i>
Check Broker Status	<i>Progress/bin/wtbman -name BrokerName -query</i>

- 1 Check to see if the AdminServer is running using the following command:

```
Progress/bin/proadsv -query
```

If the AdminServer is not already running, start it using the following command:

```
Progress/bin/proadsv -start
```
- 2 Start the NameServer process for your broker. These instructions use the default NameServer, NS1:

```
nsman -name NS1 -start
```

You can check the status of the NameServer using the following command:

```
nsman -name NS1 -query
```
- 3 Start the MFG/PRO database servers for the Production database set. To start the database servers, use the database startup script for the Production database set; for example, `start.Production`.

When you start a WebSpeed broker, the broker's agents attempt to connect to the databases specified in the NetUI .pf file. This connection fails if the servers for the databases are not running.

Important Always start the appropriate MFG/PRO database servers before starting WebSpeed server or AppServer brokers. Additionally, if you need to stop the WebSpeed or AppServer brokers, you should stop the associated database servers.
- 4 Use the startup command in Table 4.6 to start the WebSpeed server instance.

- 5 Use the query command in Table 4.6 to check the status of the WebSpeed server instance.
- 6 After you check the status, leave the instance running.

Note In addition to the `wtbman` management utility, you may also need to refer to the log files for the WebSpeed server instance. These files are located in the directory you specified in the agent and broker log file entries (`srvrLogFile=` and `brokerLogFile=` respectively).

Set Up the Primary AppServer

Use these instructions to set up the primary AppServer instance. You must complete this setup to use XUI-enabled programs.

Setting Up

- 1 Change directories to the `properties` subdirectory of your Progress installation directory. In following example, *Progress* refers to the directory containing the Progress software:
`cd Progress/properties`
- 2 Open the `ubroker.properties` file in a text editor:
`vi ubroker.properties`
- 3 Locate the `Sample Definitions` section in the `ubrokers.properties` file.
- 4 Copy the sample AppServer broker definition and make it the next entry after the WebSpeed broker definition.

Figure 4.4 is the sample AppServer broker definition.

```
#
# Sample AppServer Broker definition
#
[UBroker.AS.asbroker1]
  srvrLogFile=$WRKDIR/asbroker1.server.log
  brokerLogFile=$WRKDIR/asbroker1.broker.log
  portNumber=3090
  userName=
  groupName=
  defaultService=1
  appserviceNameList=asbroker1
  controllingNameServer=NS1
  environment=asbroker1
  uuid=932.99.999.XXX:1ee77e:cf3bbe3d33:-8000
  description=A sample AppServer setup for State-reset
```

Fig. 4.4
Configuration
Comments

Figure 4.5 shows the copied sample broker definition as the second entry after the comments section.

Fig. 4.5
Copied Sample
Broker Definition

```

userName=
groupName=
appserviceNameList=wsprod
controllingNameServer=NS1
environment=wsprod
uuid=167.3.22.46:1dacd46a:d9f59832ad:-8000
description=The Production WebSpeed Broker Definition

#
# Sample AppServer Broker definition
#
[UBroker.AS.asbroker1]
  svrLogFile=$WRKDIR/asbroker1.server.log
  brokerLogFile=$WRKDIR/asbroker1.broker.log
  portNumber=3090
  userName=
  groupName=
  defaultService=1
  appserviceNameList=asbroker1
  controllingNameServer=NS1
  environment=asbroker1
  uuid=932.99.999.XXX:1ee77e:cf3bbe3d33:-8000
  description=A sample AppServer setup for State-reset

```

- 5 Change the comment line to indicate that this definition is for the primary AppServer broker:
NetUI Primary AppServer Broker for Production
- 6 Change the value after the second period in the [UBroker.AS.asbroker1] entry to give the broker a meaningful name:

```
[UBroker.AS.asprod1]
```

Record the broker name in Table 12.2 on page 222. You must supply this name when you configure the NetUI constants.

- 7 Use the following table to configure the default entries in the sample broker definition.

Entry	Description
svrLogFile=	Enter the path and name of a log file for the broker's server processes; for example: <code>/qad/mfgpro/logs/asprod1.server.log</code>
brokerLogFile=	Enter the path and name of a log file for the broker; for example: <code>/qad/mfgpro/logs/asprod1.broker.log</code>
portNumber=	Enter an available port with which the broker can communicate. Record the broker port in Table 12.2 on page 222.
userName=	Leave this entry blank.
groupName=	Leave this entry blank.
defaultService=	Accept the default for this entry.
appserviceNameList=	Enter the name you gave this broker; for example, <code>asprod1</code> .
controllingNameServer=	Accept the default for this entry
environment=	Enter the name you gave this broker; for example, <code>asprod1</code> .
uuid=	Accept the default for this entry. In step 12, you will generate a new UUID for this broker.
description=	Enter a description of the broker if you want.

- 8 Add a `svrStartupParam=` entry to the broker definition. This entry defines the startup parameters for the application server processes this broker controls.

- 9 In the `svrStartupParam` entry, add the `-pf` parameter and specify the path and name of the NetUI parameter file:

```
svrStartupParam= -pf /Path/Production.pf
```

◆ See “Creating a NetUI Parameter File” on page 58.

- 10 Add a `svrConnectProc=` entry to the broker definition. In this entry, specify `pxldgbl.p`. This program loads the run-time parameters required by the XUI:

```
svrConnectProc=pxldgbl.p
```

- 11** Add the entries described in the following table to your broker definition.

Entry	Description
svrMinPort=	Enter the lowest port number for the application server processes this broker controls.
svrMaxPort=	Enter the highest port number for the application server processes this broker controls.
PROPATH=	Enter the Progress search path (PROPATH) you want the application server processes to use. You must include the following directories: • The current working directory (.). • The directory containing the MFG/PRO character client code. • The bbi subdirectory below the language-code directory of the language you are implementing. The language-code subdirectory is located below the MFG/PRO character client code directory. • You can add directories for custom code. Example PROPATH= . : /qad/eB/ : /qad/eB/us/bbi

- 12** Generate a new UUID for the broker using the following command. In this command, *Progress* refers to the directory containing the Progress software:

```
Progress/bin/genuuid
```

- 13** When the UUID is generated, paste it into the `uuid=` entry in your broker definition.

Verifying

To verify that the primary AppServer instance is set up correctly, you should start it and check its status. Use the commands in Table 4.7 to verify an AppServer instance. In Table 4.7, *Progress* refers to the Progress installation directory and *BrokerName* refers to the name of the broker for the AppServer instance.

For additional commands, see the *Progress Building Distributed Applications Using the Progress AppServer* documentation.

Action	Command
Start Broker	<i>Progress/bin/asbman -name BrokerName -start</i>
Stop Broker	<i>Progress/bin/asbman -name BrokerName -stop</i>
Check Broker Status	<i>Progress/bin/asbman -name BrokerName -query</i>

Table 4.7
Broker Start, Stop,
and Status
Commands

- 1

Verify that the AdminServer is running using the following command.

Progress/bin/proadsv -query
- 2

Verify that the NameServer process for your broker is running using the following command. These instructions use the default NameServer, NS1:

nsman -name NS1 -query
- 3

Verify that the MFG/PRO database servers for the Production database set are running.
- 4

Use the startup command in Table 4.7 to start the primary AppServer instance.
- 5

Use the query command in Table 4.7 to check the status of the primary AppServer instance.
- 6

After you check the status, leave the instance running.

Tip
If either the AdminServer or NameServer are not running, use the instructions in “Verifying” on page 87 to start them.

Tip
If the database servers are not running, use the instructions in “Verifying” on page 87 to start them.

Note In addition to the asbman management utility, you may also need to refer to the log files for an AppServer instance. These files are located in the directory you specified in the server process and broker log file entries (srvrLogFile= and brokerLogFile= respectively).

Set Up the Auxiliary AppServer

Use these instructions to set up the auxiliary AppServer instance. You must complete this setup to use XUI-enabled programs.

Setting Up

- 1

Change directories to the `properties` subdirectory of your Progress installation directory. In following example, *Progress* refers to the directory containing the Progress software:

cd Progress/properties

- 2 Open the `ubroker.properties` file in a text editor:

```
vi ubroker.properties
```
- 3 Copy the primary AppServer broker definition and make it the next entry.
- 4 Change the comment line to indicate that this definition is for the auxiliary AppServer broker:

```
# NetUI Auxiliary AppServer Broker for Production
```
- 5 Specify a broker name for the auxiliary AppServer instance. Record the broker name in Table 12.2 on page 222:

```
[UBroker.AS.asprod2]
```
- 6 Use the following table to configure the entries in the broker definition.

Entry	Description
<code>svrLogFile=</code>	Enter the path and name of a log file for this broker's agents; for example: <code>/qad/mfgpro/logs/asprod2.server.log</code>
<code>brokerLogFile=</code>	Enter the path and name of a log file for this broker; for example: <code>/qad/mfgpro/logs/asprod2.broker.log</code>
<code>portNumber=</code>	Enter an available port with which the broker can communicate. Record the broker port in Table 12.2 on page 222.
<code>userName=</code>	Leave this entry blank.
<code>groupName=</code>	Leave this entry blank.
<code>defaultService=</code>	Accept the default for this entry.
<code>appserviceNameList=</code>	Enter the name you gave this broker; for example, <code>asprod2</code> .
<code>controllingNameServer=</code>	Accept the default for this entry
<code>environment=</code>	Enter the name you gave this broker; for example, <code>asprod2</code> .
<code>uuid=</code>	Accept the default for this entry. In a subsequent step, you will generate a new UUID for this broker.
<code>description=</code>	Enter a description of the broker if you want.
<code>svrStartupParam=</code>	Use the parameters you specified during the primary AppServer setup.

Entry	Description
svrConnectProc=	Use the parameter you specified during the primary AppServer setup.
svrMinPort=	Enter the lowest port number for the application server processes this broker controls.
svrMaxPort=	Enter the highest port number for the application server processes this broker controls.
PROPATH=	Use the PROPATH you specified during the primary AppServer setup.

- 7 Generate a new universal unique identifier (UUID) for the broker using the following command. In this command, *Progress* refers to the directory containing the Progress software:

Progress/bin/genuuid

- 8 When the UUID is generated, paste it into the uuid= entry in your broker definition.

Verifying

To verify that the auxiliary AppServer instance is set up correctly, you should start it and check its status while running. Use the commands in Table 4.7 on page 93—replacing the primary AppServer broker name with the auxiliary AppServer broker name—to verify the auxiliary AppServer instance.

CHAPTER 5

Installing the NetUI Software on UNIX

This chapter provides instructions for installing the NetUI files on a UNIX Web server.

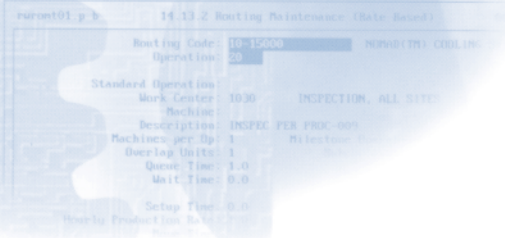
Installation Overview **98**

Installing the Language Files **98**

Installing the Server Files **101**

Installing the NetUI Service Pack **107**

Items Installed **107**



Installation Overview

Installing the NetUI files consists of the following major tasks:

- Install the language-specific NetUI files from the MFG/PRO language file media. You *must* load the language-specific NetUI files first.
- Install the NetUI Web server files from the MFG/PRO NetUI media.
- If a service pack is included in your NetUI product package, install the service pack in your NetUI environment.

Installing the Language Files

Complete these steps to install the NetUI language files on your UNIX Web server.

The NetUI language files are located on the MFG/PRO language file media that you received with your MFG/PRO eB product package. This is the same media you used to install the MFG/PRO language-specific files on your MFG/PRO database server.

Important You must load the NetUI language files before you load the NetUI Web server files.

Mount the CD-ROM

Follow these steps to mount the MFG/PRO language file CD-ROM.

- 1 Log on as the root user ID.
- 2 Create a CD-ROM directory if one does not already exist. The installation process assumes the directory name is `/cdrom`.
- 3 Place the MFG/PRO Language File CD-ROM into the CD-ROM drive.

- 4 Mount the CD-ROM. The mount command differs from system to system. Listed below are sample commands for mounting the NetUI CD-ROM.

Hardware	Mount Command
Sun	<code>volcheck cdrom</code>
HP	<code>mount /dev/dsk/<i>YourCDDevice</i> /cdrom</code>
Digital	<code>mount -r -o noversion -t cdcs /dev/<i>YourCDDevice</i> /cdrom</code>
All others	Refer to your hardware system documentation or vendor for requirements to mount a CD-ROM. You may be able to type <code>man mount</code> to determine the correct command.

Run the Installation Script

- 1 Change to the `netui` subdirectory below the CD-ROM directory.

```
cd /cdrom/netui
```

- 2 Change to the `install/unix` subdirectory below the `netui` directory.

```
cd /install/unix
```

- 3 Launch the `install` script.

```
./install
```

- 4 Read the MFG/PRO NetUI software license agreement and at the following prompt, accept the agreement to continue with the installation.

```
Do you accept all the terms of the preceding License
Agreement? If you choose No, The Install will stop. To
install MFG/PRO eB NetUI Language File, you must accept
this agreement. (y/n) [y]
```

- 5 Enter your terminal type or accept the default. The default is based on the setting of the \$TERM environment variable.

```

      qqqqqqqqqq      aaaa      dddddddddd
      qqqqqqqqqqqqqq      aaaaaaaa      dddddddddd
      qq      qq      aaa      aaa      ddd      ddd
      qq      qq      aaa      aaa      ddd      ddd
      qq      qq      aaa      aaa      ddd      ddd
      qq      qq      aaaaaaaaaaaaaa      ddd      ddd
      qq      qq      aaaaaaaaaaaaaa      ddd      ddd
      qqqqqqqqqqqqqq      aaaa      aaaa      dddddddddd
      qqqqqqqqqqqq      aaaa      aaaa      dddddddddd
      qq
    
```

MFG/PRO eB NetUI Language File Installation Script

Enter Your terminal type
[default = dtterm]: █

See Table 12.2 on page 222.

- 6 Enter your Web server's document directory. The NetUI installation process creates a `multinet` subdirectory below the directory you specify. The language-specific NetUI files are copied into this subdirectory.

```

      qqqqqqqqqq      aaaa      dddddddddd
      qqqqqqqqqqqqqq      aaaaaaaa      dddddddddd
      qq      qq      aaa      aaa      ddd      ddd
      qq      qq      aaa      aaa      ddd      ddd
      qq      qq      aaa      aaa      ddd      ddd
      qq      qq      aaaaaaaaaaaaaa      ddd      ddd
      qq      qq      aaaaaaaaaaaaaa      ddd      ddd
      qqqqqqqqqqqqqq      aaaa      aaaa      dddddddddd
      qqqqqqqqqqqq      aaaa      aaaa      dddddddddd
      qq
    
```

MFG/PRO eB NetUI Language File Installation Script

Please Specify the Target Directory:

Note Before beginning the copy, the installation script checks the available space in the document directory. If there is not enough space, the installation stops; obtain adequate disk space and restart the installation.

- 7 At the prompt, specify Yes to clean up the temporary directories used by the installation script.

- 5 Enter the path to the directory containing the Progress WebSpeed software or accept the default.

Note When you perform a complete Progress installation, the WebSpeed components are installed in the Progress installation directory. For this reason, the installation script prompts for the Progress/WebSpeed directory.

```

      qqqqqqqqqq      aaaa      dddddddddd
      qqqqqqqqqqqqqq      aaaaaaaa      dddddddddd
      qq      qq      aaa      aaa      ddd      ddd
      qq      qq      aaa      aaa      ddd      ddd
      qq      qq      aaa      aaa      ddd      ddd
      qq      qq      aaaaaaaaaaaaaa      ddd      ddd
      qq      qq      aaaaaaaaaaaaaa      ddd      ddd
      qqqqqqqqqqqqqq      aaaa      aaaa      dddddddddd
      qqqqqqqqqqqq      aaaa      aaaa      dddddddd
      qq
MFG/PRO eB NetUI Web Server Installation Script ....

Enter Progress/WebSpeed installation directory [/dr01/progress/d1c90b19]:
```

See Table 12.2 on page 222.

- 6 Enter your Web server’s document directory. The NetUI Java, HTML, and client setup files will be installed in this directory. The NetUI installation process copies the HTML, Java, and client setup files to the `multinet` subdirectory below the directory you specify.

```

      qqqqqqqqqq      aaaa      dddddddddd
      qqqqqqqqqqqqqq      aaaaaaaa      dddddddddd
      qq      qq      aaa      aaa      ddd      ddd
      qq      qq      aaa      aaa      ddd      ddd
      qq      qq      aaa      aaa      ddd      ddd
      qq      qq      aaaaaaaaaaaaaa      ddd      ddd
      qq      qq      aaaaaaaaaaaaaa      ddd      ddd
      qqqqqqqqqqqqqq      aaaa      aaaa      dddddddddd
      qqqqqqqqqqqq      aaaa      aaaa      dddddddd
      qq
MFG/PRO eB NetUI Web Server Installation Script ....

Enter the Webserver Document directory where the Java, HTML and
Client Setup files will be installed []:
```


See Table 12.2 on page 222.

- 9 Enter your Web server's executables directory. This directory was set up during the Web server installation process.

Additionally, the WebSpeed messenger executable should have been installed in this directory during the WebSpeed installation process. The NetUI Web Server Installation Script verifies that the default messenger executable is installed in the directory you specify.

```

          qqqqqqqqqq      aaaa      dddddddddd
        qqqqqqqqqqqqqqqq      aaaaaaaa      dddddddddd
      qqq      qqq      aaa      aaa      ddd      ddd
      qqq      qqq      aaa      aaa      ddd      ddd
      qqq      qqq      aaa      aaa      ddd      ddd
      qqq      qqq      aaaaaaaaaaaaaa      ddd      ddd
      qqq      qqq      aaaaaaaaaaaaaa      ddd      ddd
      qqqqqqqqqqqqqqqq      aaaa      aaaa      dddddddddd
      qqqqqqqqqqqqqq      aaaa      aaaa      dddddddddd
          qqq

```

MFG/PRO eB NetUI Web Server Installation Script

Enter the Webserver CGI Scripts Directory []:

Note If the NetUI Web Server Installation Script does not detect the default messenger executable in the directory you specify, the following warning displays:

Default Webspeed Messenger not installed. Verify that this directory is configured in the Webserver as a CGI Scripts directory.

Either install the default WebSpeed messenger executable (cgiip.exe) in your Web server's executables directory, or verify that you are using a different messenger executable. For information on the messengers supplied with the WebSpeed product, see the *Progress WebSpeed Installation and Configuration Guide*.

When ready, press Enter to continue.

- 10 Enter the name of the WebSpeed broker you set up for the NetUI.

See Table 12.2 on page 222.

<pre> qqqqqqqqqq aaaa dddddddddd qqqqqqqqqqqqqq aaaaaaaa dddddddddd qq qq aaa aaa ddd ddd qq qq aaa aaa ddd ddd qq qq aaa aaa ddd ddd qq qq aaaaaaaaaaaaaa ddd ddd qq qq aaaaaaaaaaaaaa ddd ddd qq qq aaaaa aaaa dddddddddd qq qq aaaa aaaa dddddddddd qq </pre>
MFG/PRO eB NetUI Web Server Installation Script

Enter the Webspeed broker name [netui]:

- 11 At the following prompt, specify the working directory you want the NetUI processes to use on the Web server.

<pre> qqqqqqqqqq aaaa dddddddddd qqqqqqqqqqqqqq aaaaaaaa dddddddddd qq qq aaa aaa ddd ddd qq qq aaa aaa ddd ddd qq qq aaaaaaaaaaaaaa ddd ddd qq qq aaaaaaaaaaaaaa ddd ddd qq qq aaaaa aaaa dddddddddd qq qq aaaa aaaa dddddddddd qq </pre>
MFG/PRO eB NetUI Web Server Installation Script

Enter the Working Directory [/tmp]:

- 12** At the confirmation prompt, use Table 12.2 on page 222 to verify the installation information.

If all of the installation information is correct, specify Yes and press Enter.

If any of the installation information is incorrect, specify No and press Enter. The Progress installation directory prompt displays. As you proceed through the prompts, your previous entries display as the defaults. Press Enter to accept a default or change the entry if it is incorrect.

```
CONFIRMATION
Progress/WebSpeed Directory : /dr01/progress/d1c90b19
Progress with WebSpeed Version : 9.0B19
Document Directory : /users/cwp/work91/netui/docs
HTTP Port Number : 80
URL Prefix : cgi-bin
CGI Directory : /users/cwp/work91/netui/scripts
Webspeed Broker : netui
Working Directory : /tmp

Is this information correct (y/n) [y]:
```

- 13** At the expand media prompt, specify Yes.

```

      qqqqqqqqqq      aaaa      dddddddddd
    qqqqqqqqqqqqqqqq      aaaaaaaa      dddddddddd
    qq      qq      aaa      aaaa      ddd      ddd
    qq      qq      aaa      aaaa      ddd      ddd
    qq      qq      aaa      aaaa      ddd      ddd
    qq      qq      aaaaaaaaaaaaaa      ddd      ddd
    qq      qq      aaaaaaaaaaaaaa      ddd      ddd
    qqqqqqqqqqqqqqqq      aaaa      aaaa      dddddddddd
    qqqqqqqqqqqqqq      aaaa      aaaa      dddddddddd
      qq
MFG/PRO eB NetUI Web Server Installation Script ....

Expand the media (y/n) [y]?
```

Installing the NetUI Service Pack

If a service pack is included in your NetUI product package, use the instructions on the service pack CD-ROM to install the service pack in your NetUI environment. The service pack updates your NetUI Java code, HTML files, and client setup files.

Which service pack you are installing can also affect the version of Java required to use the NetUI.

◆ See page 10.

If the service pack includes updates to files on the RMI server, you must install these files from the service pack when you configure the RMI server.

◆ See step 3 in “Setting Up the RMI Server” on page 150.

Items Installed

This section provides a brief description of the directory structures created by the NetUI installation process. The installation creates a `multinet` subdirectory below the Web server’s document directory. This directory contains the NetUI Java, HTML, online help, and client files.

The following is an outline of the `multinet` directory structure. Files in bold are added with Service Pack 2.

YourWebServer/DocDir/multinet

- `client_files <dir>` (NetUI client directory)
 - `client.js`
 - `client_setup.html`
 - `client_update.html`
 - `identitydb.obj`
 - `labels.js`
 - **`netui13.js`**
 - `qad_version.js`
 - `version.js`
 - `win32 <dir>`
 - `eBEnvironmentSetup.exe`
 - `eBLocalSetup.exe`
 - **`j2re-1_3_0_02-win-i-exe`**
 - **`QAD_eB_SP2.exe`**
 - `QADNetUIeBSetup.exe`
- `default.htm`
- `embedded.html`
- **`embedded13.html`**
- `example.html`
- `full.html`
- **`full13.html`**
- `index.html`
- `multinet.html`
- **`multinet13.html`**
- `qadbg.jpg`
- `qadlogo.gif`
- `title.html`
- `trace.html`
- **`trace13.html`**
- `version.html`
- `examples <dir>`
- `java <dir>`
 - `gnu <dir>` (used for MFG/PRO telnet maintenance programs)
 - `images <dir>` (user-defined image files)
 - `multinet.jar`
 - `o4glrt.jar`
 - `Ssl.cab`
 - `Ssl.jar`
- `help <dir>` (NetUI online help)
 - `cgi-bin <dir>` (Help script directory)
 - `doc <dir>` (Help HTML files)

CHAPTER 6

Installing the NetUI Software on Windows NT

This chapter provides instructions for installing the NetUI software on a Windows NT Web server.

Installation Overview **110**

Installing the Language Files **110**

Installing the Server Files **113**

Installing the NetUI Service Pack **117**

Items Installed **117**

param91.p 14.13.2 Routing Maintenance (Main Screen)

Routing Code:	10-15000	NONAD(75) CDR IN:
Operation:	20	
Standard Operation		
Work Center	1030	INSPECTION, ALL SITE
Machin:		
Description:	INSPEC PER PROX-000	
Machines per Op:	1	Milestone P
Overlap Units:	1	
Queue Time:	1.0	
Wait Time:	0.0	
Setup Time:	0.0	
Ready for Production:	0.0	

Buttons: [Back] [Forward] [Print] [Exit] [Check Master]

Installation Overview

Installing the NetUI files consists of the following major tasks:

- Install the language-specific NetUI files from the MFG/PRO language file media. You *must* load the language-specific NetUI files first.
- Install the NetUI Web server files from the MFG/PRO NetUI media.
- If a service pack is included with your NetUI product package, install the service pack in your NetUI environment.

Installing the Language Files

Complete these steps to install the NetUI language files on your Windows NT Web server.

The NetUI language files are located on the MFG/PRO language file media that you received with your MFG/PRO eB product package. This is the same media you used to load the MFG/PRO language-specific files on your MFG/PRO database server.

Important You must load the NetUI language files before you load the NetUI Web server files.

