

Progress
Installation and Configuration Guide
Version 9
for Windows

© 2002 Progress Software Corporation. All rights reserved.

Progress® software products are copyrighted and all rights are reserved by Progress Software Corporation. This manual is also copyrighted and all rights are reserved. This manual may not, in whole or in part, be copied, photocopied, translated, or reduced to any electronic medium or machine-readable form without prior consent, in writing, from Progress Software Corporation.

The information in this manual is subject to change without notice, and Progress Software Corporation assumes no responsibility for any errors that may appear in this document.

The references in this manual to specific platforms supported are subject to change.

Progress, Powered by Progress, Progress Fast Track, Progress Profiles, Partners in Progress, Partners en Progress, Progress en Partners, Progress in Progress, P.I.P., Progress Results, ProVision, ProCare, ProtoSpeed, SmartBeans, SpeedScript, and WebSpeed are registered trademarks of Progress Software Corporation in the U.S. and other countries. A Data Center of Your Very Own, Allegrix, Apptivity, AppsAlive, AppServer, ASPen, ASP-in-a-Box, Empowerment Center, Fathom, Future Proof, IntelliStream, OpenEdge, PeerDirect, POSSE, POSSENET, Progress Dynamics, Progress Software Developers Network, SectorAlliance, SmartObjects and WebClient are trademarks or service marks of Progress Software Corporation in the U.S. and other countries.

SonicMQ is a registered trademark of Sonic Software Corporation in the U.S. and other countries.

Vermont Views is a registered trademark of Vermont Creative Software in the U.S. and other countries.

Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Any other trademarks and/or service marks contained herein are the property of their respective owners.

Progress Software Corporation acknowledges the use of Raster Imaging Technology copyrighted by Snowbound Software 1993-1997, the IBM XML Parser for Java Edition, and software developed by the Apache Software Foundation (http://www.apache.org/).

© IBM Corporation 1998-1999. All rights reserved. U.S. Government Users Restricted Rights — Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Progress is a registered trademark of Progress Software Corporation and is used by IBM Corporation in the mark Progress/400 under license. Progress/400 AND 400® are trademarks of IBM Corporation and are used by Progress Software Corporation under license.

The implementation of the MD5 Message-Digest Algorithm used by Progress Software Corporation in certain of its products is derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm.

May 2002



Product Code: 4549
Item Number: 89428W;V91D

Contents

Preface . . . . . **xv**
Purpose xv
Audience xv
Organization Of This Manual xv
Typographical Conventions xvii
Syntax Notation xviii
Progress Messages xxii
Other Useful Documentation xxiii
Getting Started xxiii
Development Tools xxv
Reporting Tools xxvi
4GL xxvii
Database xxviii
DataServers xxviii
SQL-89/Open Access xxix
SQL-92 xxix
Deployment xxx
WebSpeed xxx
Reference xxxi

PART I Installation

1. Installation Requirements **1–1**
1.1 System Requirements 1–2
1.1.1 Java 1–2
1.1.2 Windows NT On Intel 1–4
1.1.3 Windows 2000 1–6
1.1.4 Windows 98 1–7

1.1.5 Windows Millennium Edition (Me) 1–9
1.1.6 Windows XP 1–10
1.1.7 Citrix MetaFrame 1–11
1.1.8 Disk Space 1–12
1.2 4GL Client/Database Server Compatibility 1–15
1.3 Product Life Cycle 1–15

2. Installing Progress 2–1
2.1 Installing Visual Translator 2–3
2.2 Installing WebClient 2–3
2.3 Performing Preinstallation Tasks 2–3
2.3.1 Shutting Down Nonessential Applications Or Tasks 2–3
2.3.2 Resetting Read-only Files 2–4
2.3.3 Installing Other Required Applications 2–4
2.3.4 Saving An Existing Progress Installation 2–5
2.4 Beginning the Installation 2–7
2.4.1 Other Considerations Before You Begin the Progress Installation 2–8
2.5 Installing Required Applications During Progress Installation 2–9
2.5.1 Installing the Microsoft Management Console 2–10
2.5.2 Installing Microsoft HTML Help 2–10
2.5.3 Installing Microsoft Java 2–11
2.5.4 Installing the MERANT SQL-92 ODBC Drivers and the MERANT ODBC Branded Drivers 2–11
2.6 Installing Progress On Your System 2–14
2.7 Performing a Typical Or Complete Installation 2–20
2.8 Performing a Custom Installation 2–28
2.9 Finishing the Installation 2–34
2.10 InstallShield Silent (Batch Mode) Utility 2–35
2.10.1 Running the InstallShield Silent Utility 2–35
2.10.2 Sample setup.iss File 2–37
2.11 Installing Additional Products 2–39
2.11.1 Changes In the Progress Core Installation Program For Progress Messages Files 2–40
2.12 Installing Additional Components To Previously Installed Products 2–46
2.13 Viewing Registry Information 2–48
2.14 Performing Postinstallation Tasks 2–49
2.15 Downloading Executables For Heterogeneous Environments 2–50
2.16 Uninstalling Progress 2–51
2.16.1 Using the Uninstall Or Add/Remove Programs Utility 2–52
2.16.2 Manually Removing Progress Version 9.1D 2–54

3. Sharing a Progress Installation On a Network 3–1
3.1 Networking Overview 3–2
3.1.1 Application Workstation 3–2
3.1.2 Database Server Machine 3–2
3.1.3 Network File Server Machine 3–2
3.2 Preinstallation Considerations 3–3
3.3 Using the Shared Network Installation Utility 3–3
3.4 Uninstalling the Shared Network Installation Utility 3–4
3.5 New Silent Installation Option For the Shared Network Installation Utility 3–5
3.5.1 Creating a User-Defined Response File 3–5
3.5.2 Executing NetSetup With the Silent Installation Option 3–7

4. Administration Utilities 4–1
4.1 Using the Product Update Utility 4–2
4.2 Displaying License Information Using the SHOWCFG Utility 4–3
4.2.1 Using the SHOWCFG Utility 4–3
4.2.2 Displaying License Information 4–4
4.3 Managing User Licenses 4–5
4.3.1 Progress License Information 4–5
4.3.2 Using the Progress License File 4–6
4.4 Using the Progress Explorer Tool 4–8
4.5 Using the ProControl Utility 4–10
4.6 Using Progress Resources 4–11
4.6.1 Shared Memory 4–11
4.6.2 Processes 4–12

PART II Configuration

5. Setting Up Your Environment 5–1
5.1 Setting System Environment Variables 5–2
5.2 Setting Java Environment Variables 5–2
5.2.1 JDKHOME 5–2
5.3 Setting Progress Program Item Properties 5–3
5.4 The Windows Registry and the progress.ini File 5–3
5.4.1 Environment Variables 5–3
5.5 The Progress Working Directory 5–9
5.6 Using the Proenv Utility 5–9
5.7 Managing the AdminServer 5–9
5.7.1 Starting the AdminServer 5–10
5.7.2 Stopping the AdminServer 5–10
5.7.3 Changing the Default Port 5–11

5.7.4 Changing the Default Adminport 5–11
5.7.5 Changing the Startup Setting 5–12
5.7.6 Running Multiple AdminServers 5–12
5.7.7 Querying the AdminServer 5–12
5.8 Creating a Version 9 Database 5–13
5.9 Creating and Configuring a Version 9 Database Server 5–13
5.10 Running Progress 5–14
5.11 Maintaining Multiple Versions Of Progress 5–14

6. Configuring Progress 6–1
6.1 The Progress Explorer Framework 6–2
6.1.1 AdminServer 6–2
6.1.2 Progress Explorer 6–2
6.1.3 Command-line Configuration Utilities 6–3
6.1.4 Managing Progress Product Configurations 6–4
6.2 Shared-memory Configurations 6–8
6.3 Multi-tier Configurations 6–10
6.3.1 Shared-memory Architecture 6–11
6.4 Client/Server Configurations 6–13
6.4.1 Terminology 6–13
6.4.2 Simple Client/Server Configurations 6–15
6.5 Client/Server and Progress AppServer In the Network Environment . . . 6–16
6.6 Preparing To Run Progress In a TCP/IP Network 6–23
6.6.1 Installing Progress On Your TCP/IP Network 6–23
6.6.2 Typical TCP/IP Configuration With a Hard Disk On Each Machine 6–24
6.6.3 Setting Up Network Files To Run Progress 6–25
6.6.4 Configuring Progress On a Network Operating System 6–25