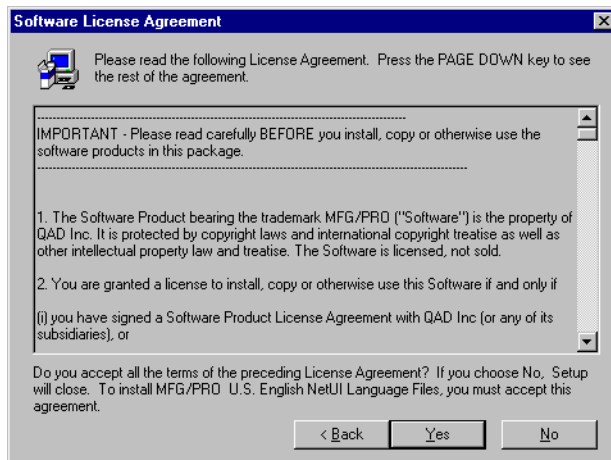
- 1 Place the MFG/PRO Language File CD-ROM into the CD-ROM drive.
- 2 Using File Manager, Windows Explorer, or the Start button, access the `netui` subdirectory and run the `setup` program:

`CD-ROMDrive:\netui\setup.exe`

- 3 At the Welcome window, click Next.



- 4 To continue with the installation, read the license agreement and click Yes to accept it.



See Table 12.2 on page 222.

- 5 In the Choose Destination Location window, specify your Web server's document directory. Click Next to continue.

The NetUI installation process creates a `multinet` subdirectory below the directory you specify. The language-specific NetUI files are copied into this subdirectory.



- 6 At the confirmation window, verify the installation information. Click Next to begin copying the NetUI language files, or click Back to make changes.



- 7 When the file copy completes, the Installation Complete window displays. Click Finish to exit the installation program.



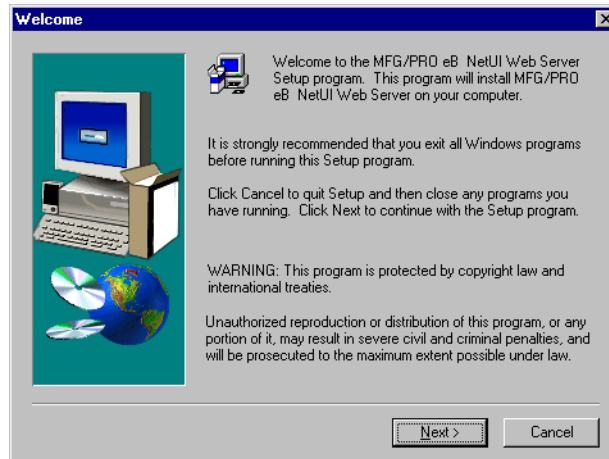
Installing the Server Files

Complete these steps to install the NetUI server files on your Windows NT Web server.

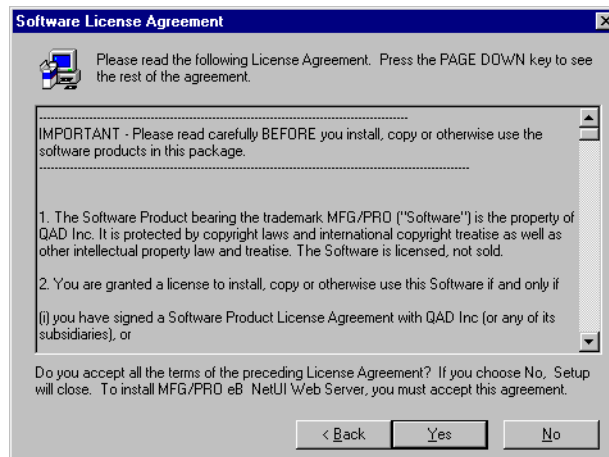
- 1 Place the NetUI CD-ROM into the CD-ROM drive.
- 2 Using File Manager, Windows Explorer, or the Start button, run the setup program:

CD-ROM Drive:\nt\setup

- 3 At the Welcome window, click Next.



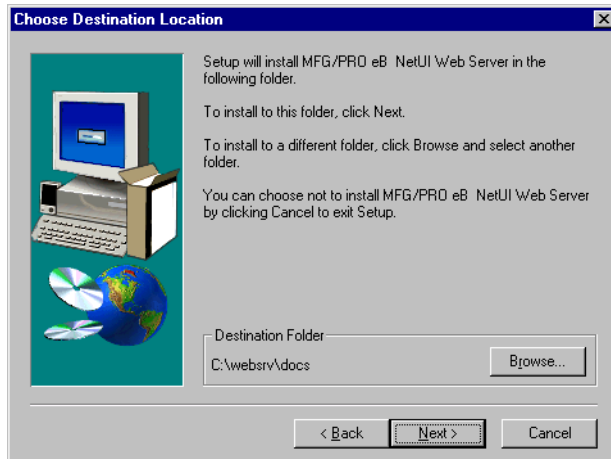
- 4 To continue with the installation, read the license agreement and click Yes to accept it.



- 5 Enter the Web server's document directory. Accept the default or use the Browse button to select an alternate directory. Click Next to continue.

See Table 12.2 on page 222.

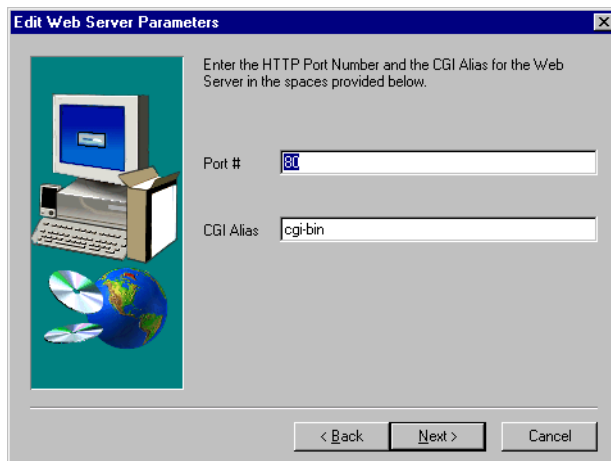
The NetUI installation process copies the NetUI HTML, Java, and client setup files to the `multinet` subdirectory below the directory you specify.



- 6 Enter your Web server's HTTP port number and the CGI alias assigned to your CGI directory. Click Next to continue.

The port number and CGI alias were assigned during the Web server installation process.

See Table 12.2 on page 222.

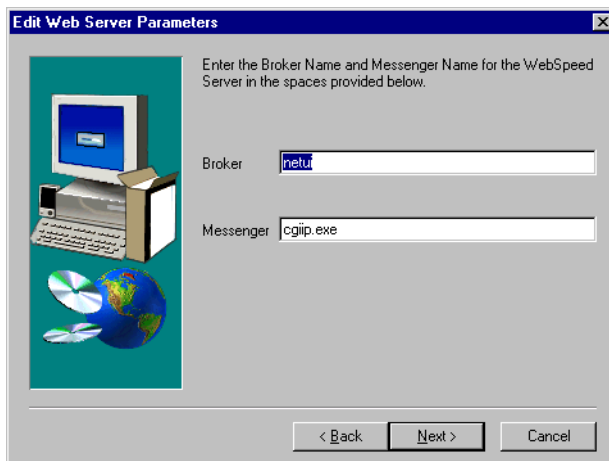


- 7 In the following window, enter the name of your WebSpeed broker and the WebSpeed messenger executable you want to use.

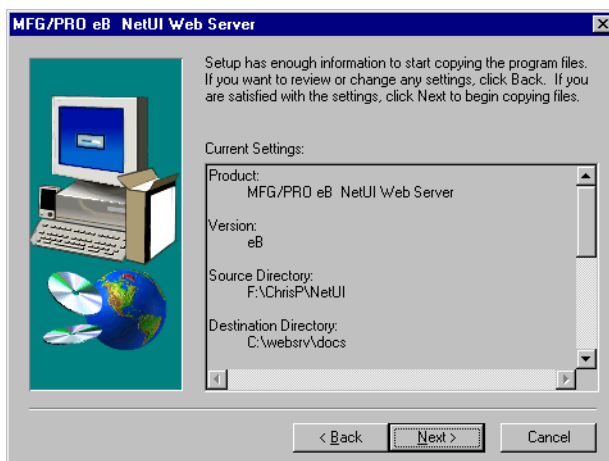
See Table 12.2 on page 222.

You created the WebSpeed broker using the instructions in “Set Up the WebSpeed Server” on page 68.

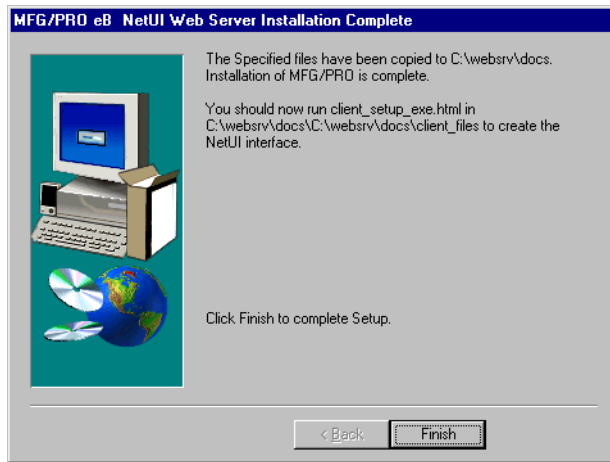
The WebSpeed messenger executables were installed during the WebSpeed installation process. For more information, see the *Progress WebSpeed Installation and Configuration Guide*.



- 8 At the confirmation window, use Table 12.2 on page 222 to verify the installation information. Click Next to begin copying the database server files, or click Back to make changes.



- 9 When the file copy completes, the Installation Complete window displays. Click Finish and exit the installation program.



Installing the NetUI Service Pack

If a service pack was included with your NetUI product package, use the instructions on the service pack CD-ROM to install the service pack in your NetUI environment. The service pack updates your NetUI Java code, HTML files, and client setup files.

Which service pack you are installing can also affect the version of Java required to use the NetUI.

◆ See page 10.

If the service pack includes updates to files on the RMI server, you must install these files from the service pack when you configure the RMI server.

◆ See step 3 in “Setting Up the RMI Server” on page 150.

Items Installed

This section provides a brief description of the directory structures created by the NetUI installation process. The installation creates a **multinet** subdirectory below the Web server’s document directory. This directory contains the NetUI Java, HTML, online help, and client files.

The following is an outline of the `multinet` directory structure. Files in bold are added with Service Pack 2.

YourWebServer\DocDir\multinet

- `client_files <dir>` (NetUI client directory)
 - `client.js`
 - `client_setup.html`
 - `client_update.html`
 - `identitydb.obj`
 - `labels.js`
 - **`netui13.js`**
 - `qad_version.js`
 - `version.js`
 - `win32 <dir>`
 - `eBEnvironmentSetup.exe`
 - `eBLocalSetup.exe`
 - **`j2re-1_3_0_02-win-i-exe`**
 - **`QAD_eB_SP2.exe`**
 - `QADNetUIeBSetup.exe`
- `default.htm`
- `embedded.html`
- **`embedded13.html`**
- `example.html`
- `full.html`
- **`full13.html`**
- `index.html`
- `multinet.html`
- **`multinet13.html`**
- `qadbg.jpg`
- `qadlogo.gif`
- `title.html`
- `trace.html`
- **`trace13.html`**
- `version.html`
- `examples <dir>`
- `java <dir>`
 - `gnu <dir>` (used for MFG/PRO telnet maintenance programs)
 - `images <dir>` (user-defined image files)
 - `multinet.jar`
 - `o4glrt.jar`
 - `Ssl.cab`
 - `Ssl.jar`
- `help <dir>` (NetUI online help)
 - `cgi-bin <dir>` (Help script directory)
 - `doc <dir>` (Help HTML files)

CHAPTER 7

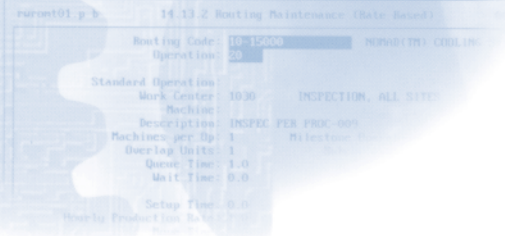
Configuring Telnet on UNIX

This chapter provides instructions for configuring the NetUI environment to use a UNIX telnet server.

Configuration Overview **120**

Configuring the Telnet Environment **120**

Configuring Telnet Options in MFG/PRO **123**



Configuration Overview

► For more information, see “Maintenance and Transaction Components” on page 20.

There are two ways to access maintenance and transaction programs: a telnet interface and a full GUI Java interface. To run telnet maintenance and transaction programs, the NetUI environment requires a telnet server.

To configure a UNIX telnet server, you must perform the following major tasks:

- Configure the telnet environment.
- Configure the telnet settings in MFG/PRO.

Important These instructions assume that you are using the same machine for both the MFG/PRO database server and NetUI telnet server. This ensures that the MFG/PRO character client code required for telnet sessions is located on the telnet server.

Configuring the Telnet Environment

To access MFG/PRO maintenance programs, the NetUI uses telnet to log on to the server where MFG/PRO is installed. Because the NetUI communicates via HTTP, the account log-in ID and password are sent using unencrypted text. Since this may compromise system security, you should configure the telnet environment with special server-side security measures in mind.

A range of security options exists to solve the unencrypted log in and password problem. This section outlines two sample security setups: one providing a maximum level of security and one providing less security but more flexibility for Progress client session and home directory access. In both setups, it is recommended that you use a restricted shell (rsh).

Restricted Shells

Restricted shells are restricted versions of the common UNIX Bourne shell or Korn shell. In the Bourne shell, the restricted shell is run as rsh (/usr/lib/rsh), while in the Korn Shell it is known as rksh (/usr/bin/rksh). The restricted versions of these shells allow users to log in with restricted access. They cannot:

- Use the cd command to change directories
- Specify a path or command using /
- Use redirection (>, >>)
- Set the value of \$PATH

Note A user's path should not include /usr/bin. This lets the user run another shell, thereby inheriting access to any commands that the child shell allows.

Tip
The default shell for a user is located in the /etc/passwd file.

Examples of Security Measures

Case 1: Maximum Security

One UNIX account with the following characteristics is used for all NetUI telnet sessions:

- No write permissions to home directory. Temporary files are written elsewhere.
- \$PATH, \$DLC, and \$PROPATH environment variables are set in .profile and inaccessible to the user.
- Startup command and/or scripts run from .profile.
- Telnet disconnects immediately after the user exits the MFG/PRO session.

Use the following instructions to set up Case 1:

- 1 Create the unique NetUI account for log in to UNIX through telnet.
- 2 Make the default shell for this account the restricted shell.
- 3 Remove all write permissions for this user in their home directory. Use the -T option in the remote script to specify an alternate temporary directory.

- 4 Set up the `.profile` to set minimal environment variables.
- 5 Set up the `.profile` to run the script automatically.

Example `.profile` for Case 1:

```
/*Sample .profile for NetUI maintenance session, single NetUI login*/
#set default for error (STOP) condition handling
stty intr ^C

#set environment variables
PATH=/dlc91:/dlc91/bin
DLC=${DLC - /dlc91}
PROEXE=${PROEXE - $DLC/bin/_progres}
export PATH DLC PROEXE

#Autorun remote script for NetUI Access and automatically exit
exec remote.script
exit
```

Case 2: Less Security

Users have their own unique log in and password, but run the restricted shell by default:

- Write permission to directory is possible, but not necessary.
- `$PATH`, `$DLC`, and `$PROPATH` environment variables are set in `.profile` and inaccessible to the user.
- Users run a subset of UNIX commands, which you add to `/usr/rbin`.
- Users can run MFG/PRO manually from command line or script.

Use the following instructions to set up Case 2.

- 1 Create or modify accounts for NetUI users of MFG/PRO maintenance programs by changing their default shell in the `/etc/passwd` file to the restricted shell.
- 2 Create the directory `/usr/rbin` and copy the UNIX commands necessary for these users. Make the `/usr/bin` directory read-only so users cannot change path variables.
- 3 Set up a special `.profile` for the NetUI maintenance program users.
- 4 Set the minimal environment variables, remembering to include `/usr/rbin`.

- 5 Copy the NetUI telnet connection script to each user's home directory with read-only access.
- 6 Put any other necessary read-only script files in the home directory.

Example .profile for Case 2:

Sample .profile for NetUI maintenance session, for individual logins remote.script should be in home directory; executable by NetUI/

```
#set default for error (STOP) condition handling
stty intr ^C
```

```
#set environment variables
```

```
PATH=/dlc91:/dlc91/bin:/usr/rbin#don't forget /rbin directory
```

```
DLC=${DLC - /dlc91}
```

```
PROEXE=${PROEXE - $DLC/bin/_progres}
```

```
export PATH DLC PROEXE
```

Configuring Telnet Options in MFG/PRO

To use NetUI telnet programs, you must configure telnet settings in User Option Telnet Maintenance (36.20.10.3).

mgusrmt.p b+		36.20.10.3 User Option Telnet Maintenance		08/11/00	
User ID:					
Telnet Options					
Host:		Script Timeout:			
Host O/S: UNIX		Idle Timeout:			
Port:					
Image:					
Script Lines					
Sequence:					
Script Pattern:					
Script Value:					
Script Status:					
Telnet Connections					
Maximum:		Min Telnet Connecti:			
F1=Go 2=Help 3=Ins 4=End 6=Menu 7=Recall 8=Clear 9=Prev 10=Next					

Fig. 7.1
User Option Telnet
Maintenance
(36.20.10.3)

This program lets you define unique settings for individual users or enter an asterisk (*) in the User ID field to create a generic record of settings. This generic record is automatically associated with every user in the system that does not have a user-specific record. These instructions use the generic record.

Configure Telnet Server Settings

Use the following instructions to configure the telnet server settings:

- 1 Start an MFG/PRO session and access User Option Telnet Maintenance (36.20.10.3).
- 2 In the User ID field, enter * to specify the generic record and press Go.
- 3 Use the following table to complete the Telnet Options fields in User Option Telnet Maintenance.

mgusrmt.p b+		36.20.10.3 User Option Telnet Maintenance		08/11/00	
User ID: *					
Telnet Options					
Host: █		Script Timeout: 240			
Host O/S: UNIX		Idle Timeout: 240			
Port: 23					
Image: _____					

Field	Description
Host	Enter the machine name or IP address of the telnet server. The NetUI Java applet uses this information to establish the telnet connection.
Host O/S	Enter UNIX.
Port	Enter the port number for the telnet server.
Image	Optionally, enter the name of a .gif image file to use as the menu icon for telnet programs. The image file must be in graphics interchange format (extension .gif). When you enter the name of the file in this field, do not include the .gif extension. The image file must be located in the images subdirectory below the java subdirectory, which is in the multinet directory in the Web server's documents directory. The image size should be 24 by 24 pixels. If you leave this field blank, the standard QAD-supplied icon is used.

Field	Description
Script Timeout	Enter the number of seconds (1-999) the system will wait for the telnet log-in script to execute. If this value is exceeded, a time-out message displays and the session closes.
Idle Timeout	Enter the number of seconds (1-999) the system will wait after a telnet session was begun for an MFG/PRO program to execute.

- 4 When you are ready, press Go to access the Script Lines frame.

Configure Telnet Log-In Sequence

To log into and begin a telnet session on the telnet server, you must provide the NetUI Java applet with the telnet server log-in prompts and responses to these prompts. Taken together the prompts and responses make up the telnet log-in sequence.

You define the telnet log-in sequence in the Script Lines frame in User Option Telnet Maintenance.

Sequence: 1	Script Lines
Script Pattern: login: Script Value: netui Script Status: logging in	

Overview

Each line in the log-in sequence is stored in the MFG/PRO administration schema as a single `chui_det` record containing three values: pattern, value, and status. Each of these values is represented by a field in the Script Lines frame.

Table 7.1 provides a general description of the fields in the Script Lines frame.

Table 7.1
Script Lines Fields

Field	Description
Sequence	A number for the script line that indicates the order that the line appears in the overall log-in sequence.
Script Pattern	Enter the prompt generated by the telnet server when a telnet log-in occurs. Important The values you enter in the Script Pattern field must be identical to the prompts the telnet server displays when users attempt to log in.
Script Value	Enter the response to the telnet log-in prompt defined in Script Pattern.
Script Status	Optionally enter a description of the activity that the prompt and response represent. For example, if the prompt is <code>Login:</code> and the response is a user ID, enter <code>Logging In</code> to describe the activity. If an error occurs during the execution of the prompt and response, the description in the Script Status field display in the Java Plug-In Console on the NetUI client. You can use these descriptions as an aid in troubleshooting telnet session issues. Note For more information on the Java Plug-In Console, see “trace.html” on page 202.

Telnet Connection Script

To identify the MFG/PRO database that you want the NetUI telnet session to run against, you need a telnet connection script. On UNIX servers, MFG/UTIL generates telnet connection scripts for each database set for which you are generating scripts.

The telnet connection scripts are located in your MFG/PRO database server administration directory. These scripts use the following naming format: `qma.DBSetName` where *DBSetName* is the name of the database to which the script connects. For example the telnet connection script for the Production database set is `qma.Production`.

In general the connection script performs the following functions:

- Changes to the user’s home directory
- Provides connection parameters for the databases in the set

You can modify the telnet connection script to fit your environment.

You must specify a telnet connection script somewhere in your log-in sequence to connect the telnet session to the MFG/PRO databases.

Define the Log-In Sequence

The following instructions are for an example telnet log-in sequence. You must configure your log-in sequence specifically for your environment.

Review the following instructions to become familiar with the configuration process. Then create your log-in sequence based on your environment.

- 1 In the Sequence field, enter 1 and press Enter. Then use the following table to complete the Script Pattern, Script Value, and Script Status fields. When you are ready, press Go to save your entries.

Field	Description
Script Pattern	Enter the first prompt the server displays when you attempt to log into a telnet session. Example login:
Script Value	Enter the response you want the NetUI Java applet to supply at this prompt. For example, a user ID.
Script Status	Enter a description of the activity this prompt and response represent. Example Logging In

- 2 In the Sequence field, enter 2 and press Enter. Then use the following table to complete the Script Pattern, Script Value, and Script Status fields. When you are ready, press Go to save your entries.

Field	Description
Script Pattern	Enter the second prompt the server displays when you attempt to log into a telnet session. Example password:
Script Value	Enter the response you want the NetUI Java applet to supply at this prompt. For example, the password assigned to the user ID you specified in step 1.
Script Status	Enter a description of the activity this prompt and response represent. Example Supply Password

- 3 In the Sequence field, enter 3 and press Enter. Then use the following table to complete the Script Pattern, Script Value, and Script Status fields. When you are ready, press Go to save your entries.

Field	Description
Script Pattern	Enter the third prompt the server displays when you attempt to log into a telnet session. Example \$
Script Value	Enter the response you want the NetUI Java applet to supply at this prompt. For example, change directories to the MFG/PRO database server administration directory. Example <code>cd /qad/eB/</code> Note Depending on how you set up your telnet environment, the response you enter varies. For example, for maximum security you might enter a command to change to a restricted shell before changing to the directory containing the telnet script. See “Configuring the Telnet Environment” on page 120.
Script Status	Enter a description of the activity this prompt and response represent. Example Accessing DBServer Dir

- 4 In the Sequence field, enter 4 and press Enter. Then use the following table to complete the Script Pattern, Script Value, and Script Status fields. When you are ready, press Go to save your entries.

Field	Description
Script Pattern	Enter the fourth prompt the server displays when you attempt to log into a telnet session. Example \$
Script Value	Enter the response you want the NetUI Java client to supply. For example, launch the telnet connection script. Example <code>./qma.Production</code>
Script Status	Enter a description of the activity this prompt and response represent. Example Launching Production telnet connect script

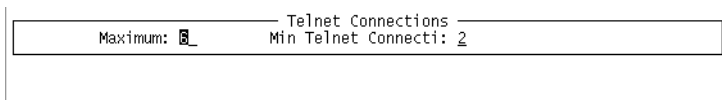
Verify the Log-In Sequence

To verify the log-in sequence, from a remote machine attempt to log into the telnet server using the log-in sequence you configured in MFG/PRO. You should receive a blank telnet screen after the telnet connection script is launched.

Configure Telnet Connection Settings

Use the following instructions to configure the telnet connection settings. These settings define the maximum and minimum number of telnet connections available to the associated user.

- 1 Once you configure and verify your telnet log-in sequence, access the Telnet Connections frame.



The screenshot shows a window titled "Telnet Connections". Inside, there are two fields: "Maximum:" followed by a text box containing the number "8", and "Min Telnet Connecti:" followed by a text box containing the number "2".

- 2 Use the following table to complete the Telnet Connections fields.

Field	Description
Maximum	Specify the maximum number of concurrent telnet connections this user can have open per session. This setting affects telnet maintenance programs only; it does not impact the number of browses and reports that can be run concurrently. Valid values are: • Unlimited: The associated user can have an unlimited number of concurrent telnet connections open. • Disabled: The associated user cannot have any telnet connections open and therefore cannot access any telnet maintenance programs in the NetUI. • Any numerical value between 1 and 99. Note Until you create a log-in script to initiate telnet sessions for this user, you cannot set this field to any value other than Disabled.
Minimum	Enter a value between one and nine to indicate the minimum number of telnet connections to be available to the associated user at all times. Set this value to the number of maintenance programs the user is likely to run simultaneously. Specifying a value here can dramatically reduce the wait time for these programs to display in the NetUI. However, setting this value too high depletes system resources.

CHAPTER 8

Configuring Telnet on Windows NT

This chapter provides instructions for loading the Georgia SoftWorks Windows NT Telnet Server (GSNTS) and configuring the NetUI environment to use a Windows NT telnet server.

Configuration Overview **132**

Installing the Telnet Server **133**

Configuring the Telnet Environment **135**

Configuring Telnet Options in MFG/PRO **141**

param91.p 14.13.2 Routing Maintenance (Main Screen)

Routing Code:	10-15000	NONAD (75) CDB (0)
Operation:	20	
Standard Operation:	10:30	INSPECTION, ALL SITE
Work Center:	Machine	
Description:	INSPEC PER PROX-000	
Machines per Op:	1	Allocation: 1
Overlap Units:	1	
Queue Time:	1.0	
Wait Time:	0.0	
Setup Time:	0.0	

Check Master

Configuration Overview

► For more information, see “Maintenance and Transaction Components” on page 20.

There are two ways to access maintenance and transaction programs: a telnet interface and a full GUI Java interface. To run telnet maintenance and transaction programs, the NetUI environment requires a telnet server. Using the Georgia SoftWorks Windows NT Telnet Server, you can implement the NetUI telnet server on a Windows NT machine.

To configure a Windows NT telnet server, you must perform the following major tasks:

- Install the Georgia SoftWorks telnet server
- Configure the telnet environment
- Configure the telnet settings in MFG/PRO

Download Updated Software

Although QAD provides the Georgia SoftWorks Windows NT Telnet Server on the NetUI media, you should download the latest version of the software to obtain the most recent patches and functionality.

You can download the telnet server software from the Georgia SoftWorks Web site:

<http://www.georgiasoftworks.com/>

Prerequisite Components

Before you install the Georgia SoftWorks Windows NT Telnet Server (GSNTS) files from your NetUI media, be sure the following components are installed on the same machine that you want use as the Windows NT telnet server.

- An MFG/PRO character client. This ensures that the MFG/PRO character client code required for telnet sessions is located on the telnet server. The character client startup shortcuts also provide the MFG/PRO parameter files used to start a telnet session. Although these files are not required and can be created manually, the following instructions assume that they exist.

- The following Progress client products:
 - Query
 - Network Client
- Optionally, QAD *highly recommends* that you obtain the Windows NT Resource Kit when implementing the NetUI telnet server on a Windows NT server. The kit provides utility programs, `Pulist.exe` and `Kill.exe`, that let you monitor and remove `_progress.exe` processes from the telnet server without shutting down the server.

Installing the Telnet Server

Use the following instructions to install and register the GSNTS on your Windows NT server.

Install the Software

- 1 Insert the NetUI CD-ROM in the CD-ROM drive.
 - 2 Run the setup program:
`CD-ROMDriveName:\nt\GeorgiaSoftworks\setup.exe`
 - 3 At the Welcome window, click Continue.
 - 4 Select Full Install and click Continue.
 - 5 Enter the GSNTS installation directory and choose Continue. Make sure to use the default `\gs_tnet` installation directory. You can place this directory on any local drive that can be accessed through the network.
- Note** If you do not install the GSNTS on the system drive—in most cases the C: drive—make sure that you have full access to the drive and to the GSNTS installation directory and subdirectories.
- 6 When GSNTS has been successfully installed, the Setup Succeeded window displays and GSNTS is automatically started.
 - 7 Double-click on the Registration icon in the GSNTS program group.

▶ See “Download Updated Software” on page 132 first.

Register the Software

To register the GSNTS, you must provide a product ID to Georgia SoftWorks so that a serial number can be generated for your product. This serial number is necessary to complete the GSNTS registration. Use the following instructions to register GSNTS.

Note Your serial number is based upon the product ID, which combines elements of the Windows NT server's hardware and software components. If these components are changed or upgraded, your GSNTS registration may no longer operate. If this occurs, contact Georgia SoftWorks about generating a new product ID and serial number.

- 1** In the Georgia SoftWorks Product Registration window, enter your customer information.

The information that displays in the Product Information section of the product registration window is automatically generated by the system.
- 2** Choose the Save to File button to save this information to a file, or choose print. Then, follow the appropriate step to supply the product ID to Georgia SoftWorks.
 - a** If you printed your registration information, FAX it to Georgia SoftWorks at 706-265-1020. When your information is received, a serial number is generated for your product and is sent to you by FAX.
 - b** If you saved your registration information to a file, E-mail the file to Georgia SoftWorks at the following address:
`registration@georgiasoftworks.com`
When your information is received, a serial number is generated for your product and is sent to you by E-mail.
- 3** When you receive your serial number, enter it in the appropriate field in the registration information section and click Register.

Configuring the Telnet Environment

Use the following instructions to set up Windows NT telnet environment. To configure the telnet environment, you must perform the following activities:

- “Create a Telnet Directory” on page 135
- “Create a General Startup File” on page 136
- “Create User-Specific Startup Files” on page 138
- “Set Up Local User Accounts” on page 140

Important The following instructions assume you have installed an MFG/PRO character client on the Windows NT telnet server.

▶ See “Prerequisite Components” on page 132.

Create a Telnet Directory

Use the following instructions to create a `telnet` directory in which you can store user- and group-specific telnet startup files:

- 1 Locate your MFG/PRO character client working directory. This directory was created when you set up the MFG/PRO character client on the Windows NT telnet server. For instructions, see the Character Client Setup chapter in the *MFG/PRO eB Installation Guide: Progress on Windows NT*.
- 2 Using File Manager or Windows NT Explorer, create a `telnet` subdirectory in your character client working directory.

In the following example, *CharClientWorkDir* refers to the character client working directory:

```
c:\CharClientWorkDir\telnet
```

Create a General Startup File

When a user connects to the telnet host by accessing a NetUI telnet program, the GSNTS general startup file automatically executes. This file performs the following functions:

- Starts a GSNTS session
- Determines which items the user can access on the host computer—whether MFG/PRO maintenance programs only or any application or folder on the host

The two types of GSNTS general startup files are described in Table 8.1.

Table 8.1
GSNTS Startup
Files

File	Description
k_start.bat	When this file is used to start the telnet session, the host machine's command prompt displays on the client terminal after a user exits the NetUI telnet program. From this prompt, the user can access directories and applications other than MFG/PRO on the host. They can also set environment variables, establish network connections, and perform a variety of other system administration tasks.
c_start.bat	When this file is used to start the telnet session, the NetUI client's telnet session automatically terminates when the user exits MFG/PRO. This means that users cannot access non-MFG/PRO programs or other directories on the telnet host from the NetUI.

In general, you should use the `c_start.bat` startup file for most NetUI users. For those users who require administrative access to the telnet host use the `k_start.bat` startup file.

Use the following instructions to set up telnet accessing using `c_start.bat`.

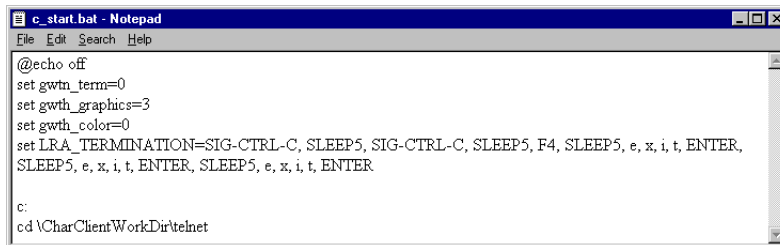
- 1 Open a new document in a text editor such as Notepad.
- 2 To create connection and termination parameter entries, enter the following:

```
@echo off
set gwtn_term=0
set gwtn_graphics=3
set gwtn_color=0
set LRA_TERMINATION=SIG-CTRL-C, SLEEP5, SIG-CTRL-C, SLEEP5,
F4, SLEEP5, e, x, i, t, ENTER, SLEEP5, e, x, i, t, ENTER, SLEEP5,
e, x, i, t, ENTER
```

- 3 Create an entry designating the drive where your MFG/PRO Windows character client files are installed; for example, C:.
- 4 Create an entry to change directories to your telnet subdirectory:
- 5 Save this file as `c_start.bat` in the `scripts` subdirectory below the GSNTS installation directory.

The `c_start.bat` file shown in the following example was saved in `c:\gs_tnet\scripts`.

See step 2 on page 135.



```
c_start.bat - Notepad
File Edit Search Help
@echo off
set gwtn_term=0
set gwtn_graphics=3
set gwtn_color=0
set LRA_TERMINATION=SIG-CTRL-C, SLEEP5, SIG-CTRL-C, SLEEP5, F4, SLEEP5, e, x, i, t, ENTER,
SLEEP5, e, x, i, t, ENTER, SLEEP5, e, x, i, t, ENTER
c:
cd \CharClientWorkDir\telnet
```

Create User-Specific Startup Files

The following steps provide instructions for creating user-and group-specific startup files. Each file specifies:

- The specific connection parameters for a user or group
- The databases to which that user or group can connect

See “Configure the Telnet Log-In Sequence” on page 143.

When a user connects to the telnet host, the system determines which user- or group-specific startup file to use based on parameters you set up in User Option Telnet Maintenance (36.20.10.3).

- 1 In the `telnet` directory, create subdirectories for each of the users or groups for which you want to create startup files; for example, `telnet\user1`.

Note Multiple MFG/PRO user IDs can reference the same startup file.

- 2 Open a new document in a text editor such as Notepad.

- 3 Add the following `set` lines to the file.

```
set DLC=
set MFGPRO=
set INI=
set PF=
set START_MFG=%DLC%\bin\_progres -pf %PF% -ini %INI%
-p %MFGPRO%\mfwb01a.p
```

- 4 Using the following table, provide environment variables for each `set` line.

set Line	Description
set DLC=	Enter the path to the Progress directory; for example: C:\dlc91
set MFGPRO=	Enter the path to the directory containing the MFG/PRO character client code; for example: C:\uschrcl

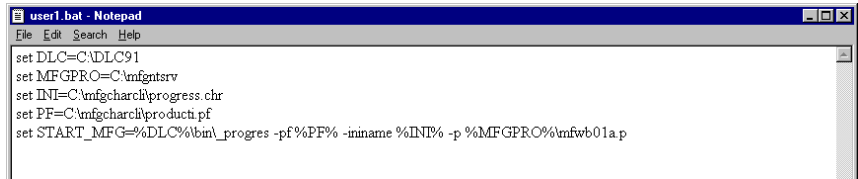
set Line	Description
set INI=	Enter the path to the MFG/PRO character client initialization file. This file is located in the character client working directory; for example: C:\uscharcli\progress.chr
set PF=	Enter the path to the MFG/PRO startup parameter file. This file is located in the character client working directory; for example: C:\uscharcli\Producti.pf

5 Save the file as the .bat file for the applicable user or group. For example, user1.bat.

6 Place this file in the appropriate user- or group-specific directory.
 \CharClientWorkDir\telnet\user1\user1.bat

See step 1 on page 138.

The following figure shows the user1.bat file created in the previous steps. The key variables in the final command line are discussed below.



```
user1.bat - Notepad
File Edit Search Help
set DLC=C:\DLC91
set MFGPRO=C:\mfgntsrv
set INI=C:\mfgcharcli\progress.chr
set PF=C:\mfgcharcli\producti.pf
set START_MFG=%DLC%\bin\progres -pf %PF% -iname %INI% -p %MFGPRO%\mfwb01a.p
```

- The first variable is the full path to the Progress character client executable. This variable should not be modified unless you move the Progress software.
- The -pf variable is the full path to the .pf file that specifies the MFG/PRO databases to which the telnet session connects.
- The -iname variable is the full path to the MFG/PRO character client’s initialization file.
- The -p variable indicates the MFG/PRO NetUI startup program.

Set Up Local User Accounts

Before NetUI users can access the telnet server, they must have local accounts set up on the Windows NT server. Follow these steps to define local accounts for individual users and groups.

- 1 From the system Start menu, open Programs\Administrative Tools(Common)\UserManager.
- 2 From the User menu in the User Manager window, choose Select Domain.
- 3 In the Domain field, enter the name of the Windows NT telnet server and press OK. All existing local accounts on the server display.
- 4 From the User menu, select New User.
- 5 Use fields in the New User window to set up a local user account. Be sure to specify the following:

Field	Value
Username:	Enter a user or group ID.
Password:	Enter the password for this user or group.
Confirm Password:	Reenter the user or group password.

Note The user IDs and passwords you define here must be identical to the user IDs and passwords you enter in the Script Lines portion of User Option Telnet Maintenance (36.20.10.3).

- 6 Press Add to save the local account.

Use the following table to record the user IDs and passwords you set up. You can use this table as a reference when entering script lines in User Option Telnet Maintenance.

User ID	Password

See “Configuring Telnet Options in MFG/PRO” on page 141.

Table 8.2
Local Windows NT
Server Users

Configuring Telnet Options in MFG/PRO

To access MFG/PRO maintenance and transaction programs using the NetUI's telnet interface, you must configure telnet settings in User Option Telnet Maintenance (36.20.10.3).

This program lets you define unique settings for individual users or enter an asterisk (*) in the User ID field to create a generic record of settings. This generic record is automatically associated with every user in the system that does not have a user-specific record. These instructions use the generic record.

Configure Telnet Server Settings

Use the following instructions to configure the telnet server settings:

- 1 Start an MFG/PRO session and access User Option Telnet Maintenance (36.20.10.3).

The screenshot shows the 'User Option Telnet Maintenance' window. At the top is a menu bar with 'User Menu', 'Edit', 'Queue', 'Options', and 'Help'. Below the menu is a toolbar with icons for file operations and help. The main area is divided into several sections:

- User ID:** A text field containing an asterisk (*) and a small icon to its right.
- Telnet Options:** A section containing:
 - Host: crsu04
 - Host O/S: UNIX
 - Port: 23
 - Image:
 - Script Timeout: 120
 - Idle Timeout: 240
- Script Lines:** A section containing:
 - Sequence: 1
 - Script Pattern: login:
 - Script Value: ckm
 - Script Status: logging in
- Telnet Connections:** A section containing:
 - Maximum: 6
 - Minimum: 2

- 2 In the User ID field, enter * to specify the generic record and press Go.

- 3 Use the following table to complete the Telnet Options fields in User Option Telnet Maintenance.

Field	Description
Host	Enter the machine name or IP address of the telnet server. The NetUI Java applet uses this information to establish the telnet connection.
Host O/S	Enter NT.
Port	Enter the port number for the telnet server.
Image	Optionally, enter the name of an image file to use as the menu icon for telnet programs. The image file must be in graphics interchange format (extension .gif). When you enter the name of the file in this field, do not include the .gif extension. The image file must be located in the <code>images</code> subdirectory below the <code>java</code> subdirectory, which is in the <code>multinet</code> directory in the Web server's document directory. The image size should be 24 by 24 pixels. If you leave this field blank, the standard QAD-supplied icon is used.
Script Timeout	Enter the number of seconds (1-999) the system will wait for the telnet log-in script to execute. If this value is exceeded, a time-out message displays and the session closes.
Idle Timeout	Enter the number of seconds (1-999) the system will wait after a telnet session is begun for an MFG/PRO program to execute.

- 4 When you are ready, press Go to access the Script Lines frame.

Configure the Telnet Log-In Sequence

To log in to and begin a telnet session on the telnet server, you must provide the NetUI Java applet with the telnet server log-in prompts and responses to these prompts. Taken together the prompts and responses make up the telnet log-in sequence.

You define the telnet log-in sequence in the Script Lines frame in User Option Telnet Maintenance.

Script Lines

Sequence: 1

Script Pattern: login:

Script Value: ckm

Script Status: logging in

Overview

Each line in the log-in sequence is stored in the MFG/PRO administration schema as a single `chui_det` record containing three values: pattern, value, and status. Each of these values is represented by a field in the Script Lines frame.

Table 8.3 provides a general description of the fields in the Script Lines frame.

Field	Description
Sequence	A number for the script line that indicates the order that the line appears in the overall log-in sequence.
Script Pattern	Enter the prompt generated by the telnet server when a telnet log-in occurs. Important The values you enter in the Script Pattern field must be identical to the prompts the telnet server displays when users attempt to log in.

Table 8.3
Script Lines Fields

Field	Description
Script Value	Enter the response to the telnet log-in prompt defined in Script Pattern.
Script Status	Optionally enter a description of the activity that the prompt and response represent. For example, if the prompt is Login: and the response is a user ID, enter Logging In to describe the activity. If an error occurs during the execution of the prompt and response, the description in the Script Status field display in the Java Plug-In Control Panel on the NetUI client. You can use these descriptions as an aid in troubleshooting telnet session issues. Note For more information on the Java Plug-In Console, see “trace.html” on page 202.

Define the Log-In Sequence

Use the following instructions to define the log-in sequence using the Script Lines fields in User Option Telnet Maintenance.

- 1 In the Sequence field, enter 1 and press Enter. Then use the following table to complete the Script Pattern, Script Value, and Script Status fields. When you are ready, press Go to save your entries.

Field	Description
Script Pattern	Enter: login:
Script Value	Enter a user ID; for example: user1. Note This ID must correspond to a local user account on the Windows NT server. Use Table 8.2 on page 140 as a reference.
Script Status	Enter a description of the activity this prompt and response represent; for example: Logging In.

- 2 In the Sequence field, enter 2 and press Enter. Then use the following table to complete the Script Pattern, Script Value, and Script Status fields. When you are ready, press Go to save your entries.

Field	Description
Script Pattern	Enter: password:

Field	Description
Script Value	Enter the password assigned to the user ID specified in step 1.
Script Status	Enter a description of the activity this prompt and response represent; for example: Supply Password.

- 3 In the Sequence field, enter 3 and press Enter. Then use the following table to complete the Script Pattern, Script Value, and Script Status fields. When you are ready, press Go to save your entries.

Field	Description
Script Pattern	Enter: domain:
Script Value	Enter your Windows NT telnet server's domain.
Script Status	Enter a description of the activity this prompt and response represent; for example: Supply Domain.

- 4 In the Sequence field, enter 4 and press Enter. Then use the following table to complete the Script Pattern, Script Value, and Script Status fields. When you are ready, press Go to save your entries.

Field	Description
Script Pattern	Enter the full directory path to the telnet subdirectory followed by a right angle bracket (>). Note See step 2 on page 135 for the location of the telnet subdirectory.
Script Value	Enter the user- or group-specific directory and startup script for the system to use to start telnet sessions for this user ID. Note See "Create User-Specific Startup Files" on page 138 for reference.
Script Status	Enter a description of the activity this prompt and response represent; for example: Starting GTNTS as user1.

The information in the following figure is based on the examples provided in previous sections. Your information may be different.

Script Lines

Sequence: 4

Script Pattern:	C:\uscharcli\telnet>
Script Value:	\user1\user1.bat
Script Status:	Starting GSNTS as user1

- 5 In the Sequence field, enter 5 and press Enter. Then use the following table to complete the Script Pattern, Script Value, and Script Status fields. When you are ready, press Go to save your entries.

Field	Enter
Script Pattern	Enter the full directory path to the <code>telnet</code> subdirectory (see step 2 on page 135) followed by a right angle bracket (<code>></code>).
Script Value	Enter the start MFG/PRO system variable: <code>%start_mfg%</code> . Note You set up this variable in step 3 on page 138.
Script Status	Enter a description of activity this prompt and response represent; for example: Launching Progress character client exe.

Verify the Log-In Sequence

To verify the log-in sequence, from a remote machine attempt to log into the telnet server using the log-in sequence you configured in MFG/PRO. You should receive a blank telnet screen after the telnet connection script is launched.

Configure Telnet Connection Settings

Use the following instructions to configure the telnet connection settings. These settings define the maximum and minimum number of telnet connections available to the associated user.

- 1 Once you configure and verify your telnet log-in sequence, access the Telnet Connections frame.

Telnet Connections	
Maximum: 6	Minimum: 2

2 Use the following table to complete the Telnet Connections fields.

Field	Description
Maximum	Specify the maximum number of concurrent telnet connections this user can have open per session. This setting affects telnet maintenance programs only; it does not impact the number of browses and reports that can be run concurrently. Valid values are: • Unlimited: The associated user can have an unlimited number of concurrent telnet connections open. • Disabled: The associated user cannot have any telnet connections open and therefore cannot access any telnet maintenance programs in the NetUI. • Any numerical value between 1 and 99. Note Until you create a log-in script to initiate telnet sessions for this user, you cannot set this field to any value other than Disabled.
Minimum	Enter a value between one and nine to indicate the minimum number of telnet connections to be available to the associated user at all times. Note Always set this to a value greater than 0 (zero) or an error displays when users access MFG/PRO maintenance programs. Set this value to the number of maintenance programs the user is likely to run simultaneously. Specifying a value here can dramatically reduce the wait time for these programs to display in the NetUI. However, setting this value too high depletes system resources.

CHAPTER 9

Configuring the NetUI Environment

This chapter provides instructions for configuring the NetUI environment.

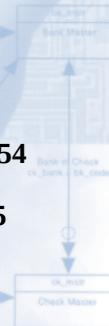
Configuration Summary **150**

Setting Up the RMI Server **150**

Creating the NetUI Help Script **154**

Configuring NetUI Constants **155**

Installing Client Files **168**



Routing Maintenance (Main Screen)	
Routing Code:	10-15000
Operation:	20
Standard Operation	
Work Center	1030
Machine	INSPEC, ALL SITE
Description	INSPEC PER PROX-000
Machines per Op	1
Overlap Units	1
Queue Time	1.0
Wait Time	0.0
Setup Time	0.0
Ready for Production	0.0

Configuration Summary

After installing the NetUI files on the Web server, you must complete the following major tasks to configure the NetUI environment:

- Set up the RMI server
- Create the NetUI help script
- Define the NetUI constants
- Install files on each client machine

Setting Up the RMI Server

The NetUI uses a remote method invocation (RMI) server to manage the interface elements of the extended user interface (XUI).

The Java programming language uses RMI for remote procedure call (RPC) functionality. RMI lets Java programs call procedures and retrieve files from remote machines to facilitate dispersed computing environments. In the case of the NetUI, the RMI server lets the NetUI Java applet access procedures and files on the Web server.

The JDK is required for RMI. The version of the JDK required depends on your service pack level:

- Prior to Service Pack 2, JDK 1.1.8
- Starting with Service Pack 2, JDK 1.3.0_0.2 or higher

♦ See *Installation Guide: eB Desktop*.

Important The optional eB Desktop product always requires at least JDK 1.3.0_02. If you are planning to install eB Desktop with Service Pack 1, do not install the RMI now. You will install an updated RMI as part of the eB Desktop installation. Starting with Service Pack 2, you can use the NetUI RMI for eB Desktop.

To set up the RMI server, perform the following steps on the machine where you want the RMI server to run. These instructions assume that you have installed the JDK on the machine where the RMI server will run using the instructions in “Installing the Java Development Kit” on page 57.

- 1 Insert the NetUI software media into the CD-ROM drive of the machine where you want the RMI server to run.
If you are implementing the RMI server on a UNIX machine, use the instructions in “Mount the CD-ROM” on page 101 to mount the NetUI CD-ROM.
- 2 From the CD-ROM, copy the `rmi` directory to a directory of your choice on your machine. Record the location of the `rmi` directory in Table 12.3 on page 224. This directory is referred to as the RMI installation directory.
- 3 If you received a service pack with your installation media, back up your current RMI directory. Then insert the service pack media into the CD-ROM drive and copy any files located in the `rmi` directory to the location used in step 2. Follow the directions in the service pack installation guide for updating the RMI.

- 4 Two MFG/PRO functions update files on the RMI server: the UI Configurator and the Appearance Manager. If you plan to use these functions to tailor the appearance of NetUI programs, make sure the appropriate permissions are set up for the RMI installation directory.
In addition, trace output may be written to the `error` directory by any user of the NetUI.

♦ See *User Guide: eB Desktop and Network User Interface*.

When NetUI users modify XUI programs using the UI Configurator, change color and font schemes, or generate errors, they do so as the user and group that started the RMI server processes.

- This user and group should have read, write, and execute permissions to the RMI installation directory.
- This user should have read and write permissions to the files in the RMI installation directory.
- All users should have read and execute permissions to the RMI installation directory and read permissions to the files in the installation directory.

Note As part of managing your RMI server, you may need to delete obsolete user files. This topic is discussed in Chapter 11.

♦ See “RMI Server Administration” on page 215.

- 5 Access the RMI installation directory, and launch the RMI configuration script appropriate to the operating system of your machine.

- For UNIX servers, run `config.ksh`.
 - For Windows NT servers, run `config.exe`.
- ♦ See Table 12.2 on page 222.
- 6 When prompted, enter the path to the directory containing the JDK.
 - 7 The configuration script attempts to detect the host name of your machine and prompts you to verify it. If the host name is correct, specify Yes. If the host name is incorrect, specify No and enter the host name at the subsequent prompt.