7. Configuring Progress Unified Broker Products 7–1
7.1 The Progress Unified Broker Administration Framework 7–2
7.1.1 Unified Broker Clients 7–4
7.1.2 Progress Explorer and Management Utilities 7–5
7.1.3 NameServers 7–7
7.1.4 Unified Broker 7–8
7.1.5 Unified Broker Properties File (ubroker.properties). 7–9
7.1.6 Using a Text Editor and Configuration Utilities 7–11
7.1.7 Overview Of Unified Broker Administration. 7–12
7.1.8 Distributing Machines For a Unified Broker Product 7–16
7.1.9 Configuring NameServer Communications. 7–17
7.2 Configuring Progress Unified Broker Components 7–18
7.2.1 Downloading NameServer Executables 7–19
7.2.2 Order Of Configuration 7–19
7.2.3 Configuring and Using NameServer Instances. 7–19

7.2.4 Configuration Using the Progress Explorer Tool 7–20
7.2.5 Configuring the NameServer In the Progress Explorer 7–21
7.2.6 Editing the Properties File 7–22
7.3 Setting Up the Progress Unified Broker Environment 7–24
7.3.1 Setting Environment Variables. 7–24
7.3.2 Configuring Progress Unified Broker Log Files 7–25
7.3.3 Other Preparations Prior To Startup 7–26
7.4 Starting Progress Unified Brokers 7–26
7.4.1 Using the Progress Explorer To Start a Unified Broker Instance 7–26
7.4.2 Starting and Managing a NameServer Using the Progress Explorer 7–27
7.4.3 Using the Management Utilities To Start a Unified Broker Instance 7–28
7.4.4 Starting and Managing a NameServer Using the NSMAN Utility 7–29
7.5 Understanding Load Balancing 7–31
7.6 Understanding Server-level and Connection-level Fault Tolerance 7–33
7.6.1 Connection-level Fault Tolerance 7–34
7.6.2 Using UDP Broadcasting 7–35
7.6.3 Using NameServer Replication 7–35
7.6.4 Using NameServer Neighbors 7–38
7.6.5 Performance Implications Of Broadcasting 7–41

8. Starting and Running Progress 8–1
8.1 Starting Progress 8–2
8.1.1 Startup and Shutdown 8–2
8.1.2 Starting Progress As a Windows NT Service 8–6
8.1.3 Starting Single-user Progress In Interactive Mode 8–8
8.1.4 Starting Single-user Progress In Batch Or Background Mode 8–8
8.1.5 Starting the Multi-user Server Or Broker 8–9
8.1.6 Starting the Multi-user Server Or Broker As a Windows NT Service 8–10
8.2 Running Progress Clients and Servers On a Network 8–11
8.2.1 Using Network Startup Parameters 8–11
8.2.2 Specifying the Network Type (-N) 8–13
8.2.3 Network Addressing With Progress (-S and -H). 8–13
8.2.4 Starting Applications On a Network 8–13
8.2.5 Starting Multiple Brokers Using the Same Protocol 8–14

8.2.6 Accessing a Server Behind a Firewall. 8–16
8.2.7 Starting and Running Multi-user Progress In Interactive Mode 8–17
8.2.8 Starting and Running Multi-user Progress Clients In Batch Or Background Mode 8–17

9. Progress Event Logging 9–1
9.1 The Progress Event Log File 9–2
9.2 Managing the Progress Event Log File Size 9–2
9.3 Event Logging On Windows NT 9–2
9.3.1 Managing Progress Events On Windows NT 9–3
9.3.2 Understanding the Windows NT Application Event Log Components 9–4
9.3.3 Windows NT Event Log and Registry 9–6

A. The Progress Program Group A–1
A.1 Changing Program Item Properties A–2

B. The ProControl Utility B–1
B.1 Starting ProService B–2
B.2 Starting Single-user Progress In Batch Or Background Mode B–4
B.3 Starting the Multi-user Server Or Broker B–6
B.4 Starting Windows NT Multi-user Progress In Batch Or Background Mode B–8

C. The Progress Directory Structure C–1
C.1 Installation Path Directory Structure C–2

D. Progress National Language Support D–1
D.1 Packaging D–2
D.2 Directory Structure D–3
D.3 Contents Of Each Directory D–3
D.4 Implementing Regional Support D–4
D.5 International Databases D–4
D.6 Progress Messages D–5
D.7 Environment Variables Of the SQL-92 Client D–5
D.8 Regional Parameter Files D–7
D.9 The progress.ini File and the Windows Registry D–8

E. Integrating Progress With Windows Explorer E–1
E.1 Progress File Types E–2
E.2 Icons E–3
E.3 Shortcut Menus E–3
E.4 Properties E–3

F. Progress Version 9 Typical and Complete Installation Components F–1
F.1 Progress Version 9 Installation Options F–2
F.2 Progress Product Components and Subcomponents F–2
F.2.1 4GL Development F–3
F.2.2 AppServer Internet Adapter F–7
F.2.3 AppServer Internet Adapter/S F–8
F.2.4 Client Networking F–8
F.2.5 Client-side Security F–11
F.2.6 Debugger F–12
F.2.7 Enterprise Database F–13
F.2.8 ESQL-89/C F–16
F.2.9 NameServer F–16
F.2.10 NameServer Load Balancing Option F–17
F.2.11 ODBC Enterprise DataServer F–18
F.2.12 ODBC Personal DataServer F–21
F.2.13 Open Client Toolkit F–23
F.2.14 Oracle DataServer F–24
F.2.15 Personal Database F–27
F.2.16 Progress AppServer F–30
F.2.17 ProVision F–32
F.2.18 ProVision Development Server F–37
F.2.19 ProVision Plus F–41
F.2.20 ProVision Plus Development Server F–48
F.2.21 Query/Results F–52
F.2.22 Report Builder F–54
F.2.23 Secure AppServer F–54
F.2.24 SonicMQ Adapter F–57
F.2.25 SQL Server7 Enterprise F–58
F.2.26 SQL Server7 Personal F–61
F.2.27 SQL-92 Client Access F–63
F.2.28 Toolkit F–65
F.2.29 Translation Manager F–65
F.2.30 Visual Translator F–66
F.2.31 WebClient F–67
F.2.32 WebSpeed Development Server F–68
F.2.33 WebSpeed Enterprise Transaction Server F–71
F.2.34 WebSpeed Messenger F–74
F.2.35 WebSpeed Transaction Server F–75
F.2.36 WebSpeed Workshop F–78
F.2.37 Workgroup Database F–82

Index Index–1

Figures

Figure 2–1: Question dialog box 2–26
Figure 2–2: Warning dialog box 2–26
Figure 3–1: Contents Of the Sample Response File - Setup.iss 3–6
Figure 3–2: Contents of the Sample Log File - PscNetSetupMsg.log 3–8
Figure 4–1: Product Configuration Information From SHOWCFG Utility 4–3
Figure 6–1: Database Access On a Shared-memory System 6–9
Figure 6–2: Sample Multi-tier Configuration 6–10
Figure 6–3: Shared-memory Progress Architecture 6–11
Figure 6–4: Simple Client/Server Configuration 6–15
Figure 6–5: Multiple System Client/Server Configuration 6–16
Figure 6–6: Simple Progress Network Configuration 6–18
Figure 6–7: Network File Server For Application Files 6–19
Figure 6–8: Network File Server As a Database Server 6–20
Figure 6–9: Network File Server For Application and Database Files 6–21
Figure 6–10: LAN Configuration With the Progress AppServer 6–22
Figure 6–11: Typical TCP/IP Configuration (File Server Not Used) 6–24
Figure 7–1: Progress Unified Broker Administration Framework 7–2
Figure 7–2: Progress Unified Broker Administration Overview 7–13
Figure 7–3: Maximum Unified Broker Machine Distribution 7–16
Figure 7–4: Sample Unified Broker Client Services File 7–18
Figure 7–5: Server-level and Connection-level Fault Tolerance 7–33
Figure 7–6: NameServer Replication 7–36
Figure 7–7: NameServer Neighbors 7–40
Figure 9–1: Windows NT Application Event Log Components 9–4
Figure 9–2: Windows NT Application Event Detail Dialog Box 9–6