Record the RMI server host name in Table 12.4 on page 225. You must supply this information when you configure the NetUI constants.
 - 8 When prompted, enter the port number you want the RMI server to use. Record the RMI port number in Table 12.4 on page 225. You must supply this information when you configure the NetUI constants.
 - 9 Once you complete these prompts, the configuration script creates an RMI startup script in the `rmi` directory. Depending on your machine's operating system, the script has different names.
 - For UNIX servers, the script is `rmi.ksh`. When you run this script, two log files are created: `rmi.log` and `rmipid.log`.
Use `rmi.log` to review the functions the RMI server performs.
Use `rmipid.log` to determine the process IDs for the RMI server processes. This lets you shut down the RMI server if necessary.
 - For Windows NT servers, the script is `rmi.bat`. When you run this script, the RMI server processes are launched in separate windows. One window contains the RMI Registry process and the other contains the Java RootManager process. Both of these processes must be running for the RMI server to function.
 - 10 Using the appropriate startup script, start the RMI server.
 - 11 To verify that the RMI server is functioning correctly, use the instructions appropriate to your operating system.

- For UNIX servers, access the `rmipid.log` file and record the RMI process IDs. There should be one process for the RMI Registry and one for the Java RootManager. Then, verify that the processes are running using the following command:

```
ps -ef | grep ProcessID
```

- For Windows NT servers, verify that there is an active command prompt window for each of the RMI processes: one for the RMI Registry and one for the Java RootManager.

RMI Directory Structure

The NetUI uses the following directories installed below the root directory on the RMI server:

Data. Contains the dataspace definition for each XUI application.

Error. Contains directories corresponding to NetUI user log-in IDs for tracking Java exceptions. The system appends messages that would normally be directed to the Java console to a text file named `NetUIErrorLog.txt` in the directory that corresponds to the log-in ID of the user generating the message.

Managers. Contains the management setup for application servers and WebSpeed. Also contains the `UIElements.xml` file, which stores the default NetUI color and font scheme set with the Appearance Manager.

MessageService. Contains the messaging service setup for each XUI application.

Presentation. Contains subdirectories of application presentation files, including:

- **QADStandard.** This directory contains the original application screen definitions provided when the NetUI is installed.
- **Public.** Initially, `Public` contains exact copies of the screen definitions in `QADStandard`. Screen definitions that are modified with the UI Configurator are stored in the `Public` directory when they will be used by more than one person. These screens are presented to users when the system does not find a screen associated with their user ID.

- Individual directories. When screen definitions exist for individual users, any number of subdirectories representing user IDs may exist. A screen definition created for a specific individual is stored automatically in a directory that matches the user's NetUI log-in ID.

`Process`. Contains the process definition for each XUI application.

Note The error directory and `UIElements.xml` file exist only after you install Service Pack 1.

Creating the NetUI Help Script

To enable online help for the NetUI, you must create a help script containing your environment variables. During the NetUI software installation, a help configuration script was installed along with the help files in the `multinet` subdirectory below the document directory on your Web server. You use the configuration script to create the help script.

- 1 On your Web server, access the `multinet` subdirectory below the document directory.
- 2 In the `multinet` directory, access the `help` subdirectory.
- 3 From the `help` directory, copy the contents of the `cgi-bin` subdirectory to your Web server's executables directory.
- 4 Access the Web server's executables directory.
- 5 Depending on your Web server's operating system, launch the appropriate help configuration script.
 - For UNIX Web servers, run `config.ksh`.
 - For Windows NT servers, run `config.exe`.
- 6 When prompted, enter the path to the directory where you installed the JDK.
- 7 When prompted, enter the path to the Web server's document directory.

- 8 Once you complete these prompts, the configuration script creates the NetUI help script in the Web server's executables directory. Depending on the Web server's operating system, the help script has different names:
 - For UNIX servers, the script is `ebhelp.ksh`.
 - For Windows NT servers, the script is `ebhelp.exe`.
- 9 To verify that the help script is functioning correctly, run the script. This should produce an HTML page as output.
- 10 Record the path and name of the NetUI help script in Table 12.4 on page 225. You must supply this information when you configure the NetUI constants.

Configuring NetUI Constants

At run time, NetUI client sessions obtain environment information—referred to as constants—from tables in the MFG/PRO administration schema. The following tables contain NetUI constants.

- Label Master (`lbl_mstr`). You define three different types of NetUI constants in `lbl_mstr` using Label Master Maintenance (36.4.17.1).
 - NetUI service information. This includes RMI and primary AppServer connection values as well as NetUI online help values. You must provide these values based on your environment.
 - Mapping of MFG/PRO language codes to ISO language and country codes, and to appropriate number and date formats. In most cases, you can use the default mapping for your environment. However, in some cases you may need to modify the ISO country code to ensure correct display and field validation in the NetUI.
 - Mapping of Progress code page values to Java code page values. Use the default mapping supplied by QAD.
- AppServer Master (`aps_mstr`). You define the auxiliary AppServer constants in `aps_mstr` using AppServer Service Maintenance (36.19.1).

Use the following instructions to verify and define NetUI constants for your environment.

▶ See *Installation Guide: Service Pack 2*.

Important All constants in the label master are stored under the US language. If you are installing the NetUI in a non-English environment, you still need to load this required subset of US label data. The file containing this data is provided on Service Pack 2.

Configure NetUI Service Constants

NetUI constants are configured in Label Master Maintenance using QAD supplied terms. For each term, you specify your environment information using the long label value. Table 9.1 lists the service terms in Label Master Maintenance and provides a description of the long labels associated with each term.

Table 9.1
Server Information

Term	Description
OPENAPPHOST	The primary AppServer host machine and the NameServer port number. This value uses the following format: Host Name:Port Number
OPENAPPSERVICE	Primary AppServer broker name.
RMIHOST	RMI server host.
RMIPORT	RMI server port.
HELPHOST	The machine name and port number for the Web server. This value uses the following format: Host Name:Port Number Note You only need to include the port number if you did not use the default Web server port, 80.
HELPSCRIPT	The path and name of the NetUI online help script.

Configure Primary AppServer Constants

Use these instructions to define the primary AppServer constants using the OPENAPPHOST and OPENAPPSERVICE parameters. You must define these constants to use XUI-enabled programs.

- 1 Launch an MFG/PRO session and access Label Master Maintenance (36.4.17.1).
- 2 In the Language ID field, specify US and press Go.

Important All NetUI constants must be created with US as the language, even when you are using the NetUI in a non-English environment.

- 3 In the Term field, enter OPENAPPHOST and press Enter.
- 4 In the Long Label field, use the following steps to specify the host information for the primary AppServer.
 - a Enter the machine name of the computer where the primary AppServer broker is located.
 - b Enter a colon.
 - c Enter the port number for the controlling NameServer for the primary AppServer. If you set up the primary AppServer using the instructions in this guide, the controlling NameServer is the standard server provided by Progress, NS1. By default, this server uses port 5162.

See Table 12.2 on page 222.

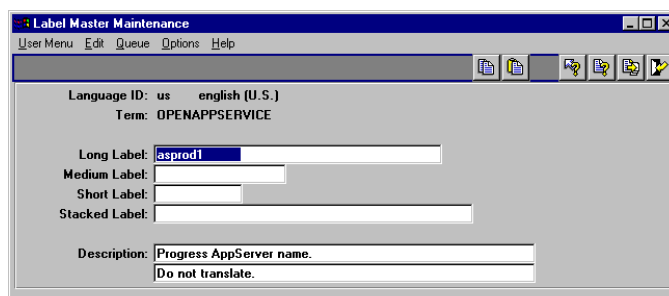
If you configured a different NameServer and set it up as the controlling NameServer for the primary AppServer, specify the appropriate port.

The screenshot shows a window titled "Label Master Maintenance" with a menu bar (User Menu, Edit, Queue, Options, Help) and a toolbar. The main area contains the following fields:

- Language ID: us english (U.S.)
- Term: OPENAPPHOST
- Long Label: crsu03:5162
- Medium Label: (empty)
- Short Label: (empty)
- Stacked Label: (empty)
- Description: Progress AppServer host.
Do not translate.

- 5 Press Go. You are returned to the Language ID field. Accept the default US and press Enter.
- 6 In the Term field, enter OPENAPPSERVICE and press Enter.
- 7 In the Long Label field, enter the name of the primary AppServer broker name.

See Table 12.2 on page 222.



- 8 Press Go to return to the Language ID field.

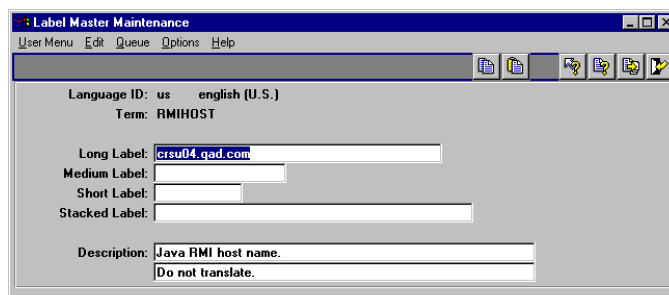
Configure RMI Server Constants

Use these instructions to define the RMI server constants. You must define these constants to use XUI-enabled programs.

Important All NetUI constants must be created with US as the language, even when you are using the NetUI in a non-English environment.

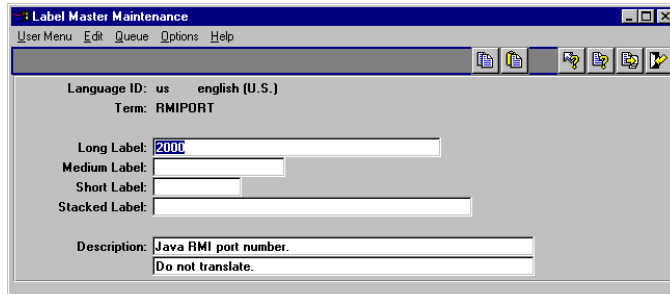
- 1 Accept the default US in the Language ID field and press Enter.
- 2 In the Term field, enter RMIHOST and press Enter.
- 3 In the Long Label field, specify the machine name of the computer where the RMI server is running.

See Table 12.4 on page 225.



- 4 Press Go. You are returned to the Language ID field. Accept the default US and press Enter.
- 5 In the Term field, enter RMIPORT and press Enter.
- 6 In the Long Label field, specify the port number of the RMI server.

See Table 12.4 on page 225.



- 7 Press Go to return to the Language ID field.

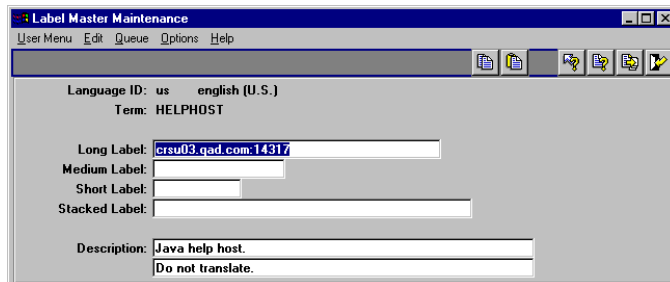
Configure NetUI Online Help Constants

Use these instructions to define the NetUI online help constants. You must define these constants to access online help in the NetUI.

- 1 Accept the default US in the Language ID field and press Enter.
- 2 In the Term field, enter HELPHOST and press Enter.
- 3 In the Long Label field, specify the machine name of the computer where the Web server is located. Then, enter a colon and specify the Web server's HTTP port number.

See Table 12.2 on page 222.

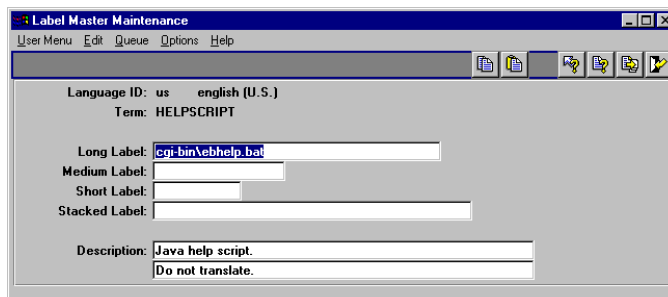
Note You do not need to include a colon and port number if your Web server uses the default HTTP port, 80.



- 4 Press Go. You are returned to the Language ID field. Accept the default US and press Enter.
- 5 In the Term field, enter HELPSCRIPT and press Enter.

See Table 12.2 on page 222.

- 6 In the Long Label field, enter the Web server's CGI alias. Then specify the NetUI help script, `ebhelp.ksh` or `ebhelp.exe` depending on the Web server's operating system.



- 7 Press Go to return to the Language ID field.

Map Language Codes

Overview

See the Multiple Language Setup chapter in the MFG/PRO installation guide for your system.

MFG/PRO is available in a number of different language translations. Each MFG/PRO language is identified by a two-letter language code. You can use this code to associate different language displays with different MFG/PRO users to create a multiple-language environment.

When a user logs in to MFG/PRO from either the Windows or character interfaces, MFG/PRO determines the language to use for labels, messages, and other interface elements based on the MFG/PRO language code associated with that user.

A similar process occurs when a user logs in to MFG/PRO from the NetUI. However, the NetUI uses Java language codes. These codes are defined by the ISO (International Organization for Standardization). Since not all MFG/PRO language codes follow this standard, a mapping must occur.

In addition to language codes, Java also uses country codes to accommodate regional language differences. For example, while both the US and Great Britain use the English language, there are many differences between the two—for example, spelling differences and date format differences. In this case, Java uses the ISO language code for

English (en) and then adds the ISO country code to define the correct display for each country—en,US for the US and en,GB for Great Britain.

When a user logs in to MFG/PRO from the NetUI, the system performs the follow process to determine the MFG/PRO language code and map it to the appropriate ISO language codes.

- 1 The system checks the User Master table (usr_mstr) to find the record that determines the MFG/PRO language code associated with the user's ID.
- 2 Once the MFG/PRO language is determined, the system checks the labels created in Label Master Maintenance (36.4.17.1) to determine the associated ISO language code, country code, date separator, number format, and date format.
- 3 The system uses the ISO language code to determine the language-related aspects of the interface.

Table 9.2 lists MFG/PRO language codes and the ISO language and country codes to which they map.

Important ISO language codes are lowercase and country codes are uppercase. Java is case sensitive and expects the codes in the ISO format.

Table 9.2
MFG/PRO
Language Codes

Language	MFG/PRO eB Language Code	ISO Language Code	ISO Country Code	Country
Arabic	AR	ar	EG	Egypt
Bulgarian	BU	bg	BG	Bulgaria
Chinese (Traditional)	TW	zh	TW	Taiwan
Chinese (Simplified)	CH	zh	CN	China
Czech	CZ	cs	CZ	Czech Republic
Danish	DA	da	DK	Denmark
Dutch	DU	nl	NL	Netherlands
Finnish	FI	fi	FI	Finland
French	FR	fr	FR	France
German	GE	de	DE	Germany
Greek	GR	el	GR	Greece
Hebrew	HE	iw	IL	Israel

Language	MFG/PRO eB Language Code	ISO Language Code	ISO Country Code	Country
Hungarian	HU	hu	HU	Hungary
Italian	IT	it	IT	Italy
Japanese	JP	ja	JP	Japan
Korean	KO	ko	KR	South Korea
Lithuanian	LI	lt	LT	Lithuania
Norwegian	NO	no	NO	Norway
Polish	PL	pl	PL	Poland
Portuguese	PO	pt	BR	Brazil
Romanian	RO	ro	RO	Romania
Russian	RU	ru	RU	Russian
Spanish (Castilian)	CS	es	ES	Spain
Spanish (Latin)	LS	es	MX	Mexico
Swedish	SW	sv	SE	Sweden
Turkish	TU	tr	TR	Turkey
Ukrainian	UA	uk	UA	Ukrania
US English	US	en	US	United States

Access Language Code Records

The mapping of MFG/PRO language codes to ISO codes is accomplished through reserved terms maintained in the Label Master table (lbl_mstr) in MFG/PRO.

Use the following instructions to view the term for your language:

- 1 Launch an MFG/PRO session and access Label Master Maintenance (36.4.17.1).
- 2 In the Language ID field, specify US and press Go.
- 3 Each MFG/PRO language code has a predefined term used to map to the appropriate ISO language, country code, and associated date and number format information. In the Term field, identify the record for your language by selecting the MFG/PRO language code followed by the `_LANGUAGE` suffix.

Tip
Select US
regardless of the
language used in
your environment.

The lbl_mstr record used to map the MFG/PRO language code to the ISO language codes and display formats is shown in the Long Label field. The following example record is for US-English, MFG/PRO language code US.

Field	Value
Language ID	US
Term	US_LANGUAGE
Long Label	en,US,/,A,mdy
Description	MFG/PRO language code to Java language, country codes. Do not translate.

Important The language ID associated with all language terms is US. To modify the language term for German, you must specify US in the Language ID field as in the following example.

Field	Value
Language ID	US
Term	GE_LANGUAGE
Long Label	de,GE,,E,dmy
Description	MFG/PRO language code to Java language, country codes. Do not translate.

Language Code Elements

This section contains information on each of the elements in the Long Label field of the language code record.

The long label record for each language in the lbl_mstr has the following format:

*ISO language code,ISO country code,date separator,numeric format code,
date format*

There are five components to the long label:

- ISO language code.
- ISO country code.
- Date separator. This element is not currently referenced in the NetUI. The only date separator that is supported is the slash (/).

See Table 9.2 on page 161.

- Numeric format. This element uses one of the following values:
 - A (American). The comma separates thousands and the period is a decimal point; for example, 100,000.00.
 - E (European). The period separates thousands and the comma is a decimal point; for example, 100.000,00.
- Date format. This element uses one of the following values:
 - mdy (Month/Day/Year)
 - dmy (Day/Month/Year)
 - ymd (Year/Month/Day)

Modify Language Code Records

If you are installing MFG/PRO in an English-speaking country that does not use American date and decimal formats, then you need to modify the value of the Long Label field associated with the US_LANGUAGE term. You can change the country code, numeric format, and date format values.

For example, in England, define the label as en,GB,/,E,dmy. If you are unsure of the correct ISO country code to use, you can refer to the information at the following URL:

http://www.chemie.fu-berlin.de/diverse/doc/ISO_3166.html

Important Be careful when you make modifications to preserve consistency between the language code and date format. These two values are used differently within the NetUI and setting them inconsistently can result in inconsistent date formats between NetUI programs and printed reports.

The date format determines the date format used by the Progress session. The client uses the language code to set the locale, including the date. Setting the language code to fr (French) and the date format to mdy has the following results:

- Browse and report input screens display dates in dmy based on the date format associated with the fr locale.
- The report output, generated from Progress, displays with mdy based on the date format in the label master.

Table 9.3 lists the term and long label defined in the label master for each supported MFG/PRO language.

Table 9.3
Pre-defined
Language Terms

Language	MFG/PRO eB Language Code	Label Master Term	Long Label
Arabic	AR	AR_LANGUAGE	ar,EG/,A,dmy
Bulgarian	BU	BU_LANGUAGE	bg,BG-,E,ymd
Chinese (Traditional)	TW	TW_LANGUAGE	zh,TW-,A,ymd
Chinese (Simplified)	CH	CH_LANGUAGE	zh,CN-,A,ymd
Czech	CZ	CZ_LANGUAGE	cs,CZ-,E,ymd
Danish	DA	DA_LANGUAGE	da,DK-,E,dmy
Dutch	DU	DU_LANGUAGE	nl,NL-,E,dmy
Finnish	FI	FI_LANGUAGE	fi,FI,,E,dmy
French	FR	FR_LANGUAGE	fr,FR/,E,dmy
German	GE	GE_LANGUAGE	de,DE,,E,dmy
Greek	GR	GR_LANGUAGE	el,GR/,E,dmy
Hebrew	HE	HE_LANGUAGE	iw,IL/,A,dmy
Hungarian	HU	HU_LANGUAGE	hu,HU,,E,ymd
Italian	IT	IT_LANGUAGE	it,IT/,E,dmy
Japanese	JP	JP_LANGUAGE	ja,JP/,A,ymd
Korean	KO	KO_LANGUAGE	ko,KR-,A,ymd
Lithuanian	LI	LI_LANGUAGE	lt,LT,,E,ymd
Norwegian	NO	NO_LANGUAGE	no,NO,,E,dmy
Polish	PL	PL_LANGUAGE	pl,PL-,E,ymd
Portuguese	PO	PO_LANGUAGE	pt,BR/,E,dmy
Romanian	RO	RO_LANGUAGE	ro,RO,,E,dmy
Russian	RU	RU_LANGUAGE	ru,RU,,E,dmy
Spanish (Castilian)	CS	CS_LANGUAGE	es,ES/,E,dmy
Spanish (Latin)	LS	LS_LANGUAGE	es,MX/,A,dmy
Swedish	SW	SW_LANGUAGE	sv,SE-,E,ymd
Turkish	TU	TU_LANGUAGE	tr,TR,,E,dmy
Ukranian	UA	UA_LANGUAGE	uk,UA/,E,dmy
US English	US	US_LANGUAGE	en,US/,A,m dy

Map Code Pages

Java uses a different set of values to represent code pages than Progress does. When a user logs into the NetUI, the Progress code page value must be mapped to the Java one. This is accomplished through a set of records stored in lbl_mstr.

The Progress code page value is stored as a term consisting of the code page value with _CODEPAGE appended. The long label associated with that term represents the Java equivalent of the Progress value.

By default, the Progress code page value for your environment is based on code page value used by your MFG/PRO databases. The Progress value is then mapped to the Java value using the _CODEPAGE terms in lbl_mstr. These terms are predefined and you do not need to perform any actions for the mapping to occur.

Table 9.4 lists the Progress code page to Java code page mapping that the NetUI uses.

Important Do not modify these records.

Table 9.4
Code Page
Cross-Reference

Term (Progress Value)	Long Label (Java Value)
ISO8859-1_CODEPAGE	ISO8859_1
1250_CODEPAGE	Cp1250
1251_CODEPAGE	Cp1251
1252_CODEPAGE	Cp1252
1253_CODEPAGE	Cp1253
1254_CODEPAGE	Cp1254
BIG-5_CODEPAGE	Big-5
GB2312_CODEPAGE	EUC-CN
SHIFT-JIS_CODEPAGE	SJIS

Configure the Auxiliary AppServer

An additional AppServer must be configured in AppServer Service Maintenance (36.19.1). This server acts as an auxiliary AppServer to the primary AppServer and is used to execute short, discrete transactions in the MFG/PRO database when the client transaction has a long scope.

Use the following instructions to configure you auxiliary AppServer:

- 1 In an MFG/PRO session, access AppServer Service Maintenance (36.19.1).
- 2 In the Service Name field, enter mfgpro and press Go.
- 3 Use the following table and figure to configure the mfgpro service record for your auxiliary AppServer.

See Table 12.2 on page 222.

Field	Description
Description	Enter an optional description of the auxiliary AppServer.
Application Service	Enter the name of the broker for the auxiliary AppServer.
IP Address or Host Name	Enter either the IP address or the machine name of the machine on which the auxiliary AppServer is running.
Port Number	Enter the port number for the controlling NameServer for the auxiliary AppServer. If you set up the auxiliary AppServer using the instructions in this guide, the controlling NameServer is the standard server provided by Progress, NS1. By default, this server uses port 5162. If you configured a different NameServer and set it up as the controlling NameServer for the auxiliary AppServer, specify the appropriate port.
Parameters	Leave this field blank.

The screenshot shows the 'AppServer Service Maintenance' window. The 'Service Name' field contains 'mfgpro'. The 'Description' field contains 'auxiliary appserver'. The 'Application Service' field contains 'asprod2'. The 'IP Address or Host Name' field contains 'crsu03'. The 'Port Number' field contains '5162'. The 'Parameters' field is empty. At the bottom, the 'E-mail User ID' field is empty and the 'E-mail Level' is set to 'NONE'.

- 4 When you complete the AppServer Service Maintenance fields, press Go to save your changes.

Installing Client Files

The exact steps in the client setup vary depending on your service pack level. To install the NetUI clients, complete the procedure appropriate for your service pack level on each client machine.

Important If you plan to use eB Desktop with Service Pack 1, you do not need to perform this step. The eB Desktop client install replaces this one.

For Windows NT and Windows 2000 clients, users must be logged in with administrator privileges to complete these installation steps.

Run Client Setup before Service Pack 2

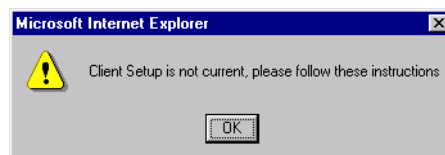
Important Before beginning, make sure that your Web server is running.

- 1 From the client, launch the Web browser, go to the NetUI Web server, and access the `multinet` subdirectory below the primary document directory.