Tables

Table 1–1: JRE/JDK Requirements By Platform 1–3
Table 1–2: System Requirements For Windows NT On Intel 1–4
Table 1–3: System Requirements For Windows 2000 1–6
Table 1–4: System Requirements For Windows 98 1–7
Table 1–5: System Requirements For Windows Millennium 1–9
Table 1–6: System Requirements For Windows XP 1–10
Table 1–7: System Requirements For Citrix MetaFrame 1–11
Table 1–8: Approximate Disk Space 1–12
Table 1–9: Client/Server Compatibility 1–15
Table 1–10: Product Life Cycle Phases 1–16
Table 2–1: Version 8 Templates 2–5
Table 2–2: Installation Options 2–8
Table 2–3: Driver Files For Windows Operating Systems For SQL-92 2–12
Table 2–4: Driver Files For Windows Operating Systems With Enterprise DataServer For ODBC 2–13
Table 2–5: Data-source Components and Version Numbers 2–14
Table 2–6: Illegal Characters 2–25
Table 2–7: Promsgs File Synchronization Process 2–43
Table 2–8: Example Of Promsgs Files Being Out Of Sync 2–45
Table 5–1: Supported Environment Variables 5–4
Table 6–1: ubroker.properties File Validation Utilities 6–7
Table 6–2: ubroker.properties File Management Utilities 6–8
Table 6–3: Network Type Options For Progress 6–17
Table 6–4: TCP/IP Network Files 6–25
Table 7–1: Unified Broker Administration Summary 7–3
Table 7–2: Weight Factors Based On Percentage 7–32
Table 7–3: Weight Factors Based On Arbitrary Sums 7–32
Table 8–1: Windows GUI Startup and Shutdown Commands 8–3
Table 8–2: Windows Startup and Shutdown Commands 8–4
Table 8–3: Client Network Parameters 8–11
Table 8–4: Server Network Parameters 8–12
Table 8–5: Default Network Types 8–13
Table 9–1: Progress Event Logging Components 9–3
Table 9–2: Event Level Values 9–3
Table 9–3: Windows NT Application Event Log Components 9–5
Table A–1: Progress Program Group Icons A–3
Table C–1: Progress (%DLC%) Directory Structure C–2
Table D–1: PROMSGS Translations Shipped With Progress D–2
Table D–2: National Language File Descriptions D–3
Table D–3: Startup Parameters For a Deployed Application D–7
Table D–4: Environment Variables D–9
Table F–1: 4GL Development Components and Subcomponents F–3
Table F–2: AppServer Internet Adapter Components and Subcomponents F–7

Table F–3: AppServer Internet Adapter/S Components and Subcomponents F–8
Table F–4: Client Networking Components and Subcomponents F–8
Table F–5: Client-side Security Components and Subcomponents F–11
Table F–6: Debugger Components F–12
Table F–7: Enterprise Database Components and Subcomponents F–13
Table F–8: ESQL-89/C Components F–16
Table F–9: NameServer Components and Subcomponents F–16
Table F–10: AppServer Load Balancing Option Components and Subcomponents F–17
Table F–11: ODBC Enterprise DataServer Components and Subcomponents F–18
Table F–12: ODBC Personal DataServer Components and Subcomponents F–21
Table F–13: Open Client Toolkit Components F–23
Table F–14: Oracle DataServer Components and Subcomponents F–24
Table F–15: Personal Database Components and Subcomponents F–27
Table F–16: Progress AppServer Components and Subcomponents F–30
Table F–17: ProVision Components and Subcomponents F–32
Table F–18: ProVision Development Server Components and Subcomponents . . . F–37
Table F–19: ProVision Plus Components and Subcomponents F–41
Table F–20: ProVision Plus Development Server Components and Subcomponents F–48
Table F–21: Query/Results Components and Subcomponents F–52
Table F–22: Report Builder Components and Subcomponents F–54
Table F–23: Secure AppServer Components and Subcomponents F–54
Table F–24: SonicMQ Adapter Components and Subcomponents F–57
Table F–25: SQL Server7 Enterprise Components and Subcomponents F–58
Table F–26: SQL Server7 Personal Components and Subcomponents F–61
Table F–27: SQL-92 Client Access Components and Subcomponents F–63
Table F–28: Toolkit Components F–65
Table F–29: Translation Manager Components and Subcomponents F–65
Table F–30: Visual Translator Components and Subcomponents F–66
Table F–31: WebClient Component F–67
Table F–32: WebSpeed Development Server Components and Subcomponents . . F–68
Table F–33: WebSpeed Enterprise Transaction Server Components and Subcomponents F–71
Table F–34: WebSpeed Messenger Components and Subcomponents F–74
Table F–35: WebSpeed Transaction Server Components and Subcomponents . . . F–75
Table F–36: WebSpeed Workshop Components and Subcomponents F–78
Table F–37: Workgroup Database Components and Subcomponents F–83

Examples

setup.iss 2–37

Preface

Purpose

This book describes how to install and configure Progress Version 9 for Windows NT, Windows 2000, Windows 98, Windows Millennium (Me), Windows XP, Citrix MetaFrame platforms 1.0, 1.8, and XP. In addition, it addresses related issues such as the Progress Program Group and directory structure, language support, and sample applications.

Audience

This book is intended for the person installing Progress. You are not required to have prior knowledge of Progress, but previous experience with the Windows NT, Windows 2000, Windows 98, Windows XP, Citrix MetaFrame or Citrix MetaFrame XP operating system is assumed. Should you require information about any of these operating systems, refer to your Windows or Citrix MetaFrame documentation.

Organization Of This Manual

Part I, Installation

Chapter 1, “Installation Requirements”

Lists system requirements for installing Progress.

Chapter 2, “Installing Progress”

Provides step-by-step instructions for installing Progress using a Typical, Complete, or Custom installation.

Chapter 3, “Sharing a Progress Installation On a Network”

Describes how to share Progress over a network using the Shared Network Installation utility.

Chapter 4, “Administration Utilities”

Provides information about the Product Update utility, the SHOWCFG utility, the Progress Explorer tool, the ProControl utility, managing user licenses, and using Progress resources.

Part II, Configuration

Chapter 5, “Setting Up Your Environment”

Discusses setting Version 9 system and Java environment variables, managing the AdminServer, and running Progress Version 9.

Chapter 6, “Configuring Progress”

Provides information about the Progress Explorer framework, shared-memory configurations, multi-tier configurations, client/server configurations, client/server and Progress AppServer in the network environment, and preparing to run Progress in a TCP/IP environment.

Chapter 7, “Configuring Progress Unified Broker Products”

Describes the common framework and tasks to configure the common components of Progress Unified Broker products. These products all support multi-tier or client/server configurations using the Progress NameServer.

Chapter 8, “Starting and Running Progress”

Provides instructions for starting and connecting to a Progress database in different modes. Also provides information about running Progress clients and servers on a network.

Chapter 9, “Progress Event Logging”

Provides information about the Progress event log file, including managing the event log file size and event logging on Windows NT.

Appendix A, “The Progress Program Group”

Lists the items in the Progress Program Group, the commands to start them, and instructions on editing item properties.

Appendix B, “The ProControl Utility”

Provides information on using the ProControl utility to start up and connect to a Progress database, including how to start ProService, starting a single-user or multi-user server or broker, and starting Windows NT multi-user Progress in batch or background mode.

Appendix C, “The Progress Directory Structure”

Lists the subdirectories of your installation path.

Appendix D, “Progress National Language Support”

Gives information about Progress messages in different languages.

Appendix E, “Integrating Progress With Windows Explorer”

Contains information about integrating Progress with Windows Explorer.

Appendix F, “Progress Version 9 Typical and Complete Installation Components”

Provides a list of the components and subcomponents that you install for each product when you choose a Typical or Complete installation.

Typographical Conventions

This manual uses the following typographical conventions:

- **Bold typeface** indicates:
  - Commands or characters that the user types
  - That a word carries particular weight or emphasis
- *Italic typeface* indicates:
  - Progress variable information that the user supplies
  - New terms
  - Titles of complete publications

- `Monospaced typeface` indicates:
  - Code examples
  - System output
  - Operating system filenames and pathnames

The following typographical conventions are used to represent keystrokes:

- Small capitals are used for Progress key functions and generic keyboard keys.

  END-ERROR, GET, GO
  ALT, CTRL, SPACEBAR, TAB

- When you have to press a combination of keys, they are joined by a dash. You press and hold down the first key, then press the second key.

  CTRL-X

- When you have to press and release one key, then press another key, the key names are separated with a space.

  ESCAPE H
  ESCAPE CURSOR-LEFT

Syntax Notation

The syntax for each component follows a set of conventions:

- Uppercase words are keywords. Although they are always shown in uppercase, you can use either uppercase or lowercase when using them in a procedure.

  In this example, ACCUM is a keyword:

  SYNTAX

  ```
  ACCUM aggregate expression
  ```

- Italics identify options or arguments that you must supply. These options can be defined as part of the syntax or in a separate syntax identified by the name in italics. In the ACCUM function above, the *aggregate* and *expression* options are defined with the syntax for the ACCUM function in the *Progress Language Reference*.

- You must end all statements (except for DO, FOR, FUNCTION, PROCEDURE, and REPEAT) with a period. DO, FOR, FUNCTION, PROCEDURE, and REPEAT statements can end with either a period or a colon, as in this example:

```
FOR EACH Customer:
   DISPLAY Name.
END.
```

- Square brackets (**[ ]**) around an item indicate that the item, or a choice of one of the enclosed items, is optional.

  In this example, STREAM *stream*, UNLESS-HIDDEN, and NO-ERROR are optional:

  SYNTAX

```
DISPLAY [ STREAM stream ] [ UNLESS-HIDDEN ] [ NO-ERROR ]
```

  In some instances, square brackets are not a syntax notation, but part of the language.

  For example, this syntax for the INITIAL option uses brackets to bound an initial value list for an array variable definition. In these cases, normal text brackets ([ ]) are used:

  SYNTAX

```
INITIAL [ constant [ , constant ] ... ]
```

  NOTE: The ellipsis (**...**) indicates repetition, as shown in a following description.

- Braces (**{ }**) around an item indicate that the item, or a choice of one of the enclosed items, is required.

  In this example, you must specify the items BY and *expression* and can optionally specify the item DESCENDING, in that order:

  SYNTAX

```
{ BY expression [ DESCENDING ] }
```

  In some cases, braces are not a syntax notation, but part of the language.

For example, a called external procedure must use braces when referencing arguments passed by a calling procedure. In these cases, normal text braces ({ }) are used:

SYNTAX

```
{ &argument-name }
```

- A vertical bar (**|**) indicates a choice.

  In this example, EACH, FIRST, and LAST are optional, but you can only choose one:

  SYNTAX

  ```
  PRESELECT [ EACH | FIRST | LAST ] record-phrase
  ```

  In this example, you must select one of *logical-name* or *alias*:

  SYNTAX

  ```
  CONNECTED ( { logical-name | alias } )
  ```

- Ellipses (**. . .**) indicate that you can choose one or more of the preceding items. If a group of items is enclosed in braces and followed by ellipses, you must choose one or more of those items. If a group of items is enclosed in brackets and followed by ellipses, you can optionally choose one or more of those items.

  In this example, you must include two expressions, but you can optionally include more. Note that each subsequent expression must be preceded by a comma:

  SYNTAX

  ```
  MAXIMUM ( expression , expression [ , expression ] ... )
  ```

  In this example, you must specify MESSAGE, then at least one of *expression* or SKIP, but any additional number of *expression* or SKIP is allowed:

  SYNTAX

  ```
  MESSAGE { expression | SKIP [ (n) ] } ...
  ```

In this example, you must specify {*include-file*, then optionally any number of *argument* or &*argument-name* = "*argument-value*", and then terminate with }:

SYNTAX

```
{ include-file
    [ argument | &argument-name = "argument-value" ] ... }
```

- In some examples, the syntax is too long to place in one horizontal row. In such cases, **optional** items appear individually bracketed in multiple rows in order, left-to-right and top-to-bottom. This order generally applies, unless otherwise specified. **Required** items also appear on multiple rows in the required order, left-to-right and top-to-bottom. In cases where grouping and order might otherwise be ambiguous, braced (required) or bracketed (optional) groups clarify the groupings.

  In this example, WITH is followed by several optional items:

  SYNTAX

```
WITH [ ACCUM max-length ] [ expression DOWN ]
  [ CENTERED ] [ n COLUMNS ] [ SIDE-LABELS ]
  [ STREAM-IO ]
```

  In this example, ASSIGN requires one of two choices: either one or more of *field*, or one of *record*. Other options available with either *field* or *record* are grouped with braces and brackets. The open and close braces indicate the required order of options:

  SYNTAX

```
ASSIGN {   { [ FRAME frame ]
               { field [ = expression ] }
               [ WHEN expression ]
           } ...
         | { record [ EXCEPT field ... ] }
       }
```

Progress Messages

Progress displays several types of messages to inform you of routine and unusual occurrences:

- Execution messages inform you of errors encountered while Progress is running a procedure (for example, if Progress cannot find a record with a specified index field value).
- Compile messages inform you of errors found while Progress is reading and analyzing a procedure prior to running it (for example, if a procedure references a table name that is not defined in the database).
- Startup messages inform you of unusual conditions detected while Progress is getting ready to execute (for example, if you entered an invalid startup parameter).

After displaying a message, Progress proceeds in one of several ways:

- Continues execution, subject to the error-processing actions that you specify, or that are assumed, as part of the procedure. This is the most common action taken following execution messages.
- Returns to the Progress Procedure Editor so that you can correct an error in a procedure. This is the usual action taken following compiler messages.
- Halts processing of a procedure and returns immediately to the Procedure Editor. This does not happen often.
- Terminates the current session.

Progress messages end with a message number in parentheses. In this example, the message number is 200:

```
** Unknown table name table. (200)
```

Use Progress online help to get more information about Progress messages. Many Progress tools include the following Help menu options to provide information about messages:

- Choose Help→Recent Messages to display detailed descriptions of the most recent Progress message and all other messages returned in the current session.
- Choose Help→Messages, then enter the message number to display a description of any Progress message. (If you encounter an error that terminates Progress, make a note of the message number before restarting.)
- In the Procedure Editor, press the **HELP** key (**F2** or **CTRL-W**).

Other Useful Documentation

This section lists Progress Software Corporation documentation that you might find useful. Unless otherwise specified, these manuals support both Windows and Character platforms and are provided in electronic documentation format on CD-ROM.

Getting Started

Progress Electronic Documentation Installation and Configuration Guide (Hard copy only)

A booklet that describes how to install the Progress EDOC viewer and collection on UNIX and Windows.

Progress Installation and Configuration Guide Version 9 for UNIX

A manual that describes how to install and set up Progress Version 9.1 for the UNIX operating system.

Progress Version 9 Product Update Bulletin

A bulletin that provides a list of new and changed features by release, a list and description of changes to documentation by release, and critical information about product changes that might require changes to existing code and configurations.

This bulletin also provides information about where to go for detailed information about the new and changed features and documentation.

Progress Application Development Environment — Getting Started (Windows only)

A practical guide to graphical application development within the Progress Application Development Environment (ADE). This guide includes an overview of the ADE and its tools, an overview of Progress SmartObject technology, and tutorials and exercises that help you better understand SmartObject technology and how to use the ADE to develop applications.

Progress Language Tutorial for Windows and *Progress Language Tutorial for Character*

Platform-specific tutorials designed for new Progress users. The tutorials use a step-by-step approach to explore the Progress application development environment using the 4GL.

Progress Master Glossary for Windows and *Progress Master Glossary for Character* (EDOC only)

Platform-specific master glossaries for the Progress documentation set. These books are in electronic format only.

Progress Master Index and Glossary for Windows and *Progress Master Index and Glossary for Character* (Hard copy only)

Platform-specific master indexes and glossaries for the Progress hard-copy documentation set.

Progress Startup Command and Parameter Reference

A reference manual that describes the Progress startup and shutdown commands that you use at the command line, and the startup parameters that you use for Progress processes. This guide also provides information about parameter usage and parameter files.

Welcome to Progress (Hard copy only)

A booklet that explains how Progress software and media are packaged. An icon-based map groups the documentation by functionality, providing an overall view of the documentation set. *Welcome to Progress* also provides descriptions of the various services Progress Software Corporation offers.

Development Tools

Progress ADM 2 Guide

A guide to using the Application Development Model, Version 2 (ADM 2) application architecture to develop Progress applications. It includes instructions for building and using Progress SmartObjects.

Progress ADM 2 Reference

A reference for the Application Development Model, Version 2 (ADM 2) application. It includes descriptions of ADM 2 functions and procedures.

Progress AppBuilder Developer's Guide (Windows only)

A programmer's guide to using the Progress AppBuilder visual layout editor. AppBuilder is a Rapid Application Development (RAD) tool that can significantly reduce the time and effort required to create Progress applications.