In the following example URL, *HostName* refers to the NetUI Web server. This example assumes that when accessing the Web server you are placed in the document directory by default:

`http://HostName/multinet/`

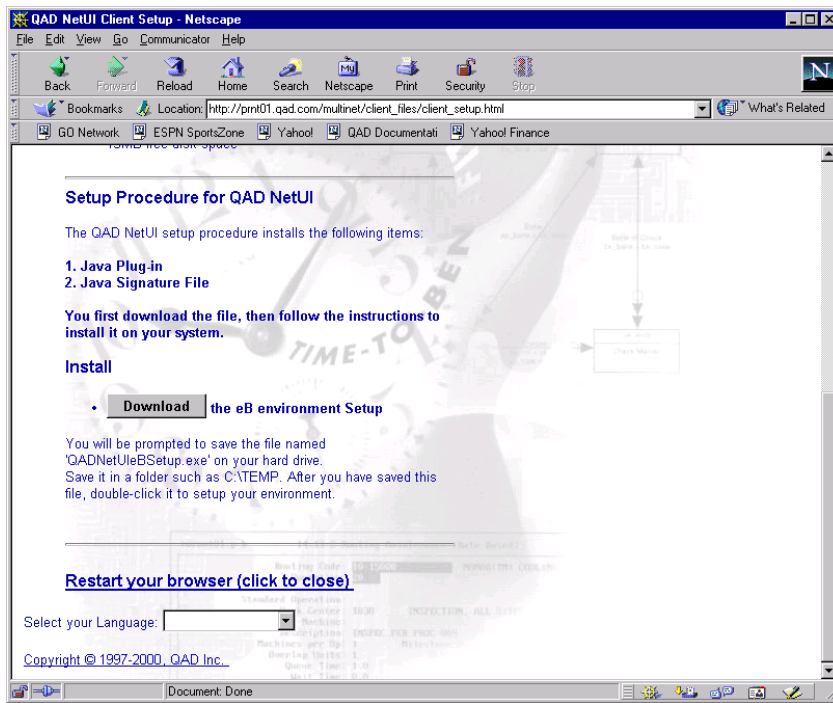
- 2 The following prompt indicates that the NetUI client files are not current on this machine. In this case, this is because they have not been installed. Click OK to continue.



You are redirected to the `client_setup.html` file in the `client_files` subdirectory. This subdirectory is located below the `multinet` directory in the Web server's document directory.

- 3 When the QAD NetUI Client Setup window displays, read the NetUI client requirements and verify that the client machine meets them.

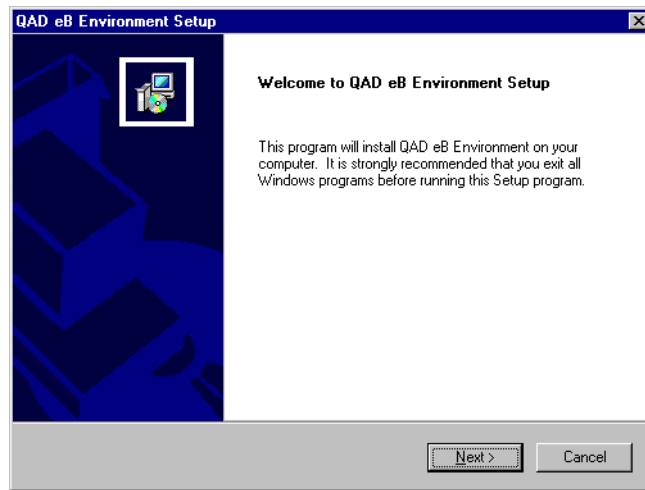
- 4 Read the instructions in the Setup Procedure for QAD NetUI section. When ready, click Download to begin installing the NetUI client files.



- 5 When prompted to save QADNetUIeBSetup.exe, specify a directory on the client machine. You will run the NetUI client setup from the directory you specify.

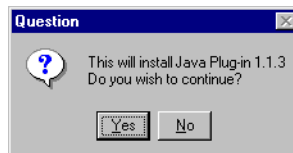
Note Depending on the type of browser you use to download the client files, you may not receive this prompt. For example, Netscape Navigator 4.7 automatically downloads and saves QADNetUIeBSetup.exe and then prompts you to install it. If this is the case for your browser, click Yes when prompted to install the client files and skip to step 7.

- 6 Access QADNetUIeBSetup.exe and double-click it.
- 7 At the Welcome window, click Next.



- 8 When prompted, click Yes to install the Java Plug-In. The client setup process installs the Java Plug-In to C:\Program Files\Java Plug-in 1.1.

Important The NetUI Java files are installed in the lib\applet subdirectory below this directory.



- 9 At the Setup Complete window, click Finish to complete the NetUI client installation.



- 10 Close the browser on the client machine and restart it. You can now access the NetUI.

Client Updates

The following list briefly describes the tasks performed by the client setup process and the items that it installs on the NetUI client machine:

- Installs the Java Plug-In 1.1.3.
- Updates the default `swingall.jar` file included with the Java Plug-In using a QAD-supplied `swingall.jar`.
- Installs the QAD signature file.
- Installs the NetUI Java code—`multinet.jar` and `o4glrt.jar`—in the `applet` subdirectory below the `lib` directory in the Java Plug-In installation directory. By default, this directory is located at:
`C:\Program Files\Java Plug-in 1.1\lib\applet`
- Sets the Java Run Time Parameters in the Java Plug-in Control Panel to `-mx64m`.
- Creates a cookie that indicates setup is complete.

Run Client Setup after Service Pack 2

Important Before beginning, make sure that your Web server is running.

To run the NetUI after Service Pack 2 is installed, clients must have the correct version of the JRE installed, and the JRE must be tailored for NetUI requirements.

The client installation occurs in two major steps:

- Complete the install of JRE 1.3.
- Complete the NetUI setup sequence.

Important You should uninstall earlier versions of the Java Plug-In on the client machines before proceeding with this installation.

If your client machines already have a version of JRE 1.3 installed, you do not need to replace it with the version on the NetUI CD. However, if you have a non-English environment, make sure you have installed the international version. This can be identified by the presence of the `i18n.jar` file in the `lib` subdirectory of the JRE installation directory.

The client installation script does not copy the `NetUI.jar` files to the client. This is managed automatically by JRE 1.3. Whenever a newer `.jar` file exists on the Web server, it is downloaded before the Java applet runs. This ensures that the client is always running the most up-to-date code.

Install JRE 1.3

- 1 From the client, launch the Web browser, go to the NetUI Web server, and access the `multinet` subdirectory below the primary document directory.

In the following example URL, *HostName* refers to the NetUI Web server. This example assumes that when accessing the Web server you are placed in the document directory by default and that `index.html` is the default file to display:

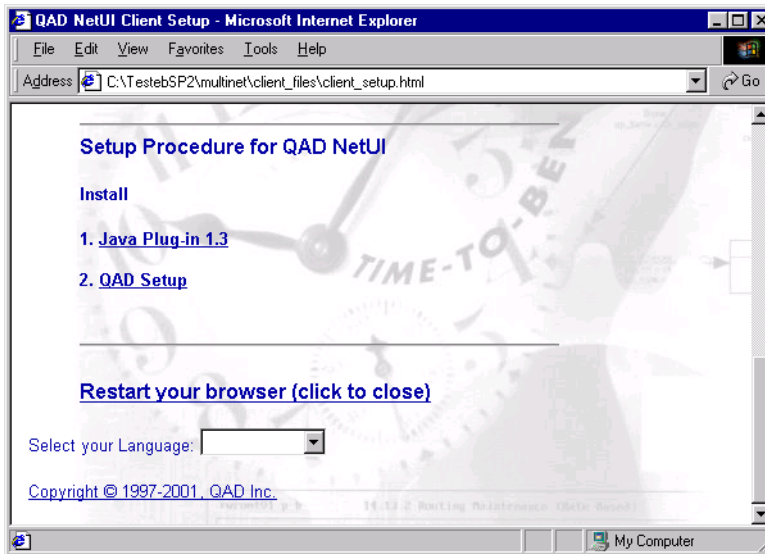
```
http://HostName/multinet/
```

- 2 The following prompt indicates that the NetUI client files are not current on this machine. Click OK to continue.

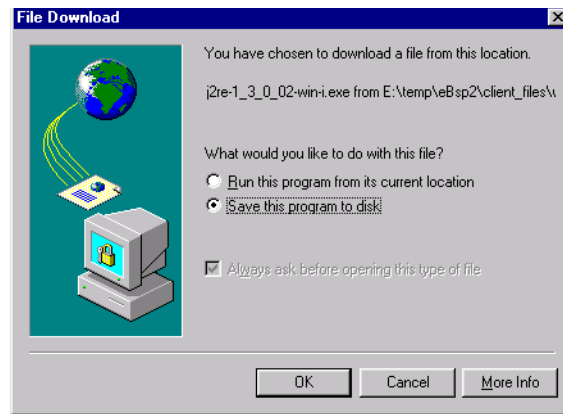


You are redirected to the `client_setup.html` file in the `client_files` subdirectory below the `multinet` directory in the Web server's document directory.

- 3 When the QAD NetUI Client Setup window displays, read the NetUI client requirements and verify that the client machine meets them.
- 4 Read the instructions in the Setup Procedure for QAD NetUI section. If you have not previously installed JRE 1.3, click step 1.

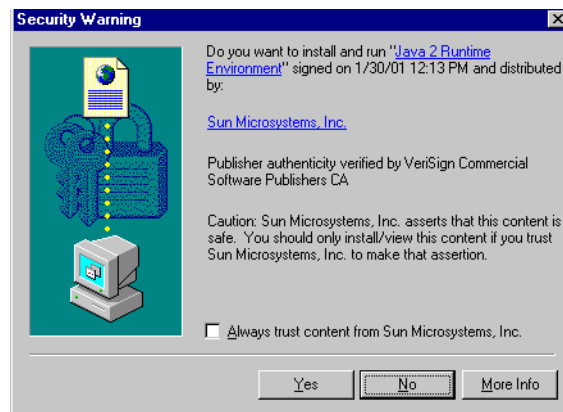


- 5 The following screen displays.



You can choose either option. However, if you choose to save the file to disk before running it, you must find it in the file system before continuing with the next step.

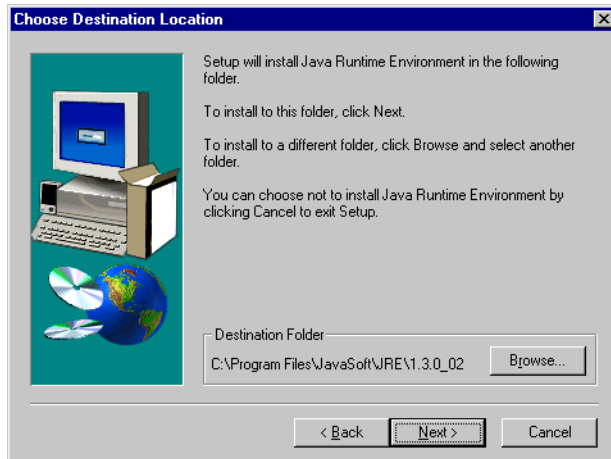
- 6 If you chose to run the install program in the previous step or after double-clicking it to run it from the file system, one or two security warning screens may display. These screens let you agree to accept installations from InstallShield and from Sun. The following figure illustrates the screen associated with Sun Microsystems, Inc.



Note If you have previously installed products from Sun and checked the option to always trust content from Sun Microsystems, Inc., this screen does not display.

Click Yes to install the JRE.

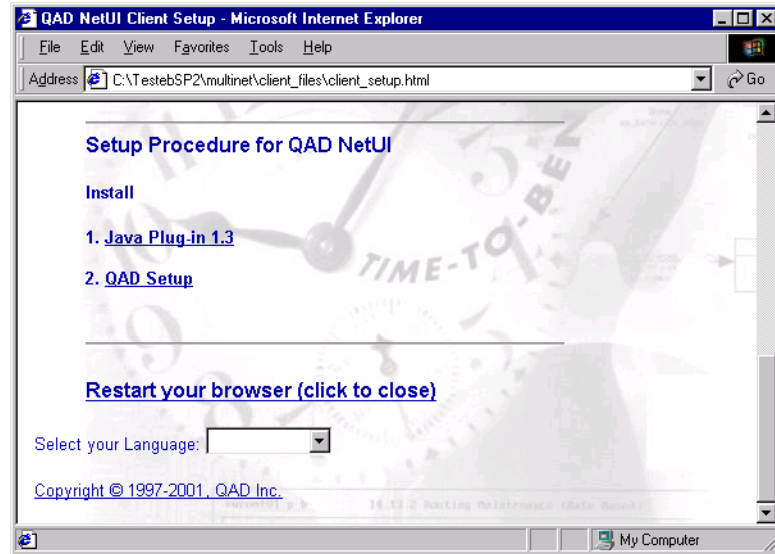
- 7 A number of messages display as the system prepares to install the JRE. When the InstallShield program displays, click Yes to accept the Software License Agreement.
- 8 The Choose Destination Location screen displays. Click Next to install to the default location (recommended). Click Browse to find an alternate location for the installation; then click Next.



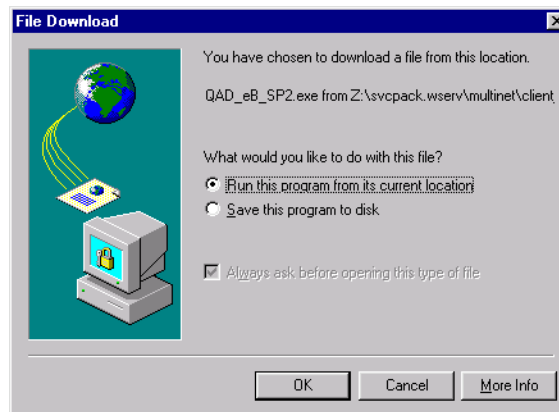
- 9 A number of messages display indicating the progress of the install.
- 10 When the install is complete, your Web browser redispays.

Complete NetUI Setup

- 1 Click the second step in the Setup Procedure for QAD NetUI section of the `client_setup.html` page.



- 2 The File Download screen displays. Click Run this program from its current location; then click OK.



Note If you receive a security warning, simply click Yes to continue.

- 3 A window displays the progress of the setup program. When it completes, close the browser on the client machine and restart it. You can now access the NetUI.

The first time the client successfully logs in to the NetUI, the required .jar files are downloaded. A message displays indicating that the download is in process.

Client Updates

The following list provides brief descriptions of the updates the client setup process performs and of the files that it installs on the client machine:

- Installs the Java Runtime Environment 1.3.0_02, which includes the Java Plug-In.
- Configures Java security files by copying the qadkeystore file and updating the java.policy file so that users can run .jar files signed by QAD. These files are located in:

C:\Program Files\JavaSoft\JRE\1.3.0_02\lib\security

The previous version of java.policy is saved as:

java.policy.orig

Important If you modified policies previously, you should combine the changes in the two policy files.

- Installs a font.properties file with settings needed by the NetUI into:

C:\Program Files\JavaSoft\JRE\1.3.0_02\lib\

- Installs required fonts into:

C:\Program Files\JavaSoft\JRE\1.3.0_02\lib\fonts

- Sets the Java Run Time Parameters field in the Java Plug-in Control Panel to -mx64m.
- Creates a cookie that indicates setup is complete.

Updating Fonts for Non-English Languages

The font.properties file may need to be updated to access locale-specific fonts, such as the Japanese font, on an English environment

platform. A number of alternate `font.properties` files are provided with the JRE, located by default in:

`C:\Program Files\JavaSoft\JRE\1.3.0_02\lib\`

▶ See Table 9.2 on page 161 for a list of language codes.

These files are identified by locale-specific extensions such as `ko` for Korean, `ja` for Japanese, and `zh` for Chinese.

The simplest way to make these fonts available on a client machine is to back up the original `font.properties` file and rename the locale-specific file to `font.properties`. For example, to access Chinese fonts, rename:

`font.properties.zh`

To

`font.properties`

For more information about the `font.properties` file and how to modify it in an international environment, review the following document on the Sun Microsystems Java Web site:

<http://java.sun.com/j2se/1.3.0/docs/guide/intl/fontprop.html>

SECTION 2

Reference

This section provides NetUI system administration and environment reference information.

Configuring Optional User Settings **181**

System Administration **197**

Environment Tables **219**

param91.p 14.13.2 Routing Maintenance (Main Screen)

Routing Code:	10-15000	NONAD (75) CDR IN:
Operation:	20	
Standard Operation		
Work Center	1030	INSPECTION, ALL SITE
Machine		
Description	INSPEC PER PROX-005	
Machines per Op	1	Milestone P
Overlap Units	1	
Queue Time	1.0	
Wait Time	0.0	
Setup Time	0.0	
Route by Production	Build Prod	
Release Time		

Configuring Optional User Settings

This chapter discusses optional NetUI settings you can configure after the initial implementation. These settings let you tailor the NetUI for specific users and groups.

Configuration Overview **182**

Defining User Settings **183**

Defining NetUI URLs **187**

Monitoring User Sessions **192**

Other NetUI Configuration Options **193**

The screenshot shows a configuration window for 'Routing Maintenance (Main Screen)'. It contains a table with the following data:

Routing Code	Operation	Standard Operation	Work Center	Machine	Description	Machines per Op	Overlap Units	Queue Time	Wait Time	Setup Time
10-15000	20	10:30			INSPEC PER PROX-000	1	1	1.0	0.0	0.0

Below the table, there is a section for 'Milestone' and 'Milestone Time'.

Configuration Overview

This chapter includes information about the programs used to configure the NetUI that are located on the NetUI Menu (36.20.10). For example, using functions on the NetUI Menu, you can define uniform resource locator (URL) links to intranet or Internet resources.

Note The UI Configurator and Appearance Manager are discussed in detail in *User Guide: eB Desktop and Network User Interface*.

◆ See *User Guide: Manager Functions* for details.

Some of the manager functions that are used to configure the character and Windows interfaces can also be used to affect the appearance and behavior of the NetUI. These programs are also discussed briefly in this chapter.

Table 10.1 lists the functions available on the NetUI Menu.

Table 10.1
NetUI Menu
(36.20.10)

Number	Program	Menu Label	Notes
36.20.10.1	mgusromt.p	User Option Maintenance	Lets you tailor the NetUI for specific users and groups
36.20.10.2	mgbr062.p	User Option Browse	Lets you browse user option settings
36.20.10.3	mgusrtmt.p	User Option Telnet Maintenance	Discussed in Chapter 7 and Chapter 8.
36.20.10.4	mgusrrp.p	User Option Report	Lets you generate a report of user option settings
36.20.10.8	mgurlmt.p	Menu URL Maintenance	Lets you define the URLs that display in the User URLs folder in the NetUI Program menu
36.20.10.9	mgbr220.p	Menu URL Browse	Lets you browse the URLs in the User URLs folder
36.20.10.11	mgburlmt.p	Browse URL Maintenance	Lets you define the URLs that display in NetUI browses
36.20.10.12	mgbr221.p	Browse URL Browse	Lets you browse the URLs in the NetUI browses
36.20.10.15	mgsessmt.p	Session Master Maintenance	Lets you monitor current NetUI sessions

Number	Program	Menu Label	Notes
36.20.10.17	mguixu.p	UI Configurator	Discussed in <i>User Guide: eB Desktop and Network User Interface</i>
36.20.10.17	mgamxu.p	Appearance Manager	Discussed in <i>User Guide: eB Desktop and Network User Interface</i>

Defining User Settings

Use User Option Maintenance (36.20.10.1) to define settings that control NetUI features and determine which interface options users can configure in browses, inquiries, and reports.

You can use this program to define unique settings for individual users or enter an asterisk (*) in the User ID field to create a generic record of settings. This generic record is automatically associated with every user in the system that does not have a user-specific record.

Note Some of the settings in this program do not apply to eB Desktop. If users are going to access MFG/PRO eB only from eB Desktop, you do not need to define values for Restrict Menu, Menu Substitution, and Program Execution.

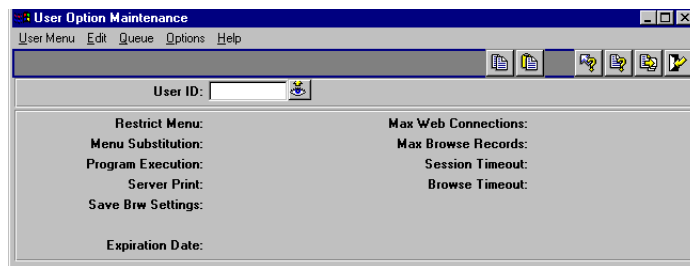
The system is installed with a default record that applies to all users. You can change it to accommodate your own requirements.

You should create separate records for individual users only when appropriate settings vary from user to user or User Restriction is Yes in the NetUI Security Control File (36.3.21.24). When User Restriction is Yes, you must use this program to create a record of settings for each user in the system; you cannot create a generic record and apply it to multiple users.

♦ See *User Guide: eB Desktop and Network User Interface*.

Individual user IDs must be defined in User Maintenance (36.3.18) before you can reference them in User Option Maintenance.

Fig. 10.1
User Option
Maintenance
(36.20.10.1)



Use the following field descriptions to configure the User Option Maintenance settings for your environment.

◆ For more details on the following fields, see *User Guide: eB Desktop and Network User Interface*.

Restrict Menu. This field determines whether programs to which a user does not have access display on NetUI menus when that user logs in to the NetUI.

You can restrict user access to programs as needed using Menu Password Maintenance (36.3.1) or NetUI security features.

No: All programs, including those to which the user does not have access, display in NetUI menus.

Restricted programs display as disabled—that is, light gray—icons and do not execute when selected. Individual users can turn on the Menu Hiding feature to automatically remove all disabled items from the menu.

Yes: Only programs to which the user has access display in interface menus.

Menu Substitution. Enter No to indicate that the associated user cannot enable menu substitution. Enter Yes to indicate that they can.

Program Execution. Enter No to indicate that the associated user cannot execute programs that are not included on the NetUI menu. Enter Yes to indicate that they can.

Server Print. Enter No to indicate that the associated user cannot print to a server-based (remote) printer. Enter Yes to indicate that they can. If this is No, the remote print option is disabled for the user.

Save Brw Settings. Enter Yes to allow the associated user to save custom browse settings or configurations—for example, column order or column hiding settings.

When you save custom browse features, the associated browse displays the same way the next time you execute it.

Expiration Date. Enter a date in this field to indicate that the associated user cannot access the system after this date. Leave this field blank to indicate that an expiration date does not apply to this user.

Max Web Connections. Specify the maximum number of WebSpeed agent and AppServer server process connections that the associated user can use simultaneously. 0 (zero) indicates that no limit applies to this user.

Max Web Connections applies to browses, inquiries, reports, and XUI programs. It also applies to look-ups accessed from another browse, inquiry, report, or telnet program. It does not apply to telnet maintenance and transaction programs.

Web connections are calculated in the following way:

- One connection is used when the NetUI is initially launched. This connection is also used for the first report, browse, or inquiry program accessed or for the first XUI program accessed.
- Each additional report, browse, inquiry, look-up, or XUI program accessed while the first program is active counts as an additional connection.

When a program is closed, the associated connection is released.

Note In the case of XUI programs, the connection is established and released through an AppServer process. If the AppServer shuts down while an XUI program is active, the connection is not released when the program is closed. In this case, you must exit your NetUI session and close the Web browser to release the connection.

In an environment with a limited number of available WebSpeed agents and AppServer server processes, setting Max Web Connections to a high value may result in decreased performance for other users. However, if enough WebSpeed and AppServer processes have been configured system-wide, you should set this value to the maximum number of applicable NetUI programs the associated user is expected to execute simultaneously.

▶ See “WebSpeed Agent Sizing and Behavior” on page 206.

You can use the following guidelines to specify the initial value in this field. As the access patterns of different NetUI users become clear, you can modify this value to fit your environment.

▶ See “Agent Usage for Reports and Inquiries” on page 207.

▶ See “Agent Usage for Browsers and Look-Ups” on page 207.

- For users who typically run short reports or inquiries, assign a lower Max Web Connections value than for users who must simultaneously generate multiple reports or inquiries containing large amounts of data.
- Since it is difficult to predict in advance whether browses will require WebSpeed agents for short or long time periods, you should plan to have one WebSpeed agent available per open browse for high-priority users.
- For the Java-based maintenance programs (XUI), a dedicated AppServer server process is required for each open XUI application.

Max Browse Records. Enter the maximum number of records to display in a browse for this user before notifying the server to retrieve more records. The default value is 0 (zero), which corresponds to a 10,000-record maximum.

This value determines the limit users can set in for the Download Size field in the NetUI client Preferences window. Download size defaults to 50 unless you set this value to less than 50. Individual users can decrease or increase the value. However, increasing the download size to a number greater than the Max Browse Records value has no effect.

Example If you set Max Browse Records to 25 for a user and the user sets the Download Size value in the Preferences window to 100, browse data windows still display only 25 records at a time.

Session Timeout. Specify an amount of time in minutes after which the system automatically logs the associated user out of an inactive NetUI session.

Browse Timeout. Enter the maximum amount of time in minutes during which the associated user can view a state-aware browse without requesting the server to retrieve additional records. When this time period is exceeded, the browse displays a time-out message when the user scrolls to the bottom of the window and does not

retrieve any more records. The user must restart the browse if they need to view additional records.

When a user accesses a browse containing more records than the specified download size, the system retrieves the first set of records and then waits for the user to scroll down to the bottom of the browse window before retrieving the next set of records. If the browse is state-aware during this time, other users cannot use the WebSpeed agent that is waiting to execute this user's request. Setting a Browse Timeout value prevents WebSpeed agents from being inaccessible to other users for extended time periods.