Progress Basic Database Tools (Character only; information for Windows is in online help)

A guide for the Progress Database Administration tools, such as the Data Dictionary.

Progress Basic Development Tools (Character only; information for Windows is in online help)

A guide for the Progress development toolset, including the Progress Procedure Editor and the Application Compiler.

Progress Debugger Guide

A guide for the Progress Application Debugger. The Debugger helps you trace and correct programming errors by allowing you to monitor and modify procedure execution as it happens.

Progress Help Development Guide (Windows only)

A guide that describes how to develop and integrate an online help system for a Progress application.

Progress Translation Manager Guide (Windows only)

A guide that describes how to use the Progress Translation Manager tool to manage the entire process of translating the text phrases in Progress applications.

Progress Visual Translator Guide (Windows only)

A guide that describes how to use the Progress Visual Translator tool to translate text phrases from procedures into one or more spoken languages.

Reporting Tools

Progress Report Builder Deployment Guide (Windows only)

An administration and development guide for generating Report Builder reports using the Progress Report Engine.

Progress Report Builder Tutorial (Windows only)

A tutorial that provides step-by-step instructions for creating eight sample Report Builder reports.

Progress Report Builder User's Guide (Windows only)

A guide for generating reports with the Progress Report Builder.

Progress Results Administration and Development Guide (Windows only)

A guide for system administrators that describes how to set up and maintain the Results product in a graphical environment. This guide also describes how to program, customize, and package Results with your own products. In addition, it describes how to convert character-based Results applications to graphical Results applications.

Progress Results User's Guide for Windows and *Progress Results User's Guide for UNIX*

Platform-specific guides for users with little or no programming experience that explain how to query, report, and update information with Results. Each guide also helps advanced users and application developers customize and integrate Results into their own applications.

4GL

Building Distributed Applications Using the Progress AppServer

A guide that provides comprehensive information about building and implementing distributed applications using the Progress AppServer. Topics include basic product information and terminology, design options and issues, setup and maintenance considerations, 4GL programming details, and remote debugging.

Progress External Program Interfaces

A guide to the external programming interfaces supported by Progress. This manual covers the Host Language Call (HLC) Interface, the system clipboard, named pipes, shared libraries and DLLS, Windows Dynamic Data Exchange (DDE), COM objects, ActiveX Automation, ActiveX controls, sockets, XML, SAX, and the SonicMQ 4GL Adapter.

Progress Internationalization Guide

A guide to developing Progress applications for markets worldwide. The guide covers both internationalization—writing an application so that it adapts readily to different locales (languages, cultures, or regions)—and localization—adapting an application to different locales.

Progress Language Reference

A three-volume reference set that contains extensive descriptions and examples for each statement, phrase, function, operator, widget, attribute, method, and event in the Progress language.

Progress on the Web

A manual that describes how to use the new WebClient, AppServer Internet Adapter, SmartObjects, and SonicMQ Adapter to create applications tailored for Internet, intranet, and extranet environments.

Progress Programming Handbook

A two-volume handbook that details advanced Progress programming techniques.

Database

Progress Database Design Guide

A guide that uses a sample database and the Progress Data Dictionary to illustrate the fundamental principles of relational database design. Topics include relationships, normalization, indexing, and database triggers.

Progress Database Administration Guide and Reference

This guide describes Progress database administration concepts and procedures. The procedures allow you to create and maintain your Progress databases and manage their performance.

DataServers

Progress DataServer Guides

These guides describe how to use the DataServers to access non-Progress databases. They provide instructions for building the DataServer modules, a discussion of programming considerations, and a tutorial.

Each DataServer has its own guide as follows:

- *Progress/400 Product Guide*
- *Progress DataServer for Microsoft SQL Server Guide*
- *Progress DataServer for ODBC Guide*
- *Progress DataServer for ORACLE Guide*

MERANT ODBC Branded Driver Reference

The Enterprise DataServer for ODBC includes MERANT ODBC drivers for all the supported data sources. For configuration information, see the MERANT documentation, which is available as a PDF file in *installation-path*\odbc. To read this file you must have the Adobe Acrobat Reader Version installed on your system. If you do not have the Adobe Acrobat Reader, you can download it from the Adobe Web site at: http://www.adobe.com/products/acrobat/readstep.html.

SQL-89/Open Access

Progress Embedded SQL-89 Guide and Reference

A guide to Progress Embedded SQL-89 for C, including step-by-step instructions on building ESQL-89 applications and reference information on all Embedded SQL-89 Preprocessor statements and supporting function calls. This guide also describes the relationship between ESQL-89 and the ANSI standards upon which it is based.

Progress Open Client Developer's Guide

A guide that describes how to write, build, and deploy Java and ActiveX applications, and Java applets that run as clients of the Progress AppServer. This guide includes information about how to expose the AppServer as a set of Java classes or as an ActiveX server, and how to choose an Open Client distribution package fro run time.

Progress SQL-89 Guide and Reference

A user guide and reference for programmers who use interactive Progress/SQL-89. It includes information on all supported SQL-89 statements, SQL-89 Data Manipulation Language components, SQL-89 Data Definition Language components, and supported Progress functions.

SQL-92

Progress Embedded SQL-92 Guide and Reference

A guide to Progress Embedded SQL-92 for C, including step-by-step instructions for building ESQL-92 applications and reference information about all Embedded SQL-92 Preprocessor statements and supporting function calls. This guide also describes the relationship between ESQL-92 and the ANSI standards upon which it is based.

Progress JDBC Driver Guide

A guide to the Java Database Connectivity (JDBC) interface and the Progress SQL-92 JDBC driver. It describes how to set up and use the driver and details the driver's support for the JDBC interface.

Progress ODBC Driver Guide

A guide to the ODBC interface and the Progress SQL-92 ODBC driver. It describes how to set up and use the driver and details the driver's support for the ODBC interface.

Progress SQL-92 Guide and Reference

A user guide and reference for programmers who use Progress SQL-92. It includes information on all supported SQL-92 statements, SQL-92 Data Manipulation Language components, SQL-92 Data Definition Language components, and Progress functions. The guide describes how to use the Progress SQL-92 Java classes and how to create and use Java stored procedures and triggers.

Deployment

Progress Client Deployment Guide

A guide that describes the client deployment process and application administration concepts and procedures.

Progress Developer's Toolkit

A guide to using the Developer's Toolkit. This guide describes the advantages and disadvantages of different strategies for deploying Progress applications and explains how you can use the Toolkit to deploy applications with your selected strategy.

Progress Portability Guide

A guide that explains how to use the Progress toolset to build applications that are portable across all supported operating systems, user interfaces, and databases, following the Progress programming model.

WebSpeed

Getting Started with WebSpeed

Provides an introduction to the WebSpeed Workshop tools for creating Web applications. It introduces you to all the components of the WebSpeed Workshop and takes you through the process of creating your own Intranet application.

WebSpeed Installation and Configuration Guide

Provides instructions for installing WebSpeed on Windows and UNIX systems. It also discusses designing WebSpeed environments, configuring WebSpeed Brokers, WebSpeed Agents, and the NameServer, and connecting to a variety of data sources.

WebSpeed Developer's Guide

Provides a complete overview of WebSpeed and the guidance necessary to develop and deploy WebSpeed applications on the Web.

WebSpeed Product Update Bulletin

A booklet that provides a brief description of each new feature of the release. The booklet also explains where to find more detailed information in the documentation set about each new feature.

Welcome to WebSpeed (Hard copy only)

A booklet that explains how WebSpeed software and media are packaged. *Welcome to WebSpeed!* also provides descriptions of the various services Progress Software Corporation offers.

Reference

Pocket Progress (Hard copy only)

A reference that lets you quickly look up information about the Progress language or programming environment.

Pocket WebSpeed (Hard copy only)

A reference that lets you quickly look up information about the SpeedScript language or the WebSpeed programming environment.

PART I

Installation

Installation Requirements

Installing Progress

Sharing a Progress Installation On a Network

Administration Utilities

1

Installation Requirements

This chapter describes information you need before installing Progress software on a Windows-based platform.

Specifically, this chapter contains the following sections:

- System Requirements
- 4GL Client/Database Server Compatibility
- Product Life Cycle

1.1 System Requirements

This section describes the hardware and software requirements for running Progress on Windows NT, Windows 98, Windows 2000, Windows Millennium Edition (Me), and Windows XP. It also identifies the requirements for the Citrix MetaFrame platform Versions 1.0, 1.8, and XP.

NOTE: The system requirements provided in this chapter are current as of the publication date of this manual; however, requirements can change. To make sure that you have the most up-to-date information about system requirements, please be sure to refer to the *Progress Release Notes*.

1.1.1 Java

Many Version 9 products require one or more installed Java components for subsequent use by Progress. Specifically, the components are the Java Run-time Environment (JRE) and the Java Development Kit (JDK).

The JDK contains the software and tools that developers need to compile, debug, and run applets and applications written using the Java programming language. The JDK software and documentation, typically included with each new release of an operating system, are available for download at the vendor's Web site. You will require a JDK if you intend to develop Java stored procedures and triggers with the Database, or to create Java proxies with the Open Client Toolkit.

The JRE consists of the Java Virtual Machine, the Java Core Classes, and supporting files. The JRE is the runtime part of the JDK and does not include a compiler, a debugger, or development tools. You will need the JRE if you intend to use one of the following:

- Progress Explorer
- SQL Explorer
- WebSpeed Transaction Server
- AppServer
- Java application or applet

You will also need the JRE to execute Java stored procedures and triggers from the Progress database.

Depending on which products you choose to install, the installation process automatically updates your system with the JRE and the JDK, with the following exceptions:

- **Open Client Toolkit** — If you want to install the Open Client Toolkit to develop ActiveX proxies, download and preinstall the non-self-extracting Microsoft Software Development Kit (SDK) for Java Version 3.2 or later from the Microsoft Web site at `www.microsoft.com/java`. Be sure to include all the prerequisites for the SDK as specified on the Microsoft Web site.

  NOTE: Before you run Progress, make sure the PATH environment variable contains `SDK-Java.32\bin`.

  For information on open client proxy generation and system requirements, see the *Progress Open Client Developer's Guide*.

- **SQL-92** — The installation program does not automatically install the JDK component when you install the Enterprise Database, the Personal Database, or the Workgroup Database with SQL Client Access. If you intend to develop Java stored procedures and Java triggers for your database, you must install a Progress development product such as ProVision. For information on writing Java stored procedures and triggers, see the *Progress SQL-92 Guide and Reference*.

Table 1–1 lists operating systems and specifies the versions of JDK and JRE each supports. In the case of the Open Client toolkit, other versions, including versions from other vendors, can be used as well. When Javasoft provides a JDK and JRE for a certain platform, Progress Software includes the JDK and JRE in its distribution. For other systems, you must obtain the JDK from the system's operating system vendor. Please contact your operating system vendor for assistance in obtaining the JDK.

Table 1–1: JRE/JDK Requirements By Platform

Platform	JDK/JRE Required	Vendor URL
Windows NT/2000/98	1.3.0	`http://www.microsoft.com`
Windows ME	1.3.0	`http://www.microsoft.com`
Windows XP	1.3.0	`http://www.microsoft.com`
Citrix MetaFrame Versions 1.0, 1.8, and XP	1.3.0	`http://www.citrix.com`

NOTE: Since Progress Version 9.1C, Java 2 incorporates the Japanese Supplement, that is the Japanese Java Development Kit (JDK) and Just-in-Time (JIT) updates, as part of the standard shipment. These updates allow for the support of Swing, and also include Japanese localization of messages.

1.1.2 Windows NT On Intel

Table 1–2 lists the minimum requirements for running Progress on Windows NT on Intel.

Table 1–2: System Requirements For Windows NT On Intel *(1 of 2)*

Component	Requirement
Progress 4GL client	A Pentium PC (or compatible computer) with a clock speed of at least 100MHZ (200MHZ recommended). 32MB (64MB recommended).
Progress server	A Pentium PC (or compatible computer) with a clock speed of at least 200MHZ (400MHZ recommended). 64MB (96MB recommended).
Hard disk drive	Sufficient disk space to hold the Progress software you want to install. See Table 1–8 to determine the disk space that Progress products use.
Disk space	Sufficient disk space on the drive where Windows is installed for Windows system files that Progress copies there. Depending on which Windows system files are already installed on your system, Progress can install up to 10MB on your Windows directory.
Workstation or Server	Microsoft Windows NT Workstation or Windows NT Server Version 4.0.

Table 1–2: System Requirements For Windows NT On Intel *(2 of 2)*

Component	Requirement
Operating system	Microsoft Windows NT Version 4.0. Additionally: • If you are installing a product that contains Progress Explorer, SQL Explorer, or ProxyGen, you must have Microsoft Internet Explorer (MS IE) Version 4.01 or later installed on your system. **NOTE:** You can download MS IE Version 4.01 or later from `www.microsoft.com/ie`. • U.S.A. systems must have Service Pack 5.0 or later. • International systems must have the international equivalent to Service Pack 5.0 or later. • The MERANT/Intersolv Progress SQL-92 ODBC driver requires Service Pack 4 or later. **NOTE:** You can download the required service pack at `www.microsoft.com`.
Java	See the "Java" section for detailed information about the Java requirements if you are installing the Enterprise, Personal, or Workgroup Database; the Open Client Toolkit; or SQL-92 Client Access.
CD-ROM	A CD-ROM drive.

1.1.3 Windows 2000

Table 1–3 lists the minimum requirements for running Progress on Window 2000.

Table 1–3: System Requirements For Windows 2000

Component	Requirement
Progress 4GL client	A Pentium PC (or compatible computer) with a clock speed of at least 100MHZ (200MHZ recommended). 32MB (64MB recommended).
Progress server	A Pentium PC (or compatible computer) with a clock speed of at least 200MHZ (400MHZ recommended). 64MB (96MB recommended).
Hard disk drive	Sufficient disk space to hold the Progress software you want to install. See Table 1–8 to determine the disk space that Progress products use.
Disk space	Sufficient disk space on the drive where Windows is installed for Windows system files that Progress copies there. Depending on which Windows system files are already installed on your system, Progress can install up to 10MB on your Windows directory.
Workstation or Server	Microsoft Windows 2000 Workstation or Windows 2000 Server.
Operating system	Microsoft Windows 2000. Additionally: • If you are installing a product that contains Progress Explorer, SQL Explorer, or ProxyGen, you must have Microsoft Internet Explorer (MS IE) Version 4.01 or later installed on your system. **NOTE:** You can download MS IE Version 4.01 or later from `www.microsoft.com/ie`.
Java	See the "Java" section for detailed information about the Java requirements if you are installing the Enterprise, Personal, or Workgroup Database; the Open Client Toolkit; or SQL-92 Client Access.
CD-ROM	A CD-ROM drive.

1.1.4 Windows 98

Table 1–4 lists the minimum requirements for running Progress on Window 98.

Table 1–4: System Requirements For Windows 98

Component	Requirement
Progress 4GL client	A Pentium PC (or compatible computer) with a clock speed of at least 100MHZ (200MHZ recommended). 32MB (64MB recommended).
Progress server	A Pentium PC (or compatible computer) with a clock speed of at least 166MHZ (200MHZ recommended). 64MB (96MB recommended).
Hard disk drive	Sufficient disk space to hold the Progress software you want to install. See Table 1–8 to determine the disk space that Progress products use.
Disk space	Sufficient disk space on the drive where Windows 98 is installed for Windows system files that Progress copies there. Depending on which Windows system files are already installed on your system, Progress can install up to 10MB on your Windows directory.
Operating system	Microsoft Windows 98. **NOTE:** If you are installing a product that contains Progress Explorer, SQL Explorer, or ProxyGen, you must have Microsoft Internet Explorer (MS IE) Version 4.01 or later installed on your system. You can download MS IE Version 4.01 or later from `www.microsoft.com/ie`.
Java	See the "Java" section for detailed information about the Java requirements if you are installing the Enterprise, Personal, or Workgroup Database; the Open Client Toolkit; or SQL-92 Client Access.
CD-ROM	A CD-ROM drive.

NOTE: Progress Version 9 supports the Windows 98 platform, with the following exceptions:

- AppServer Internet Adapter
- AppServer Internet Adapter/S
- AppServer Load Balancing
- Enterprise Database
- NameServer
- NameServer Enterprise Edition
- Progress AppServer
- ProVISION Development Server
- ProVision Plus Development Server
- Secure Adapter
- Sonic MQ Adapter
- DataServer for Microsoft SQL Server Enterprise.
- WebSpeed Development Server
- WebSpeed Enterprise Transaction Server
- WebSpeed Messenger
- WebSpeed Transaction Server
- Workgroup Database

1.1.5 Windows Millennium Edition (Me)

Table 1–5 lists the minimum requirements for running Progress on Windows Millennuim.