This field cannot be left blank.

Note Do not confuse this parameter with the Timeout Minutes parameter in the standard Security Control File (36.3.24). Timeout Minutes specifies when to log users with inactive sessions out of MFG/PRO eB.

Defining NetUI URLs

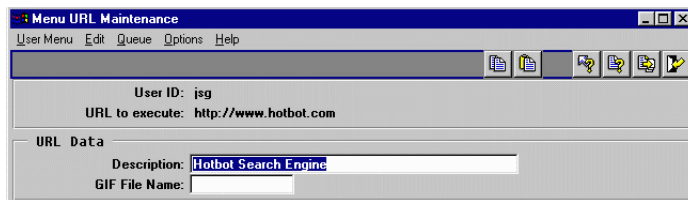
Uniform resource locators (URLs) are text strings that indicate the location of an intranet or Internet resource. In NetUI, you can use URLs to access the resources associated with them.

- Links defined in Menu URL Maintenance (36.20.10.8) are accessible from the User URLs folder in the NetUI Program menu. These links also display in the My Bookmarks folder in the eB Desktop.
- Links defined in Browse URL Maintenance (36.20.10.11) are accessible in the NetUI browse windows.

Define Menu-Level URL Links

Use Menu URL Maintenance to establish URL links that users can access from the User URLs folder in the NetUI Program menu. You can set up URLs that are available to all users or selected individuals.

Fig. 10.2
Menu URL
Maintenance
(36.20.10.8)



Use the following field descriptions to configure the Menu URL Maintenance settings for your environment.

User ID. Enter a user ID to associate with the specified URL link. To associate all users in the system with that link, enter an asterisk (*) in this field. The link only displays in the associated user's User URLs folder.

URL to Execute. Enter a URL to include in the User URLs folder for this user (maximum 132 characters).

Description. Enter a descriptive label to associate with the specified URL. This label displays as a menu item in the User URLs folder for this user.

GIF File Name. Optionally, enter the name of an image file use as an icon for the associated URL. The image file must be in the graphics interchange format (extension .gif). When you enter the name of the file in this field, do not include the .gif extension.

The image file must be located in the `images` subdirectory below the `java` subdirectory, which is in the `multinet` directory in the Web server's document directory. The image size should be 24 by 24 pixels.

If this field is blank, the standard QAD icon is used.

Define URL Links for Browsers

Use Browse URL Maintenance (36.20.10.11) to create URL links that users can activate from NetUI browse data windows.

See *User Guide: eB Desktop and Network User Interface*.

When a browse cell contains a URL link, double-clicking on it launches a Web browser and displays the intranet or Internet resource associated with the URL.

Use Browse URL Maintenance to create links to information that is related to items in the browse, as in the following example.

Example You want to establish a URL link in the Purchase Order Browse from supplier ID GS10100 to the corresponding supplier’s company Web site, located at <http://www.generalsupplies.com>. To do this, enter the following values in Browse URL Maintenance.

Field Name	Value
Browse	pobr006.p
User ID	*
Field	so_vend
Value	gs10100
URL	http://www.generalsupplies.com
Description	General Supplies Web Site
Primary	Yes

Table 10.2
Sample Field
Entries

Using Browse URL Maintenance, you can:

- Associate one or more URLs with a single column or specific field value in a browse.
- Limit access to URLs by user, if required.

URLs can contain special strings that are automatically replaced by field values in the browse. Selecting a link containing this type of string automatically replaces that string with the corresponding field value in the row.

Follow these steps to define this type of special string in a URL:

- 1 Enter #b# to indicate the beginning of the string.
- 2 After the #b#, enter an MFG/PRO field name associated with the specified browse.
- 3 Enter #e# to indicate the end of the string.

Example The Web site for one of your primary suppliers contains a catalog of items. Entering an item's identifier at this Web site accesses the catalog entry for that item, containing information such as item cost, quantity available, and ship weight. To create links from the MFG/PRO supplier item numbers to their corresponding catalog entries at the supplier's Web site, create the following URL:

http://www.generalsupplies.com/catalog/#b#vp_vend_part#e#

Then, associate it with the Supplier Item column in the Supplier Item Browse. After you establish this link, selecting a supplier item number in the Supplier Item Browse automatically inserts the selected field value. For example, selecting supplier item 10-1005 creates this URL:

http://www.generalsupplies.com/10-1005

The system then launches a Web browser to display the relevant catalog information for that item located at that URL address.

Figure 10.3 illustrates Browse URL Maintenance.

Fig. 10.3
Browse URL
Maintenance
(36.20.10.11)

Defined URL Parameters	
Fields In Browse	URL Substitution String

Use the following field descriptions to configure the Browse URL Maintenance settings for your environment.

Browse. Enter the name of the Web-enabled browse program to contain the specified URL link.

Entering an asterisk (*) in this field allows the specified URL to be associated with any Web-enabled browse in the system.

Example To associate a specific URL with the sales order number column in all NetUI browses, enter an asterisk in this field and specify the sales order column's corresponding field name (so_nbr) in Field.

Entering a value in this field automatically displays a list of valid values for Field in the Defined URL Parameters frame.

User ID. Enter a user ID to associate with the specified URL link. To associate all users in the system with that link, enter an asterisk (*) in this field.

Field. Designate the browse column in which you want to establish a URL link by entering the MFG/PRO field name associated with that column. Entering a value in the Browse field automatically displays valid values for this field in the Defined URL Parameters frame.

This field cannot be left blank.

Value. Specifying a value in this field associates the designated URL with every browse cell that contains that value and belongs to the browse column indicated in Field. Enter an asterisk (*) to associate the URL with every cell in the column.

URL. Specify a URL referring to an Internet or intranet location (maximum 132 characters).

Description. Optionally enter a description to display when this URL is selected.

Primary. Enter Yes to indicate that the specified URL is the primary URL for a cell.

In browse data windows, right-clicking on a cell or column containing multiple URLs displays a list of all the URLs associated with that cell or column. The primary link displays at the top of the list and is the default link for the cell.

Tip

You cannot specify both a URL and a URL script.

Tip

You cannot specify both a URL and a URL script.

This value applies to defined (non-scripted) URLs only.

URL Script. To associate the specified user, browse, column, field value, or combination of these with a custom URL script, enter the full path to the directory containing the custom script.

Note Scripts should be based on the supplied template located in the source code directory: `/src/urltempl.p`.

Determine. Enter Yes to have the system run the specified custom URL script upon selection of the associated cell or column to determine whether that cell or column has an associated URL.

When this field is No, the script is not run and the designated column or cell is defaulted to having a URL.

Monitoring User Sessions

Use Session Master Maintenance (36.20.10.15) to view information about users who are currently logged into MFG/PRO through the NetUI. This information displays in the form of session records, each identified by a unique session ID that is generated by the system. Session records are automatically created when a user successfully logs into MFG/PRO from the NetUI and are deleted when they log out.

The information displayed in each session record includes user settings established in User Option Maintenance, User Telnet Option Maintenance, and NetUI security programs. Additional information about the session context also displays, including the logical database name, the current entity, GL entity, and base currency.

Fig. 10.4
Session Master
Maintenance
(36.20.10.15)

The screenshot shows the 'Session Master Maintenance' window with a menu bar (User Menu, Edit, Queue, Options, Help) and a toolbar. The main display area is divided into three sections:

- Session Information:**
 - Session ID: 957909463883
 - User ID: hws
 - User Name: hws
- Session Master:**
 - Client IP Address: 167.3.57.221
 - Session Timeout: 60 Min
 - Active Web: 1
 - Number of Records: 100
 - Security Profile: Idle Time: 10016.52 Min
 - Active Telnet: 0
 - Menu Substitution: yes
- Session Context Detail:**
 - Context ID: 1
 - GL Entity: 1000
 - Logical Database: Current Entity: 1000
 - Base Currency: usd

Other NetUI Configuration Options

Most of the other functions that can be used to tailor the Windows and character user interfaces can also be used with the NetUI. For example, when you use Window Help Maintenance (36.4.21) to add a scrolling window, the new scrolling window displays in all three interfaces. The same is true when you use Drill Down Maintenance (36.20.1) to add drill downs to a field.

However, in some cases, you must set up features in a slightly different way in the NetUI. This section discusses some of the functions that vary when used in the NetUI:

- Menu System Maintenance (36.4.4)
- User Function Maintenance (36.4.11)
- User Tool Maintenance (37.20.2)

Adding New Folders to the Main Navigation Tree

Use Menu System Maintenance (36.4.4) to add new folders that display in the NetUI main navigation tree. The menu groups represented by these folders are referenced through the letter A. For example A.1 is Distribution, A.2 is Manufacturing and so on.

Folders added in this way are available to all users.

Note If you only require one folder, use the user menu to define a set of frequently used programs instead.

To add a new folder, specify A for the menu. Then specify a selection with a number greater than the last folder.

Example You have five accounting clerks who always use the same set of programs. Set up folder Accounting Tasks by entering A for menu, 8 for selection, Accounting Tasks for selection label, and A.8 for the file to execute. Then press Go to save the record. Add functions under this folder by specifying A.8 for menu and the program description and name you want to execute.

Tip

You must log out and log in again to see your changes.

Assigning Programs to the User Menu

Use User Function Maintenance (36.4.11) to define a list of functions:

- To be executed with function keys in the character interface
- To appear on the User Menu in the character and Windows interfaces so they can be accessed directly from other programs
- To display in a separate User Menu folder on the main navigation tree of the NetUI or in the My Programs folder in eB Desktop

Note In the Windows or NetUI interface, you cannot use function keys to execute programs automatically. Instead, the Function Key field determines the order in which programs appear on the User Menu pull-down menu or in the User Menu folder.

You can define a general User Menu as well as one for each user. To set up a general menu for all users, leave User ID blank when adding programs to the User Menu. To set up a user-specific menu, indicate the ID for which you are setting up each user menu item.

Assigning Buttons to the Tool Bar

Use User Tool Maintenance (36.20.2) to create buttons for running programs. Assign buttons by user and program. These buttons display in the tool bar in the Windows interface and in the NetUI. They display in the order in which they are defined.

Tip

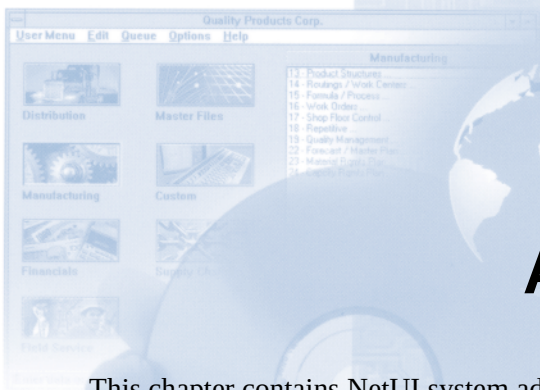
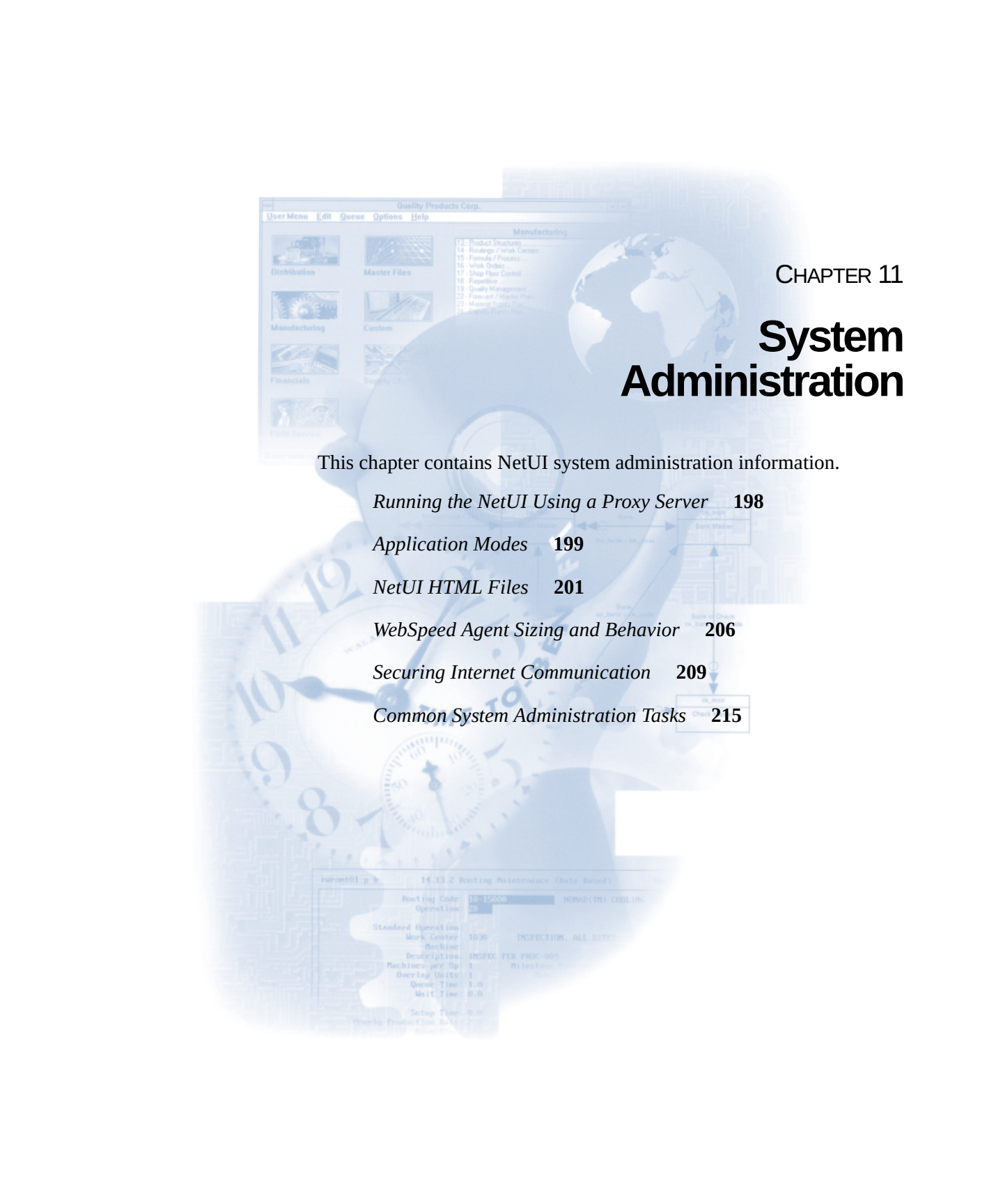
Do not use an extension when specifying the file name.

You can assign a text label to the button and an image file. If you want to use image files, you must assign them differently in the two interfaces:

- For Windows buttons, assign a bitmap (.bmp extension) located in the user's propath. The text label displays on the user menu, but only the image displays on the button.
- For NetUI buttons, assign a Graphics Interchange Format file (.gif extension). Place the GIF in the images subdirectory below the java subdirectory, which is in the multinet directory in the Web server's document directory. The image size should be 24 by 24 pixels. When you assign an image, it displays on the button in the tool bar. The text label can be viewed as a tool tip when your cursor passes over the button.

If you do not assign an image when you create a button:

- The button displays with the text label in the Windows interface.
- The button displays with a default image and the text label as a tool tip in the NetUI.



CHAPTER 11

System Administration

This chapter contains NetUI system administration information.

Running the NetUI Using a Proxy Server **198**

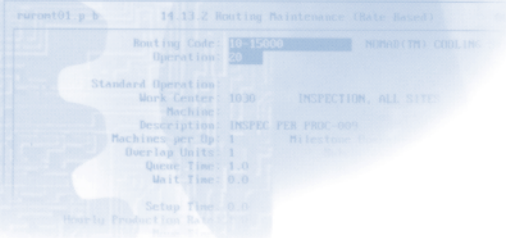
Application Modes **199**

NetUI HTML Files **201**

WebSpeed Agent Sizing and Behavior **206**

Securing Internet Communication **209**

Common System Administration Tasks **215**



Running the NetUI Using a Proxy Server

To run the NetUI using a proxy server to access the applicable Web site, you must configure proxy settings in the Java Plug-in Control Panel on each client. In addition, you must add proxy server information to the NetUI HTML files and Web browser preferences.

- 1 Open the Java Plug-in Control Panel. From the Windows Start button, access Programs|Java Plug-in Control Panel.
- 2 In the Java Plug-in Properties window, click the Proxies tab and uncheck the Use Browser Settings check box.
- 3 Complete the Proxy Address and Port fields for HTTP.
- 4 Click Apply and close the control panel.
- 5 Add the proxy IP address and port number to the appropriate NetUI HTML files.
- 6 Edit your Web browser preferences to connect to the Internet using the proxy server's IP address and port number.

For example, `http://192.199.200.31/netui/full.html`

Note The NetUI connects to the proxy server using the settings in the Java Plug-in Control Panel. Your Web browser connects to the proxy server using its own Preferences (Netscape) or Internet Options (Internet Explorer) settings.

- 7 For these changes to take effect, you must restart *all* browser sessions.

▶ See “NetUI HTML Files” on page 201.

Application Modes

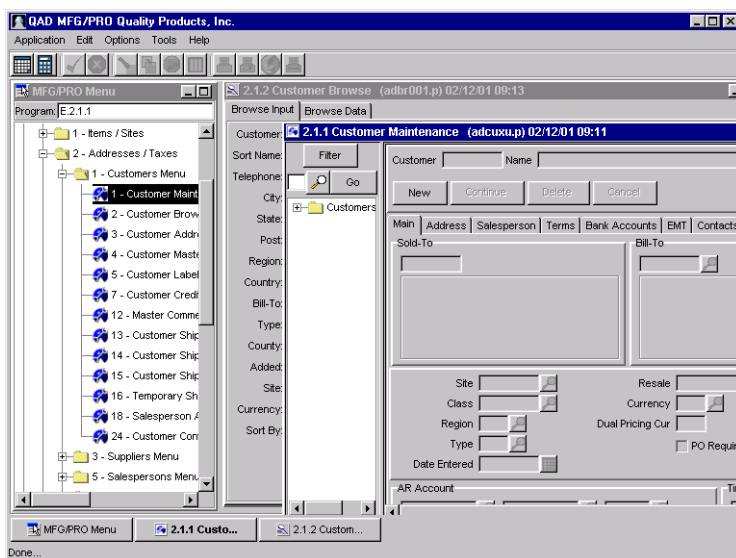
The NetUI can be used in two basic ways: directly with the full application mode or indirectly with the embedded application mode. The NetUI application mode is determined by the HTML file the user specifies when connecting to the NetUI Web server.

Full Application Mode

The full application contains features such as a tool bar, multiple windows, and an expanding tree menu structure. Figure 11.1 is an example of the NetUI in full application mode.

For more information, see *User Guide: eB Desktop and Network User Interface*.

Fig. 11.1
Full Application Mode



Embedded Application Mode

The embedded application mode enables you to build Web pages with links to specific MFG/PRO programs. It has no menu capabilities, and the linked programs are embedded in the Web page. Figures 11.2 and 11.3 are examples of the NetUI in embedded application mode.

For more information, see *User Guide: eB Desktop and Network User Interface*.

Fig. 11.2
Sample Web Page
with Links to
Embedded
MFG/PRO
Programs

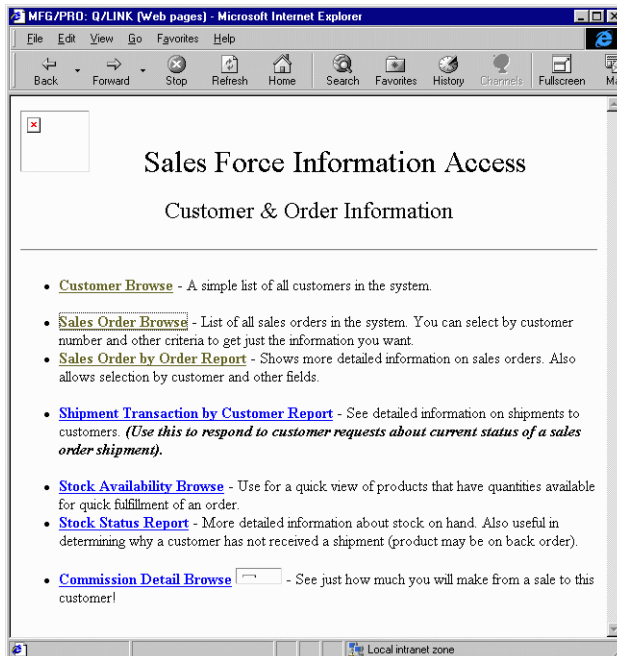
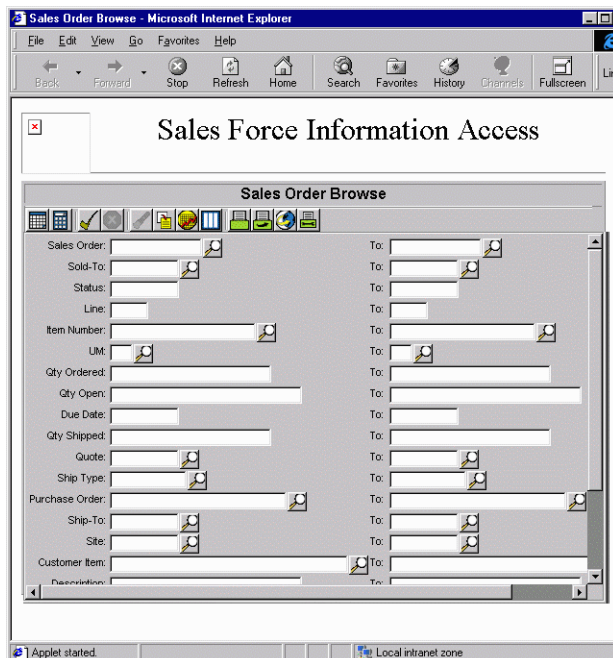


Fig. 11.3
Embedded Browse



NetUI HTML Files

The NetUI HTML files are the starting point for NetUI sessions. The HTML file a user specifies when they launch the NetUI defines many aspects of the session. These include:

- Application mode
- Session trace information
- Internet security
- Proxy access

Default HTML Files

The NetUI software includes a set of default HTML files. You can copy and modify these files to customize the NetUI to meet your requirements. The NetUI HTML files are located in the `multinet` subdirectory below the Web server's primary document directory.

Note The files with 13 appended to the name are added when you install Service Pack 2, which requires Java 1.3. You must use these files after installing Service Pack 2. The other files are not overwritten by the service pack so that you can preserve any customizations you have made.

- `full.html` or `full13.html`
- `example.html`
- `embedded.html` or `embedded13.html`
- `title.html`
- `multinet.html` or `multinet13.html`
- `index.html`
- `trace.html` or `trace13.html`
- `default.htm`
- `version.html` (obsolete)

The following sections describe each of the default NetUI HTML files and their functions.

`full.html`

Use this file to launch the NetUI in full application mode. When this file is accessed, it performs the following functions:

- Launches a new browser window in which the QAD Java applet loads
- Launches `multinet.html` to begin the NetUI session

♦ See “Full Application Mode” on page 199.

By default, this is the NetUI HTML file that users access to begin NetUI sessions.

embedded.html

◆ See “Embedded Application Mode” on page 199.

Use this file to launch the NetUI in embedded application mode. This file launches the NetUI Java code (`multinet.jar`) in the embedded application mode as a single program in the Web browser.

Note This is not the same as running the eB Desktop, which is also an embedded application. eB Desktop has its own startup file.

multinet.html

This file launches the NetUI Java code (`multinet.jar`) in the full application mode.

trace.html

Use this file to launch the NetUI in full application mode with the session trace feature enabled. When this file is accessed, it performs the following functions:

- Launches a new browser window in which the QAD Java applet loads
- Launches `multinet.html` to begin the NetUI session
- Enables more verbose messaging between the NetUI session and the Java Plug-In Console

Because the session trace messages are sent to the Java Plug-In Console, you must enable the console before accessing `trace.html`. The Java Plug-In Console can be helpful for troubleshooting. It displays pertinent information such as:

- User’s home directory
- Version of Java in use
- Proxy settings, if any
- Data and messages being transferred between the NetUI client and the WebSpeed and AppServer instances
- Error messages

To enable the console, complete the following steps:

- 1 Open the Java Plug-in Control Panel using the method appropriate for your version of Java:
 - a For Java Plug-In 1.1.3, click the Windows Start button and access Programs|Java Plug-in Control Panel.
 - b For Java Plug-In 1.3, click the Windows Start button and choose Settings|Control Panel. Then double-click on the Java icon in the Control Panel.
- 2 When the Java Plug-in Control Panel displays, click the Basic tab and select the Show Java Console option.
- 3 Click Apply and close the control panel.

Note For these changes to take effect, you must restart the browser.

- 4 Using the client browser, access `trace.html` file. A Java console window displays. The session trace messages display in this window as actions are performed in the NetUI.

Note You can use the Java console with other NetUI HTML files; however, the messaging is not as complete.