Table 1–5: System Requirements For Windows Millennium

Components	Requirements
Progress 4GL client	A Pentium PC (or compatible computer) with a clock speed of at least 100MHZ (200MHZ recommended). 32MB (64MB recommended).
Progress server	A Pentium PC (or compatible computer) with a clock speed of at least 166MHZ (200MHZ recommended). 64MB (96MB recommended).
Hard disk drive	Sufficient disk space to hold the Progress software you want to install. See Table 1–8 to determine the disk space that Progress products use.
Disk space	Sufficient disk space on the drive where Windows Me is installed for Windows system files that Progress copies there. Depending on which Windows system files are already installed on your system, Progress can install up to 10MB on your Windows directory.
Operating system	Windows Millennium.
Java	See the "Java" section for detailed information about the Java requirements if you are installing the Enterprise, Personal, or Workgroup Database; the Open Client Toolkit; or SQL-92 Client Access.
CD-ROM	A CD-ROM drive.

1.1.6 Windows XP

Table 1–6 lists the minimum requirements for running Progress on Windows XP.

Table 1–6: System Requirements For Windows XP

Components	Requirements
Progress 4GL client	A Pentium PC (or compatible computer) with a clock speed of at least 100MHZ (200MHZ recommended). 32MB (64MB recommended).
Progress server	A Pentium PC (or compatible computer) with a clock speed of at least 166MHZ (200MHZ recommended). 64MB (96MB recommended).
Hard disk drive	Sufficient disk space to hold the Progress software you want to install. See Table 1–8 to determine the disk space that Progress products use.
Disk space	Sufficient disk space on the drive where Windows XP is installed for Windows system files that Progress copies there. Depending on which Windows system files are already installed on your system, Progress can install up to 10MB on your Windows directory.
Operating system	Windows XP.
Java	See the "Java" section for detailed information about the Java requirements if you are installing the Enterprise, Personal, or Workgroup Database; the Open Client Toolkit; or SQL-92 Client Access.
CD-ROM	A CD-ROM drive.

1.1.7 Citrix MetaFrame

Table 1–7 lists the minimum requirements for running Progress on Citrix MetaFrame Versions 1.0, 1.8, and XP; these requirements are similar to those needed when running Progress on Windows NT. For more information about system sizing on the Citrix MetaFrame platform, see the product documentation.

Table 1–7: System Requirements For Citrix MetaFrame

Components	Requirements
Progress 4GL client	To support 16 Progress 4GL clients, a Pentium Pro PC (or compatible computer) with a clock speed of at least 200MHZ and a minimum of 128MB of RAM for the operating system.
Progress 4GL clients and Progress server on the same system	To support 16 Progress 4GL clients and a Progress server on the same machine, a Pentium Pro PC (or compatible computer) with a clock speed of at least 200MHZ (400MHZ recommended). 256MB of RAM.
Hard disk drive	Sufficient disk space to hold the Progress software you want to install. See Table 1–8 to determine the disk space that Progress products use.
Disk space	Sufficient disk space on the drive where Citrix MetaFrame is installed for Windows-based system files that Progress copies there.
Operating system	Microsoft Windows NT Version 4.0, Terminal Server Edition, running Citrix MetaFrame Version 1.0. Additionally: • If you are installing a product that contains Progress Explorer, SQL Explorer, or ProxyGen, you must have Microsoft Internet Explorer (MS IE) Version 4.01 or later installed on your system. You can download MS IE Version 4.01 or later from `www.microsoft.com/ie`. • U.S.A. systems must have Service Pack 4.0 or later.
Java	See the "Java" section for detailed information about the Java requirements if you are installing the Enterprise, Personal, or Workgroup Database; the Open Client Toolkit; or SQL-92 Client Access.
CD-ROM	A CD-ROM drive.

1.1.8 Disk Space

Table 1–8 lists the approximate file size that each Progress product requires. Computations are based upon a 32KB disk cluster size and a complete installation that includes support for one installed language. Actual space requirements can vary, depending on the cluster size of your machine.

Additionally, some products share common files. Therefore, an installation of several products would not necessarily require the amount of approximate disk space you calculate if you add together the Total File Size of each product, based on the information provided in Table 1–8.

NOTE: During installation, Progress adds files to the Windows directory. Make sure that you have sufficient disk space on the drive where Windows NT, Windows 2000, Windows 98, Windows Me, Windows XP, or Citrix MetaFrame is installed. Depending on which Windows system files are already installed on your system, Progress can install up to 10MB on your Windows directory.

During installation, you can add more than one language. Each language that you select will require approximately 8MB of disk space on your operating system.

Table 1–8: Approximate Disk Space *(1 of 3)*

Product	Total File Size
Application Debugger	23MB
AppServer	145MB
AppServer Internet Adapter	14MB
AppServer Internet Adapter/S	14MB
Client Connect	231MB
Client-side Security	2MB
Client Networking	231MB
DataServer for ODBC (Enterprise)	231MB
DataServer for ODBC (Personal)	187MB
DataServer for ORACLE	223MB
Enterprise Database	254MB

Table 1–8: Approximate Disk Space *(2 of 3)*

Product	Total File Size
ESQL-89/C	24MB
4GL Development	381MB
NameServer	63MB
Name Server Load Balancing	20MB
Open Client Toolkit	120MB
Personal Database	254MB
ProVision	496MB
ProVision Plus	690MB
ProVision Development Server	313MB
ProVision Plus Development Server	392MB
Query/Results	186MB
Report Builder	38MB
Secure AppServer	140MB
SonicMQ Adapter	64MB
SQL-92 Client Access	79MB
SQL Server7 Personal DataServer	142MB
SQL Server7 Enterprise DataServer	229MB
Toolkit	24MB
Translation Manager	68MB
Visual Translator	181MB
WebClient	22MB
WebSpeed Development Server	356MB

Table 1–8: Approximate Disk Space *(3 of 3)*

Product	Total File Size
WebSpeed Messenger	14MB
WebSpeed Transaction Server	282MB
WebSpeed Transaction Server (Enterprise)	282MB
WebSpeed Workshop	547MB
Workgroup Database	254MB

1.2 4GL Client/Database Server Compatibility

If you run a Progress multi-user system with more than one Progress version, make sure that your 4GL client and database server are compatible. Table 1–9 lists the 4GL client versions that are compatible with the Progress database server versions.

All releases support TCP/IP networking.

Table 1–9: Client/Server Compatibility

4GL Client	Database Server
Progress Version 9	Progress Versions 8.x and Version 9
Progress Version 9.1	Progress Version 8.x and Version 9.x
WebSpeed Versions 2.0 and 2.1	Progress Version 8.x
WebSpeed Version 3.0	Progress Version 8.x and Version 9
WebSpeed Version 3.1	Progress Version 8.x and Version 9.x

1.3 Product Life Cycle

Progress Software Corporation has developed a Product Life Cycle designed to help communicate the product development and technical support resources available during a product's life span. This section outlines the different product stages starting from the first commercial shipment (FCS) of the product to eventual product retirement, as well as the life cycle status for each Progress Software product. This information will help you develop your product plans with the knowledge and understanding of Progress Software product life cycle plans. To assist you with migration and transition strategies, Progress Software offers a wide variety of education courses and consulting services. This section does not address Progress Software's future product plans. Please be advised that this information is subject to change by Progress Software without notice. For further information, please contact your Progress Software representative or visit our Web site at `www.progress.com`.

The term "product" is used in a general sense, it refers to a combination of a particular products and versions running on a particular hardware platform and/or operating system. For example, Progress ProVision Version 9.1 for Windows refers to a particular product on a particular platform and Progress Version 8.3 for HP-UX platforms refers to several products on HP-UX platforms.

All currently shipping products are classified in various phases of the Product Life Cycle. The Product Life Cycle definitions include specific R&D efforts, technical support services, inventory policies, and certification policies. New operating environments that will be considered for certifications include new platforms, operating systems, third-party databases, browsers, Web servers, and JVMs. Our goal is to place each product in the appropriate phase of its life cycle so that you know the level of support provided for each product. There are some situations where the placement of products in the product life cycle differs from the general policy.

Table 1–10 summarizes the Product Life Cycle phases.

Table 1–10: Product Life Cycle Phases

Phase	New Features	New Certifications	Support	Inventory
Active	Yes	Yes	BasicService	Yes
Functionally Stable	No	Yes	BasicService	Yes
Mature	No	No	Limited BasicService	Yes
Retired	No	No	Limited BasicService	No, except for purchase of additional user counts for existing licenses.