- 5 To turn off the Java console, deselect the Show Java Console option in the Java Plug-in Control Panel.

version.html

This file is obsolete and no longer used.

example.html

This file provides a template for creating an HTML page from which you can launch the NetUI in the embedded application mode. This file contains references to the `embedded.html` and `title.html` files.

♦ See “Embedded Application Mode” on page 199.

In addition to the `example.html` file, QAD supplies a set of HTML files that illustrate how the NetUI embedded application mode can be used to enable a process flow for a fictional company. These files are located in the `examples` subdirectory below the `multinet` directory in the Web

server's document directory. You can use these files as the basis for designing your own process flows.

title.html

This file provides the QAD company logo in `example.html`.

index.html

This file provides directory browsing from the client Web browser for the `multinet` subdirectory on the Web server. This file is used with Netscape Web browsers.

default.html

This file provides directory browsing from the client Web browser for the `multinet` subdirectory on the Web server. This file is used with Microsoft Web browsers.

HTML File Parameters

Before looking at the HTML parameters, you should understand some important points about the NetUI HTML files.

In general, an HTML file has the following format.

```
<html>
<head> . . . </head>
<body>
    (Company-specific information)
    <OBJECT>
        (Parameters for NetUI specific to Internet Explorer)
        <COMMENT>
            <EMBED>
                (Parameters for NetUI specific to Netscape)
            </EMBED>
        </COMMENT>
    </OBJECT>
</body>
</HTML>
```

Only lines between `<OBJECT>` and `</OBJECT>` are specific to the NetUI. This section is divided into two parts—one specific to Internet Explorer and the other specific to Netscape. The same parameters apply to Internet Explorer and Netscape; only the syntax is different.

The parameters in HTML files used to access the NetUI differ from file to file depending on whether:

- The NetUI is run in the full or embedded application mode.
- The session trace feature is enabled.

Required Parameters

The HTML parameters *always* required by the NetUI are listed in Table 11.1.

Parameter	Description
WIDTH	The width in pixels of an applet as it appears on an HTML page.
HEIGHT	The height in pixels of an applet as it appears on an HTML page.
code	The name of the main Java program. This is always <code>QadApplet.class</code> (case-sensitive).
codebase	The directory location for the Java code file (extension <code>.jar</code>). By default, this directory is in the <code>Java</code> subdirectory, which is in the primary document directory on your Web server.
FileName	The name of the file containing the Progress code that starts MFG/PRO for the NetUI. The format is: <i>CGIDirAliasName/CGIExecutable/Broker/Program</i> The format components are: <i>CGIAliasName</i> – The path to the MFG/PRO Java code or a CGI directory alias as defined on the Web server. <i>WebSpeedBroker</i> – WebSpeed broker name. <i>Program</i> – Web-enabled MFG/PRO program. Example: <code>"/cgi-bin/wsis.dll/WService=netui/mfwb01.p"</code>

Table 11.1
Required HTML
Parameters

Optional Parameters

The parameters in Table 11.2 are used to enable optional NetUI features.

Table 11.2
Optional HTML
Parameters

Parameter	Description
Program	Determines whether the NetUI runs in full or embedded interface mode. When the Program parameter is present, embedded interface mode is used. This parameter is then followed by the name of the program the interface will run. If the Program parameter is not present in the .html file, the NetUI runs in full interface mode.
Trace	Sets the session trace option.
archive	The name of the JAR file that contains the Java code.
HostName	The host name or IP address of the Web server.
PortNumber	The port number of the Web server.
RMI Host	The host name or IP address of an RMI server. When present, this parameter overrides the RMI host specified in the label master table.
RMI Port	The port number of the RMI service. When present, this parameter overrides the RMI port specified in the label master table.

WebSpeed Agent Sizing and Behavior

WebSpeed agents are used to extract information from the database for Web-enabled browses, inquiries, and reports. Each WebSpeed agent services one user request at a time. WebSpeed agents also extract the data for look-ups in programs using the extended user interface (XUI).

When calculating the number of WebSpeed agent licenses required for your site, you should consider the following:

- The number of reports and large browses being run simultaneously
- The number of seconds of inactivity before an agent is automatically released
- The number of simultaneous activities each end user is allowed
- The mix of embedded UI and full UI users

A very approximate rule is to allow one agent for every eight possible embedded UI users and one agent for every two full UI users.

For most activities, the agent is only used briefly to retrieve the information required to present or update the user interface.

Table 11.3 shows the approximate amount of time during which WebSpeed agents are held for various combinations of NetUI activities and agent states.

Activity	Agent State	Time Agent Held
Building log-in screen	Stateless	Less than one second
Building menu or desktop	Stateless	Less than one second
Navigating down menu	Stateless	Less than one second
Starting browse	Stateless	Less than one second
Starting report	Stateless	Less than one second
Running report	Stateless	While report is generating data
Running a browse for which the number of records is less than the Download Size	Stateless	Less than one second
Running a browse for which the number of records exceeds the Download Size	Stateless	Less than one second
Running a browse for which the number of records exceeds the Download Size	State-aware	While browse is incomplete or until timeout is reached

Table 11.3
Summary of Agent Usage

Agent Usage for Reports and Inquiries

For reports and inquiries, a WebSpeed agent is in use—that is, it is servicing a user request—when it is extracting report or inquiry data from the database. After delivering this data to the Web server, the WebSpeed agent is free to service another request, even if the NetUI has not yet displayed all the data to the user.

Agent Usage for Browsers and Look-Ups

For browses and look-ups, a WebSpeed agent is in use when it is delivering the number of records specified as the Download Size parameter in the NetUI client Preferences window.

- If the browse request generates fewer records than the Download Size parameter, then the records display in the browse and the agent is released immediately.

♦ See *User Guide: eB Desktop and Network User Interface*.

Example If Download Size is 100 and a browse query returns 80 records, MFG/PRO displays the 80 records and releases the agent.

When download size is small, this might require only a few seconds per request. This means that a WebSpeed agent might be busy only a fraction of the time a user has a browse data window open in the NetUI.

- If the browse request generates more records than the Download Size parameter, the agent is sometimes held the entire time the browse data window is open—that is, from the time the user requests the data to the time the browse times out or is manually closed. A browse in this condition is known as *state-aware*. This occurs when:
 - The user sorts browse data by a field that is not indexed in the database. Indexed fields are indicated by an asterisk (*) in the Sort By field.
 - The WebSpeed agent returns a group of data that ends with a non-unique key.
 - The browse takes an unusually long time to begin returning data to the Web server.

When a browse is state-aware, it displays a time-out message in the title bar—for example, `Browse times out in 2 minutes`.

Example The Download Size parameter is 100 and the query returns 200 records, the browse data window displays approximately the first 100 records. It may actually display slightly more than 100 records, depending on where an appropriate break in the active index is found. When the user scrolls down to the bottom of the data window, the browse requests the next 100 records for display.

When any of the conditions above are met, the agent becomes state-aware and is held until the browse is closed or times out. In this case, the browse data window displays the following message to indicate that the agent is being held:

`Times Out in X Minutes (Port #).`

Mechanisms to Optimize Agent Usage

Several settings in the NetUI can be used to assist in optimizing WebSpeed agent usage. Define these settings in User Option Maintenance (36.20.10.1) for individual MFG/PRO user IDs.

▶ See “Defining User Settings” on page 183.

Max Web Connections. Use this setting to indicate the maximum number of agents that can be simultaneously locked by the associated user. When this amount is exceeded, an error message displays and the user cannot run a report or a browse until they have released agents.

Browse Timeout. Use this setting to indicate the maximum amount of time an agent remains locked for the associated user before it is automatically released and is available to other users.

Session Timeout. Use this setting to indicate the maximum amount of time in minutes that a NetUI session can be inactive for the associated user before the system automatically logs them out.

Securing Internet Communication

The NetUI lets clients view MFG/PRO report and browse programs over the Internet. The Internet traffic between the NetUI client and the Web server is not secure and, with the use of a network sniffer, security could be compromised.

The Secure Sockets Layer (SSL) protocol provides encryption for NetUI transmissions between the client and the Web server. Use the following instructions to enable the SSL applet provided with the NetUI.

The use of SSL security has the following limitations:

- SSL security does not secure NetUI telnet communications.
- SSL security does not secure XUI programs.
- SSL does not secure communications between the NetUI Web server and the MFG/PRO database server. If these servers are different machines that communicate over a network, their communications may also be compromised.
- The NetUI SSL applet supports only Netscape Web servers.

Secure Sockets Layer

Secure Sockets Layer (SSL) is a program layer that secures message transmissions between the NetUI client and the Web server. The SSL security programming is contained in the SSL applet, which is included with the NetUI software.

The SSL applet functions between the QAD applet and the NetUI Web server. The sockets method is used to pass data back and forth between the SSL applet, the NetUI client, and a low-level component of the Secure Hypertext Transfer Protocol (HTTPS) on the NetUI Web server.

SSL uses the public-and-private key RSA (Rivest-Shamir-Adleman) encryption system, which requires the use of a digital certificate.

The client components of SSL are included as part of the Netscape and Internet Explorer Web browsers. If you are using a Netscape Web server, SSL can be enabled and specific Web pages can be identified as requiring SSL access.

SSL Security in the NetUI

During a NetUI session, the NetUI HTML launches the QAD applet, which runs locally on the client machine. When an SSL applet tag is included in the NetUI HTML, the SSL applet on the Web server is accessed and run locally on the client.

Functioning between the QAD applet and the NetUI Web server, the SSL applet receives and transmits messages to and from the QAD applet using a network port on the client machine. Then, the SSL applet uses a secure Web service and port to transmit and receive messages to and from the NetUI Web server. This ensures security for all transmissions from the Web server to the client.

The NetUI installation process places the QAD applet in the `multinet.jar` file, which is located in the following directory on each client machine:

`C:\Program Files\Java Plug-in 1.1\lib\applet`

The SSL applet is placed in the `java` subdirectory below the `multinet` directory in the Web server's document directory. The SSL applet has two file formats:

- `Ssl.jar` is used with Netscape's Navigator or Communicator browsers.
- `Ssl.cab` is used with Microsoft's Internet Explorer.

If both of the SSL applet file formats are in the appropriate location and the standard NetUI HTML file is used, the browser automatically selects the correct format.

SSL Setup Overview

Summary

The following major steps are required to enable SSL security for the NetUI. Each of these steps is described in detail in the following sections.

- Obtain an RSA authentication certificate.
- Set up a secure Web service for the SSL applet on the Web server.
- Identify a port on the NetUI client for communications between the QAD and SSL applets.
- Identify the HTML file or files NetUI clients use for NetUI sessions.
- Create an SSL applet tag block in the NetUI client HTML file or files.

Prerequisites

- 1 Obtain an RSA authentication certificate using the instructions on the VeriSign Web site:

<http://www.verisign.com/>

- 2 Use your Web server's documentation to set up a secure Web service. NetUI SSL security requires both secure and unsecured Web services and ports.

The QAD applet and the NetUI Java Plug-In's Java Virtual Machine (JVM) access the Web server using an unsecured port, and do not accept secure Web services. The SSL applet and the Web browser's default JVM access the Web server through a secure port to send and receive encrypted transmissions from the Web server.

- 3 Identify a port on the NetUI client for communications between the SSL and QAD applets.

Because you specify this port in the NetUI HTML files and these files are shared by all NetUI clients, you must identify a standard port to

use for all of the NetUI clients. QAD has tested and recommends port 3333.

Set Up SSL

- 1 Identify the NetUI HTML file or files being used by the NetUI clients; for example, `full.html` for clients using the NetUI in full application mode. The HTML files are located in the `multinet` subdirectory below the Web server's document directory.

Because the specific HTML file used by each NetUI client may vary, these instructions refer to this file generically as `client.html`.

Note If some clients use multiple NetUI HTML files—for example, `trace.html` or `embedded.html`—be sure to complete these instructions and add the SSL applet tag block to each HTML file for which you want to provide security.

- 2 Open the `client.html` file in a text editor.
- 3 Copy the following template into your `client.html` file to create an SSL applet tag block. This template contains parameters that require information unique to your NetUI environment. These parameters are identified with bold-print placeholders. Use the directions in the next step to enter the required values for each placeholder:

```
<!-- SSLApplet -->
<APPLET CodeBase="java" Archive="Ssl.jar" Code="SSLApplet.class"
  WIDTH=400 HEIGHT=1>
  <PARAM NAME = CabBase VALUE = "Ssl.cab">
  <PARAM NAME = HostName VALUE = "WebServer.Domain.com">
  <PARAM NAME = SecurePort VALUE = "Client Port Number">
  <PARAM NAME = PortNumber VALUE = "Secure Web Service Port Number">
  <PARAM NAME = Trace VALUE = "Off">
  <PARAM NAME = FileName VALUE = "/CGI Directory/WebSpeed CGI Script
    Name/wService=WebSpeed Broker Name">
</APPLET>
<!-- END SSLApplet -->
```

- 4 Use the following descriptions of the SSL applet parameters to replace each placeholder in the tag block with the required environment-specific information.
 - a **HostName** (`WebServer.Domain.com`): Enter the full domain name of the NetUI Web server.

Note If you do not enter a full domain name in this parameter, clients using Microsoft's Internet Explorer browser will receive fatal errors.

- b** **SecurePort (Client Port Number):** Specify the NetUI client port reserved for communication between the QAD and SSL applets. ▶ See step 3 on page 211.
 - c** **PortNumber (Secure Web Service Port Number):** Enter the port number of the secure Web service on the NetUI Web server. If you do not enter a value, the SSL applet uses port number 443. This causes errors unless the secure Web service has been set up on this port. ▶ See step 2 on page 211.
 - d** **FileName (/CGI Directory/WebSpeed...):** Copy the FileName parameter value from the QAD applet tag block and replace the FileName placeholder in the SSL applet tag block with this value. Use the example in step 5 to locate the QAD applet tag block.
- 5** In the `client.html`, locate the QAD applet tag block. Use the following example to identify this tag block in the HTML file. The QAD applet tag block contains parameters that require information specific to your NetUI environment. These parameters are identified with bold-print placeholders. Use the directions in the next step to enter the required values for each placeholder:

```
<!-- QADApplet -->
<OBJECT classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
    WIDTH = 400 HEIGHT = 200>
<PARAM NAME = Code VALUE = "QadApplet.class">
<PARAM NAME = CodeBase VALUE = "java">
<PARAM NAME = Archive VALUE = "multinet.jar">
<PARAM NAME = HostName VALUE = "WebServer.Domain.com">
<PARAM NAME = PortNumber VALUE = "Secure Web Service Port Number">
<PARAM NAME = SecurePort VALUE = "Client Port Number">
<PARAM NAME = FileName VALUE = "/CGI Directory/WebSpeed CGI Script
Name/WService=WebSpeed Compiled Code">
<COMMENT>
    <EMBED type="application/x-java-applet;version=1.1"
        java_Code = "QadApplet.class"
        java_CodeBase = "java"
        java_Archive = "multinet.jar"
        WIDTH = 400
        HEIGHT = 200
        HostName = "WebServer.Domain.com"
        PortNumber = "Secure Web Service Port Number"
        SecurePort = "Client Port Number"
        FileName = "/CGI Directory/WebSpeed CGI Script
```

```

        Name/WService=WebSpeed Compiled Code">
        <NOEMBED></NOEMBED></EMBED>
    </COMMENT>
</OBJECT>
<!-- END QADApplet -->

```

- 6** Use the following descriptions of the QAD applet parameters to replace each placeholder in the tag block with the required environment-specific information. Because the QAD applet was written to function with both Internet Explorer and Netscape browsers, the environment-specific variables must be entered in two locations of the tag block:

- The <OBJECT> section
- The <EMBED> section

▶ See step 3 on page 211.

- a** **SecurePort (Client Port Number):** Specify the NetUI client port reserved for communication between the QAD and SSL applets.

▶ See step 2 on page 211.

- b** **PortNumber (Secure Web Service Port Number):** Enter the port number of the secure Web service on the NetUI Web server.

If you do not enter a value, the QAD applet uses port number 443. This causes errors unless the secure Web service has been set up on this port.

Verify SSL Security

Follow these instructions to verify that the SSL applet is functioning.

- 1** In the `client.html`, set the Trace parameter value in the SSL applet tag block to On.

```

<!-- SSLApplet -->
<APPLET CodeBase="java" Archive="Ssl.jar" Code="SSLApplet.class"
        WIDTH=400 HEIGHT=1>
    <PARAM NAME = CabBase VALUE = "Ssl.cab">
    <PARAM NAME = HostName VALUE = "WebServer.Domain.com">
    <PARAM NAME = SecurePort VALUE = "Client Port Number">
    <PARAM NAME = PortNumber VALUE = "Secure Web Service Port Number">
    <PARAM NAME = Trace VALUE = "On">
    <PARAM NAME = FileName VALUE = "/CGI Directory/WebSpeed CGI Script
        Name/WService=WebSpeed Compiled Code">
</APPLET>
<!-- END SSLApplet -->

```

- 2 From a NetUI client, use the browser's Java console to verify that the SSL applet is functioning. Use the instructions appropriate for your NetUI client's browser.
 - For Netscape browsers, select the Communicator menu and choose the Java Console.
 - For Internet Explorer, select the View menu and choose the Java Console.

As NetUI transactions are performed, SSL applet functions display in the Java console.

When you have verified that the SSL applet is functioning, open the `client.html` file and set the Trace parameter value to Off. This eliminates the extra processing associated with session traces and improves performance.

Common System Administration Tasks

This section provides information about common system administration tasks you may need to perform for the NetUI.

RMI Server Administration

Two activities can create files on the RMI server. In both cases you may need to delete obsolete user files.

UI Configurator

The UI Configurator (36.20.10.17) lets you modify the presentation of XUI programs. When you modify a program for a specific user, the presentation is saved in a user-specific subdirectory below the RMI installation directory.

◆ See *User Guide: eB Desktop and Network User Interface*.

While NetUI users can create user-specific subdirectories and presentation files within those directories, they cannot delete these subdirectories. To delete user-specific subdirectories in the RMI installation directory, you should establish procedures for removing these directories from the RMI server.

Error Logging

The system creates text files for tracking Java exceptions. These are located by user ID in the `Error` directory on the RMI server. You may need to remove obsolete files or monitor file size for existing users.

NetUI Output Administration

When NetUI clients output MFG/PRO reports to the screen, the number of pages sent to the client is determined by the Max Pages setting for the JavaUI printer in Printer Setup Maintenance (36.13.2).

By default, the Max Pages field for the JavaUI printer is set to zero, unlimited. This may cause the NetUI client to run out of memory if too many pages are output at once. You should set the Max Pages field to a reasonable number of pages for your NetUI client machines.

Memory Management

When NetUI clients run MFG/PRO reports, insufficient memory errors can occur if the Java Run Time Parameters are not set correctly. In this case, an error dialog displays in the NetUI Java applet indicating insufficient memory to complete the operation.

If this occurs, use the following instructions to increase the amount of memory available to the Java Runtime Environment (JRE).

▶ See “Client Updates” on page 171.

Note The installation sets this value correctly, but it may be changed by individual users.

- 1 Open the Java Plug-in Control Panel using the method appropriate for your version of Java:
 - a For Java Plug-In 1.1.3, click the Windows Start button and access Programs|Java Plug-in Control Panel.
 - b For Java Plug-In 1.3, click the Windows Start button and choose Settings|Control Panel. Then double-click on the Java icon in the Control Panel.
- 2 Click the Basic tab and access the Java Run Time Parameters field.

- 3 Set the memory allocation for the JRE to 64 MB by entering 64m after the -mx prefix in the Java Run Time Parameters field.

MFG/PRO Database Set Administration

This guide includes instructions for implementing NetUI clients using a single database set: the Production database set. If you want to provide NetUI clients with access to additional database sets—for example, the Training database set—you cannot share the same administration database between the sets.

Environment Tables

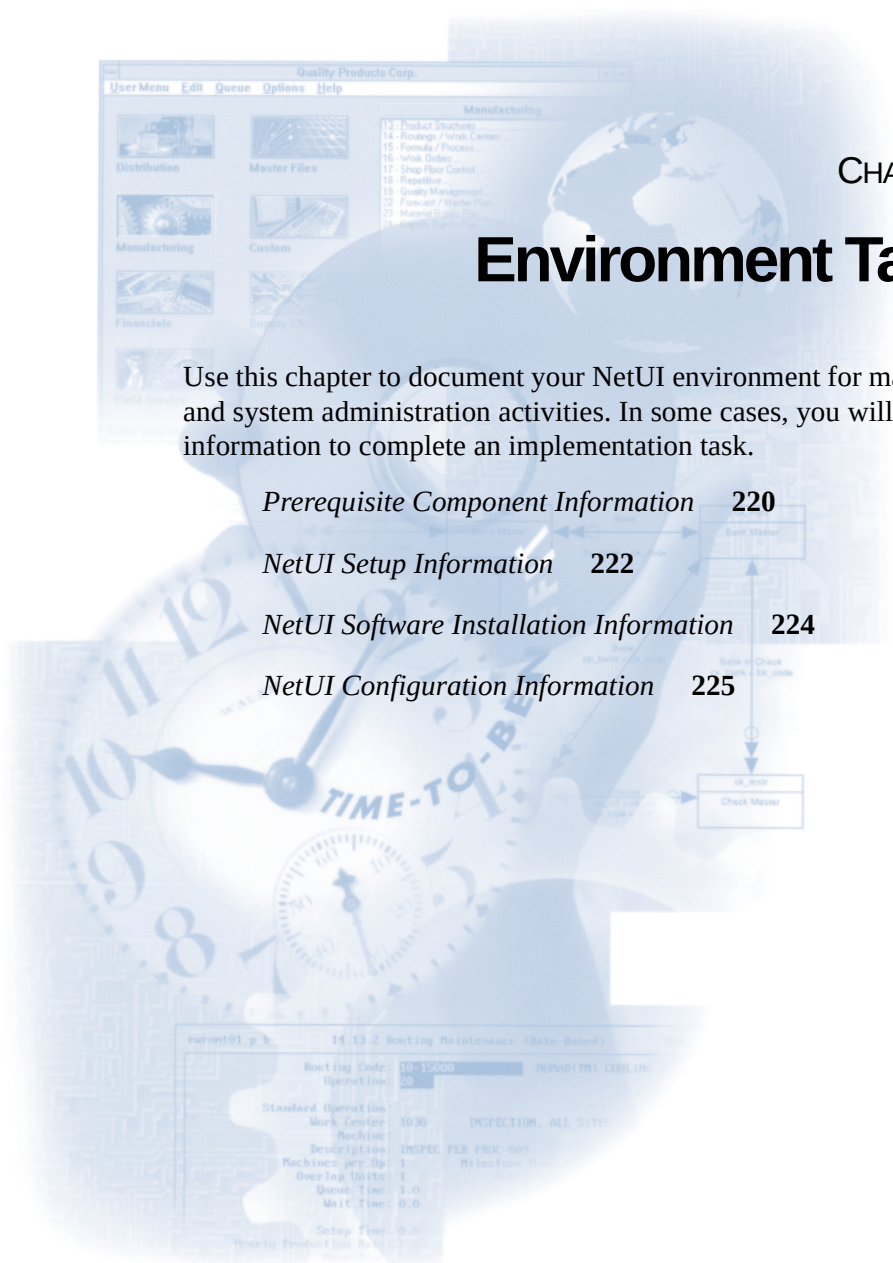
Use this chapter to document your NetUI environment for maintenance and system administration activities. In some cases, you will need this information to complete an implementation task.

Prerequisite Component Information **220**

NetUI Setup Information **222**

NetUI Software Installation Information **224**

NetUI Configuration Information **225**



param91.p 14.13.2 Routing Maintenance (Main Screen)

Routing Code:	10-15000	NONAD(75) CDR IN:
Operation:	20	
Standard Operation		
Work Center	1030	INSPECTION, ALL SITE
Machine		
Description:	INSPEC PER PROX-005	
Machines per Op	1	Allocation:
Overlap Units	1	
Queue Time	1.0	
Wait Time	0.0	
Setup Time	0.0	
Ready for Production	0.0	

Prerequisite Component Information

Before you begin installing the NetUI, fill in Table 12.1 with the configuration information for the prerequisite components.

Table 12.1
NetUI Prerequisite
Information

Required Information	Your Setting
MFG/PRO Progress software directory Notes Record the path to the directory containing your Progress software. You will need this information when you install the NetUI software.	
MFG/PRO database server administration directory Notes This is the directory into which you installed the MFG/PRO database server files and character client code. For more information, see the MFG/PRO installation guide for your system.	

Required Information	Your Setting
MFG/PRO startup shortcut or script Notes Record the location of the MFG/PRO startup script or shortcut for the Production database set. You will need this when you setup the NetUI environment.	
Telnet server • Host Machine • Operating System • Port Number Notes If you are implementing the telnet server on a Windows NT machine, gather this information after you install the Georgia Softworks telnet server software. See Chapter 8, “Configuring Telnet on Windows NT,” on page 131. If you are implementing the telnet server on a UNIX machine, gather this information before you begin the NetUI implementation. You can use the native telnet server on the UNIX server and do not need to install additional software. You will need this information when you configure the telnet server for your NetUI environment.	

NetUI Setup Information

Fill in Table 12.2 as you perform the setup activities in Chapter 4, “Setting Up the NetUI Environment,” on page 55.

Table 12.2
NetUI Setup
Information

Required Information	Your Setting
Web server • Executables directory alias (for example, cgi-bin) • Executables directory (for example, /netscape/cgi-bin) • HTTP port number (for example, 80) • Document directory (for example, /netscape/docs) Notes You must install a Web server before you install the Progress WebSpeed component. The WebSpeed installation requires a Web server. You will need this information when you install the NetUI files.	
Java Development Kit (JDK) directory Notes Record the JDK installation directory. On NT systems, the JDK may be installed with Progress in \$DLC/jdk. On UNIX systems, you download and install it. Depending on your NetUI deployment, you may need to install the JDK on more than one machine. You will need this information when you configure the NetUI environment.	
NetUI parameter file (.pf) location Notes Record the directory where the NetUI .pf file is located. You will need this information when you set up the Progress servers.	

Required Information	Your Setting
NetUI Progress software directories • NameServer • AppServer • WebSpeed Notes In most cases, all of the Progress software components are installed in the same directory. However, you can install the Progress components in different directories to facilitate a dispersed deployment.	
Progress NameServer Port Notes Record the port number of the controlling NameServer for the WebSpeed server and AppServer instances. This guide uses the default NameServer provided by Progress, NS1. The port for this NameServer is 5162. You will need this information when you configure the NetUI environment.	
WebSpeed server instance • Broker name • Broker port number Notes Collect this information when you set up your NetUI WebSpeed server instance. You will need this information when you install the NetUI software.	
Primary AppServer instance • Broker name • Broker port number Auxiliary AppServer instance • Broker name • Broker port number Notes Collect this information when you set up your NetUI AppServer instances. You will need this information when you configure the NetUI environment.	

NetUI Software Installation Information

Fill in Table 12.3 to keep track of the QAD-supplied NetUI software components in your environment. Depending on your deployment, these components may be located in different directories on one or multiple machines.

Table 12.3
NetUI Software
Information

Required Information	Your Setting
Path to the <code>multinet</code> directory Notes The NetUI software files are installed in a <code>multinet</code> directory below the Web server's document directory. You will need to supply the path to subdirectories below <code>multinet</code> when you configure the NetUI environment.	
Path to the <code>rmi</code> directory Notes To set up the RMI server, you must copy the <code>rmi</code> directory onto the machine where the RMI server will run. You should record the location of the <code>rmi</code> directory for reference.	

NetUI Configuration Information

Fill in Table 12.4 as you perform the configuration activities in Chapter 9, “Configuring the NetUI Environment,” on page 149.

Required Information	Your Setting
RMI server • Host machine name • RMI server port Notes You will need this information when you configure the NetUI environment.	
NetUI Online Help • Help Host • Help Script Notes By default, the NetUI online help is installed in the Web server’s document directory. In this case, the help host is the Web server. By default, the help script is <code>ebhelp.ksh</code> or <code>ebhelp.exe</code> depending on the Web server’s operating system. You will need this information when you configure the NetUI environment.	

Table 12.4
NetUI
Configuration
Information

Glossary

Accelerator Keys. Keystroke equivalents to mouse actions. For example, you can exit the NetUI by selecting Exit from the Application menu or by pressing Alt+X. Accelerator key commands have initial default settings; however, in NetUI, you can change most of these defaults as needed.

Applet. See *Java Applet*.

Array. A field or variable with multiple elements, each element having the same data type.

Bookmark. A saved link to a Web page that has been added to a list. Bookmarks are found on the Favorites menu in Internet Explorer.

Brightness. Defines the density of a color. Brightness ranges from zero for black, to 100 for white.

Browse. An inquiry program that displays records in a tabular format. Browse types in NetUI include look-up and drill-down browses. Drill-down browses can be specified on the menu or associated with a field. Look-up browses are always associated with individual fields for use in selecting entry values. See also *Drill-Down Browse* and *Look-Up Browse*.

Browser. An application used to view and navigate the World Wide Web and other Internet resources. Common browsers include Microsoft Internet Explorer and Netscape Navigator.

Bytecode. Compiled Java code containing instructions to the Java virtual machine. The virtual machine, in turn, interprets these instructions so they can be performed by the system. See also *Java Virtual Machine (JVM)*.

Cascading Style Sheet (CSS). A Web page derived from multiple sources with a defined order of precedence to manage conflicts among style element definitions. The cascading style sheet, level 1 (CSS1) recommendation from the World Wide Web Consortium (W3C) is implemented in the latest versions of the Netscape and Microsoft Web browsers.

CGI. See *Common Gateway Interface (CGI)*.

Character Interface. Sometimes abbreviated as CUI or CHUI, an interface to MFG/PRO eB that displays only keyboard characters on the screen rather than icons and buttons. In NetUI, telnet maintenance programs display in the character interface, while browses, reports, inquiries and XUI programs display in a full graphical user interface. Compare with *Graphical User Interface (GUI)* and *Encryption*.

Comma-Separated Values (CSV). Sometimes referred to as *flat files*, CSV files organize values as a series of ASCII text lines where each column value is separated by a comma

from the next column's value and each row starts a new line. Data in CSV files can be conveyed as input to other table-oriented applications such as Microsoft Excel.

Common Gateway Interface (CGI). A standard way for a Web server to pass a Web user's request to an application program and to receive data in response, which it then forwards to the user. CGI is part of the Web's hypertext transfer protocol (HTTP). See also *HTTP (Hypertext Transfer Protocol)*.

Constant. In NetUI security, a generic variable identifying a category of data affected by a rule. Constant values are used with rules to determine which database records users can access in Web-enabled browses, reports, and inquiries. See also *Rule*.

Cookie. Information a Web site puts on your hard disk so that it can retrieve information about you at a later time. Typically, a cookie records your preferences when using a particular site.

CSS. See *Cascading Style Sheet (CSS)*.

CSV. See *Comma-Separated Values (CSV)*.

Data Encryption. See *Embedded User Interface*.

DHTML (Dynamic Hypertext Markup Language). A combination of new HTML tags and options that support more animated and interactive Web pages. Much of dynamic HTML is specified in HTML 4.0 with Javascript and cascading style sheets. See also *HTML (Hypertext Markup Language)*.

Download Size. In NetUI browses, this is the number of records the browse window displays before the client notifies the server to retrieve more records.

Drill-Down Browse. A type of browse that is available from a menu or associated with individual fields for use in selecting entry values. You can filter, graph, and print from drill-down browses. In NetUI, drill-down browses apply to fields in other browses, inquiries, reports, and XUI programs. Programs displayed in a telnet session use look-up browses only. Compare with *Look-Up Browse*.

Embedded User Interface. In NetUI, an interface designed to allow users to access MFG/PRO eB programs from a Web page. You can access only one program at a time in the embedded interface. Compare with *Full User Interface*.

Emulation. See *Terminal Emulation*.

Encryption. Conversion of data into a form that cannot be easily intercepted by unauthorized people.

Extended User Interface (XUI). Describes the Java-based, graphical interface used with a subset of MFG/PRO eB maintenance and transaction programs in the NetUI. XUI programs feature navigation panels, tabs, grids, and application control buttons.

Extensible Markup Language (XML). A markup language that describes data. The NetUI uses XML to describe XUI screen presentations.

Extranet. A private network designed to securely share portions of business information or operations with suppliers, partners, customers, or other businesses. An extranet can be viewed as part of a company's intranet that is extended to users outside the company. See also *Internet Server Application Programming Interface (ISAPI)*.

Field Tips. Context-specific references to fields consisting of the Progress field name and its associated table name for browses, Progress field name only for reports and inquiries, or Java field name for XUI programs.

Full User Interface. In NetUI, an interface characterized by a hierarchical tree menu and ability to access multiple programs simultaneously. Compare with *Embedded User Interface*.

Georgia SoftWorks Windows NT Telnet Server (GSNTS). The third-party telnet server software supplied with the NetUI for Windows NT servers.

Graphical User Interface (GUI). A user interface that presents computer actions and options as pictures, buttons, and icons. The most common example of a graphical user interface is Microsoft Windows. Compare with *Character Interface*.

Graphics Interchange Format (GIF). A graphics file format that is the industry standard for Web and Internet use.

Grid. A screen design element that organizes related data fields in a format similar to a spreadsheet or table.

Group. See *Security Group*.

GUI. See *Graphical User Interface (GUI)*.

HTML (Hypertext Markup Language). A tag-based ASCII language used to create pages on the World Wide Web. HTML uses codes surrounding a block of text to indicate how it should display. In HTML, you can also specify that a block of text, or a word, is linked to another file on the Internet.

HTTP (Hypertext Transfer Protocol). The set of rules for exchanging text, graphic images, sound, video, and other multimedia files on the World Wide Web. See also *Common Gateway Interface (CGI)*.

Hue. Hue defines a color's position in a continuous color spectrum. Hue is measured in degrees and ranges from zero to 360, where 0 is red, 60 is yellow, 120 is green, 180 is cyan, 200 is magenta, and 240 is blue.

Interface. See *User Interface (UI)*.

International Organization for Standardization. See *ISO*.

Internet. A system of linked computer networks—international in scope—that facilitates data communication services such as remote log-in, file transfer, electronic mail, and newsgroups. The Internet is a way of connecting existing computer networks.

Internet Information Server (IIS). Web server software from Microsoft Corporation that supports a common gateway interface.

Internet Server Application Programming Interface (ISAPI). A program-level means of communicating with the Microsoft Internet Explorer Web server.

Intranet. A private network inside a company or organization that uses the same kinds of software found on the public Internet.

ISO. ISO (International Organization for Standardization), founded in 1946, is a worldwide federation of national standards bodies from some 100 countries. ISO is not an abbreviation. It is a word, derived from the Greek *isos*, meaning equal. The name ISO is used around the world to denote the organization.

ISS. See *Internet Information Server (IIS)*.

JAR File. See *Java Archive File (JAR)*.

Java. An object-oriented programming language created by Sun Microsystems. Java is a device-independent language. Programs compiled in Java can be run on any computer. Java programs can be run as free-standing applications or as applets placed on a Web page.

Java Applet. A small application program that can be sent along with a Web page to a user. Applets written in Java are served from a Web site but executed on the client computer.

Java Archive File (JAR). A file that contains the class, image, and sound components of a Java applet gathered into a single file and compressed for faster downloading to a Web browser.

Java Bean. A reusable program building block developed with a Beans Development Kit (BDK) from Sun Microsystems.

Java Class. In object-oriented programming, a class is a template definition of the methods and variables in a particular kind of object.

Java Development Kit (JDK). A software development environment from Sun Microsystems for writing applets and applications in the Java programming language.

Java Plug-In. Software provided by Sun Microsystems that replaces the default virtual machine associated with a Web browser. Using the Java plug-in allows developers to deploy Java applets that depend on the latest features of the Java platform and be assured that their applets will run reliably and consistently in both Microsoft Internet Explorer and Netscape Navigator.

Java Runtime Environment (JRE). A subset of the Java Development Kit for end users and developers who want to redistribute the Java runtime environment. The Java runtime environment consists of the Java virtual machine (JVM), the Java core classes, and supporting files.

Java Virtual Machine (JVM). The part of the Java runtime environment responsible for interpreting bytecode. See also *Bytecode*.

JDK. See *Java Development Kit (JDK)*.

JRE. See *Java Runtime Environment (JRE)*.

JVM. See *Java Virtual Machine (JVM)*.

Leaf Nodes. The lowest-level elements in a hierarchical navigation tree. Leaf nodes cannot be expanded or collapsed.

Look-Up Browse. A type of browse that is only associated with individual program fields for use in selecting entry values. Look-up browses contain less detail than drill-down browses and cannot be used to filter, graph, or print data. Compare with *Drill-Down Browse*.

Maintenance Programs. Programs used to add, modify, and delete records and codes in the MFG/PRO eB database. In NetUI, all maintenance programs can display in a character-based telnet window. Some maintenance programs can also run with the graphical, extended user interface (XUI). See also *Encryption*.

Menu Substitution. Replacing one program with another on the MFG/PRO eB menu; for example, replacing a standard program with a customized version.

MIME (Multipurpose Internet Mail Extension). A protocol for exchanging different kinds of information on the Internet. The MIME header is inserted at the beginning of a Web transmission so that client programs can select the appropriate associated application.

Netscape Server Application Programming Interface (NSAPI). A program-level means of communicating with the Netscape Web server.

Network. A series of points or nodes interconnected by communication paths. The Internet is a common example of a network. See also *Internet*, *Extranet*, *Internet Server Application Programming Interface (ISAPI)*, and *Wide Area Network (WAN)*.

Operator. A symbol or mnemonic used to perform calculations or data comparisons—for example, =, <, AND, OR, and XOR.

Perl. A script programming language (Practical Extraction and Reporting Language) that is similar in syntax to the C language and that includes a number of popular UNIX facilities. Perl is regarded as a good choice for developing Common Gateway Interface (CGI) programs because it has good text manipulation facilities.

Platform. An underlying computer system on which application programs run. Historically, most application programs had to be written to run on a particular platform. Products written in Java, however, are *cross-platform*, meaning they can be run anywhere the Java runtime environment is installed.

Point. A unit of about 1/72 inch used to measure type size.

Profile. In NetUI security, a set of MFG/PRO eB programs grouped together to simplify security setup. Profiles are assigned to security groups. For example, a group consisting of salespeople might be assigned a profile consisting of only sales programs. See also *Security Group*.

Relational Database Management System (RDBMS). The Progress software that manages and provides access to the MFG/PRO databases.

Remote Method Invocation (RMI). The Java programming language uses RMI for remote procedure call (RPC) functionality. RMI lets Java programs call procedures and retrieve files from remote machines. The NetUI uses an RMI server to manage the interface elements of the XUI and to retrieve NetUI online help.

RSA (Rivest-Shamir-Adleman). An Internet encryption and authentication system that uses an algorithm developed in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. The RSA algorithm is included with Netscape Navigator and Microsoft Explorer.

Rule. In NetUI security, rules define conditions under which specific database records can be accessed by browses, reports, and inquiries programs. Types of rules include field rules, logical rules, and dual rules. Rules are associated with individual programs as part of NetUI security setup.

Saturation. Saturation defines the intensity of a color and ranges from zero to 100.

Script. A program or sequence of instructions that is interpreted or carried out by another program.

Secure Sockets Layer (SSL). A program layer for managing the security of message transmissions in a network. The program layer exists between an application (such as a Web

browser or HTTP) and the Internet's TCP/IP layers. Sockets refers to the sockets method of passing data back and forth between a client and a server program in a network or between program layers in the same computer.

Security Group. A category of users who typically access the same programs in NetUI. Using NetUI security features, you can apply standard security settings to groups of users, then define exceptions for particular individuals within these groups as needed. See also *Profile*.

Security Profile. See *Profile*.

Socket. A convention for connecting with and exchanging data between two program processes within the same computer or across a network. A socket represents the end point in a network connection. Sockets are created and used with a set of programming requests or function calls sometimes referred to as the sockets application program interface (API). The most common sockets API is the Berkeley UNIX C language interface.

SSL. See *Secure Sockets Layer (SSL)*.

State-Aware. Describes the condition of a browse when a WebSpeed agent is in use during the entire time it is open. This can happen when the user sorts browse data by a non-indexed field, the WebSpeed agent returns a group of data that ends with a non-unique key, or the browse takes an unusually long time to begin returning data to the Web server.

TCP/IP. See *Transmission Control Protocol/Internet Protocol (TCP/IP)*.

Telnet. A user command and underlying TCP/IP protocol that lets you access applications and data on remote, or *host*, computers. See also *Transmission Control Protocol/Internet Protocol (TCP/IP)*.

Terminal Emulation. Use of a personal computer to interact with a computer with a different operating system. The terminal emulation program runs as a separate task with its own window. The application interface presented in this window is character-based or text-only.

Thin-Client. In a thin-client model, the client machine takes on only the user interface role. No business logic processing is accomplished on the client.

Tool Tips. Context-specific descriptions that display whenever your cursor is positioned over a tool button in NetUI.

Transmission Control Protocol/Internet Protocol (TCP/IP). The basic communication language or protocol of the Internet. It can also be used as a communications protocol for intranets and extranets.

UI. See *User Interface (UI)*.

Uniform Resource Locator (URL). A text string that indicates the location of an intranet or Internet resource.

Universal Unique Identifier (UUID). A hexadecimal number including a time stamp and a host identifier. Applications use uuids to identify many kinds of entities.

User Interface (UI). The portion of an application that is visible to the user and the mechanism by which the end user interacts with the application, enters information into the application, and sees the results of the interaction.

UUID. See *Universal Unique Identifier (UUID)*.

W3C. See *World Wide Web Consortium (W3C)*.

WAN. See *Wide Area Network (WAN)*.

Web Browser. See *Browser*.

Web-Enabled. Refers to a program that has code added to it so that it can be accessed from the NetUI. User interface actions in the Progress code are bypassed and rerouted to the Java client.

Web Site. A related collection of Web files that includes an introductory file called a home page. From the home page, you can get to all the other pages at that site.

WebSpeed. A product from Progress Software consisting of two parts: a set of Web-centric development tools and a transaction Web server. The server manages high-volume database transactions across multiple servers.

Wide Area Network (WAN). Generally a corporate private network that connects computers between remote company sites.

Widget. In Java screen-design terms, identifies a unique screen element. Widgets display information or provide specific ways for users to interact with application programs.

Wildcards. Characters or symbols used in search or command functions in place of one or more letters or numbers.

World Wide Web Consortium (W3C). An international industry consortium that seeks to promote standards for the evolution of the Web and interoperability among Internet products by producing specifications and reference software.

XML. See *Extensible Markup Language (XML)*.

XUI. See *Encryption*.

Index

Numerics

- 36.3.21.24 183
- 36.4.4 193
- 36.4.11 194
- 36.4.17.1 155
- 36.19.1 155
- 36.20.2 194
- 36.20.10 182
- 36.20.10.1 183
- 36.20.10.3 123, 141
- 36.20.10.8 188
- 36.20.10.11 189
- 36.20.10.15 192

A

- AdminServer
 - overview 28
 - starting 65
- agent sizing for WebSpeed 206
- Appearance Manager
 - RMI server 151
 - UIElements.xml 153
- application modes
 - embedded 199
 - full 199
- AppServer
 - auxiliary, defining constants 166
 - constants for, defining 156
 - functions 25
 - log files in Progress Explorer 78
 - log files in ubroker.properties 93
 - management utility commands 77
 - overview 30
 - setting up, using Progress Explorer 73
 - setting up, using ubroker.properties 89
 - verifying 77
- AppServer Service Maintenance 155
- architecture, of NetUI 17

B

- Browse URL Maintenance 189
- browses
 - saving settings 184
 - timing out 186

C

- client installation
 - after Service Pack 2 172
 - before Service Pack 2 168
- copying files
 - UNIX 102
 - Windows 115
 - JRE 1.3 172
- client_setup.html 176
- code pages, mapping Progress and Java 166
- constants 155
 - auxiliary AppServer constants 166
 - code pages 166
 - language codes 160
 - online help constants 159
 - primary AppServer constants 156
 - RMI server constants 158
 - service constants 156
- cookies
 - browser setting required 35
 - setup complete 171, 177

D

- date formats 164
- date separator 163
- document directory
 - creating 56
 - functions 26
- download size, user option 186

E

- eB Desktop
 - RMI server 150
 - use with NetUI 11
- embedded application mode 199
- environment tables 219
- errata sheet 3
- executables directory
 - creating 56
 - functions 26
- expiration date, user option 185
- extended user interface (XUI)
 - components 24
 - modifying files 151
 - overview 21
 - RMI server 153

F

- font.properties file 177
- fonts, non-English languages 177
- full application mode 199

G

- Georgia SoftWorks (GSNTS)
 - installing 133
 - registering 134
 - startup files 136
- Global Support Services 6

H

- help, online
 - constants for 159
 - creating script 154
- HTML files
 - default HTML 204
 - embedded HTML 202
 - example HTML 203
 - full HTML 201
 - general format 204
 - index HTML 204
 - installing 115
 - multinet HTML 202
 - optional parameters 206
 - required parameters 205
 - Service Pack 2, effect of 201
 - title HTML 204
 - trace HTML 202

I

- images directory 188, 194
- internationalization
 - code pages 166

- font.properties 177
- Java Plug-In 35, 172
- language codes 160
- ISO language and country codes 161

J

- Java Development Kit (JDK)
 - installing 57, 150
 - NetUI service pack 10
 - overview 13
- Java Plug-In
 - functions 19
 - overview 15
- Java Plug-In Console 202
- Java Plug-In Control Panel 171, 177, 216
- Java Runtime Environment (JRE), client install 172
- Java Virtual Machine (JVM) 15
- java.policy file 177
- JavaUI printer 216

L

- Label Master Maintenance 155
- language codes
 - accessing 162
 - date formats 164
 - date separator 163
 - elements 163
 - numeric formats 164
 - overview 160
- language, non-English
 - constants 164
 - font.properties 177
 - US data required for 156
- log files
 - AppServer 78, 93
 - WebSpeed 72, 89

M

- management utilities
 - AppServer commands 77
 - WebSpeed server commands 71
- Max Browse Records field 186
- Max Web Connections field 185
- menu
 - hiding disabled items 184
 - substitution 184
- Menu Password Maintenance 184
- Menu System Maintenance 193
- Menu URL Maintenance 188
- MFG/PRO application code 23
- MFG/PRO databases
 - client/server connection parameters 59

- connecting to 58
- functions 23
- local host connection parameters 59
- MIME types 56
- modes
 - embedded application 199
 - full application 199
- multinet directory
 - NT 117
 - UNIX 107
- Multi-Purpose Internet Mail Extension (MIME) 56

N

- NameServer
 - overview 28
 - port number 157, 167
 - starting 88
- NetUI interfaces 20
- NetUI Java code 16
 - functions 20
 - installing, on Web server 115
 - overview 16
- NetUI Menu 182
- NetUI Security Control File 183
- numeric formats 164

P

- parameter file
 - client/server connection parameters 59
 - creating 60
 - local host connection parameters 59
- planning NetUI installation 42
- Printer Setup Maintenance 216
- printers, remote 184
- Progress character client, functions 23
- Progress Explorer
 - overview 29
 - setting up Progress servers 65
 - setting up remote machines 66
 - starting 65
- Progress relational database management system (RDBMS) 23
- Progress servers
 - administering 28
 - management utilities 29
 - overview 27
 - setting up 63
- proxy server 198

Q

- QAD Global Support Services 6
- QAD signature file

- functions 19
- overview 15
- qadkeystore 177

R

- remote method invocation (RMI) server
 - configuration script 151
 - constants for 158
 - functions 24
 - log files 152
 - setting up 150
 - starting 152
 - verifying 152
- remote printers 184
- requirements
 - client 35
 - for installing NetUI 13
 - Java 38
 - Progress server 37
 - RMI server 38
 - Web server 36
- Restrict Menu field 184
- restricted shells 121
- rmi installation directory, creating 151

S

- saving browse settings 184
- screen definitions
 - public 153
 - QAD standard 153
- secure sockets layer (SSL) security
 - limitations 209
 - overview 209
 - setting up 211
 - verifying 214
- security
 - restricted shells 121
 - telnet, UNIX 120
- security files, NetUI 15
- service pack
 - effect on JDK version 10
 - installing
 - NT 117
 - UNIX 107
- Session Master Maintenance 192
- session timeout 186
- signature file 15
- swingall.jar 171

T

- telnet interface 22
- telnet server

- functions 23
- installing on Windows NT 133
- overview 14
- registering on Windows NT 134
- startup file types on Windows NT 136
- UNIX connection settings, configuring 129
- UNIX log-in sequence, configuring 125
- UNIX security, configuring 120
- UNIX server settings, configuring 124
- verifying on UNIX 129
- verifying on Windows NT 146
- Windows NT connection settings, configuring 146
- Windows NT installation prerequisites 132
- Windows NT log-in sequence, configuring 143
- Windows NT server settings, configuring 141
- terms. *See* constants
- timeout
 - browse 186
 - minutes 187
 - session 186

U

- UI Configurator 151
- UIElements.xml 154
- Unified Broker Properties File (ubroker.properties)
 - modifying 83
 - overview 28
- uniform resource locators (URLs)
 - browse links 189
 - menu-level links 188
- UNIX telnet security 120
- User Function Maintenance 194

- User Option Maintenance 183
- User Option Telnet Maintenance 123, 141
- user options
 - expiration date 185
 - menu substitution 184
 - program execution 184
 - restrict menu 184
- User Tool Maintenance 194

W

- Web browser
 - functions 19
 - overview 14
- Web server
 - functions 26
 - installing 56
 - overview 14
 - setting up MIME-types on UNIX 56
 - verifying 57
- WebSpeed server
 - agent sizing 206
 - functions 26
 - log files in Progress Explorer 72
 - log files in ubroker.properties 89
 - management utility commands 71
 - overview 29
 - setting up, using Progress Explorer 68
 - setting up, using ubroker.properties 83
 - verifying 71

X

- XUI. *See* extended user interface