The four phases in the Product Life Cycle are:

- **Active** — Active Phase products are fully supported and receive new features and enhancements. Progress Software recommends that customers begin all new projects with Active Phase products and migrate existing applications to Active Phase products as soon as possible. If product problems are found, Progress Software Technical Support will work with customers licensed under a BasicService maintenance contract towards an appropriate resolution. Progress Software will evaluate Active Phase products for certifications on new operating environments.

- **Functionally Stable** — Functionally Stable Phase products are fully supported but do not receive new features. If problems are found in Functionally Stable Phase products, Progress Software Technical Support will work with customers licensed under a BasicService maintenance contract towards an appropriate resolution where possible. Progress Software will evaluate Active Phase products for certifications on new operating environments.

- **Mature** — Mature Phase products do not receive new features and support services are limited. You can continue to deploy existing applications built on Mature Phase products. However, Progress Software strongly recommends that customers use newer versions to build next-generation products. Mature Phase products will **not** be evaluated for certifications on new operating environments.

  Due to the advanced age of Mature Phase products, only Limited BasicService is available. With the knowledge and skills available, Progress Software Technical Support will provide its best efforts to resolve customer issues and answer customer questions on Mature Phase products licensed under a Limited BasicService maintenance contract. However, the knowledge and skills required to resolve issues on products supported by Limited BasicService are **not** guaranteed.

  Due to the terms of Limited BasicService, Progress Software strongly recommends that customers migrate to fully supported Progress Software products as soon as possible in order to maintain the highest level of support for their applications and systems.

- **Retired** — Products placed in the Retired Phase are not available for sale except for the purchase of additional user counts for existing licenses, which do not require the delivery of media or documentation. You should not continue to develop and deploy applications with licensed Retired Phase products. Progress Software Corporation will **not** evaluate Retired Phase products for certifications on new operating environments.

  Progress Software suggests possible migration paths to similar products for Retired Phase products and platforms. Since different versions of Progress Software products have varying system requirements, it is important to consider operating system and hardware requirements when evaluating your migration options.

  With the knowledge and skills available, Progress Software Technical Support will provide its best efforts to resolve customer issues and answer customer questions on Retired Phase products licensed under a Limited BasicService maintenance contract. However, the knowledge and skills required to resolve issues on products supported by Limited BasicService are **not** guaranteed.

  Due to the terms of Limited BasicService, Progress Software strongly recommends that customers migrate to fully supported Progress Software products as soon as possible in order to maintain the highest level of support for their applications and systems.

  Progress Software provides Limited BasicService support for a specified time period after the product's retirement date. Progress Software reserves the right to adjust the cost of Limited BasicService, which will be applied upon contract renewal. Due to the terms of Limited BasicService, Progress Software strongly recommends that customers migrate to fully supported Progress Software products as soon as possible in order to maintain the highest level of support for their applications and systems.

2

Installing Progress

This chapter contains step-by-step instructions for installing Progress on Windows and Citrix MetaFrame platforms.

This chapter contains the following sections:

- Installing Visual Translator
- Installing WebClient
- Performing Preinstallation Tasks
- Beginning the Installation
- Installing Required Applications During Progress Installation
- Installing Progress On Your System
- Performing a Typical Or Complete Installation
- Performing a Custom Installation
- Finishing the Installation
- InstallShield Silent (Batch Mode) Utility
- Installing Additional Products

- Installing Additional Components To Previously Installed Products
- Viewing Registry Information
- Performing Postinstallation Tasks
- Downloading Executables For Heterogeneous Environments
- Uninstalling Progress

2.1 Installing Visual Translator

If you are installing only the Visual Translator, you can follow the instructions in the introduction chapter of the *Progress Visual Translator Guide*.

NOTE: Visual Translator users do not need to install any other components of the Progress product for the Visual Translator to be functional.

2.2 Installing WebClient

The WebClient is a stand-alone product. The WebClient is stored in a folder named *\webinstall* on the root directory of your installation CD. It cannot be installed through the standard Progress Install program.

When you deploy an application that uses WebClient, you can customize the installations (of the application and of WebClient) so that the end-user experiences a single, unified installation. For more information, see the *Progress on the Web* manual, *Progress Version 9 Product Update Bulletin*, and the hard-copy *Progress Release Notes*.

2.3 Performing Preinstallation Tasks

Complete the following preinstallation tasks before starting your Progress installation:

- Shut down nonessential applications or tasks.
- Reset read-only files.
- Install other required applications.
- Save an existing Progress installation (if applicable).

2.3.1 Shutting Down Nonessential Applications Or Tasks

Shut down any nonessential applications or tasks that might be running, such as virus protection packages. Other applications or tasks might interfere with the installation or use files that Progress needs to complete the installation. Shut down any processes where the executable itself, or a file used by the executable, is located in the directory where you intend to install Progress.

2.3.2 Resetting Read-only Files

Before you install any Progress products on a Windows NT system, you should check the directory `C:\Winnt\system32` to see whether any of the `.dll` files have the read-only bit set. If any `.dll` files in this directory are read-only, you must reset them before installing Progress. If you try to install Progress with one or more related `.dll` files set to read-only, Progress will generate a dialog box indicating that you must reset the `.dll` bit and reinstall Progress.

2.3.3 Installing Other Required Applications

If any of the applications listed below are necessary to your installation, install them before you install Progress.

Microsoft Internet Explorer

If you are installing a product that contains Progress Explorer or ProxyGen, you must have Microsoft Internet Explorer (MS IE) Version 4.01 or later installed on your system in order to use the graphical administrative tools. If you do not have MS IE Version 4.01 or later, you will receive a warning message during the installation to install MS IE Version 4.01 or later. You can obtain information about acquiring or upgrading to MS IE Version 4.01 or later from the Microsoft Web site. You can continue with the installation after viewing this message, but neither ProxyGen nor the Progress Explorer graphical administrative tools will be functional until you have installed MS IE Version 4.01 or later.

WebSpeed

If you intend to install WebSpeed products, install a Web server product and configure a Web server directory on a machine where you want these products to reside. If you are upgrading an existing WebSpeed installation, shut down your Web server and reboot your machine to remove the ISAPI messenger from memory. (For more information, see the *WebSpeed Installation and Configuration Guide*.)

Open Client Toolkit

After you install the Open Client Toolkit, in order to allow the Open Client Proxy Generator (ProxyGen) to build Java or ActiveX proxies, you might have to install and configure additional tools. For more information, see the chapter on configuration and deployment in the *Progress Open Client Developer's Guide*.

SQL-92

The installation program does not automatically install the JDK component when you install the Enterprise Database, the Personal Database, or the Workgroup Database with SQL Client Access. If you intend to develop Java stored procedures and Java triggers for your database, you must install a Progress development product such as ProVision. For information on writing Java stored procedures and triggers, see the *Progress SQL-92 Guide and Reference*.

2.3.4 Saving An Existing Progress Installation

If you have an existing Progress installation, you might want to save certain pieces of it to make configuring your new installation simpler. If you want to continue using any templates, customized procedure or code files, or a `progress.ini` file, you must copy them to another location before you begin a new installation. Follow these steps:

1 ♦ If you have any Version 8 templates that you want to continue using, copy them to another location before installing Version 9. See Table 2–1 for a complete list of Version 8 templates.

Table 2–1 lists the Version 8 templates located in `%DLC%\src\template` that are no longer supplied with Version 9. (`%DLC%` represents the Progress installation path.)

Table 2–1: Version 8 Templates

childwin.w	detail.w	menu-bar.w	navuprec.w
`custdemo.txt`	dlgclose.w	navicons.wx	order.wx
custom.txt	dlgokcan.w	navigate.wx	orderlin.wx
dbopsupp.i	dlgrecup.w	navread.w	templdef.i
defbrupd.i	doaction.i	navreadi.w	winbrows.w
deffrupd.i	mainwin.w	navupd.w	—

NOTE: The `custom.txt` file, which is delivered with Version 9, is a subset of *`install-path`*`\src\template\progress.cst`.

2 ♦ If you have information in a `progress.ini` file that you want to continue using, copy the current `progress.ini` file to a directory other than where you are installing Progress.

3 ♦ If you have any customized procedure or code files in the directory where you are installing Progress, copy them into a different directory.

For more information on saving previous versions of your `progress.ini`, customized procedures, or code files, see the "Finishing the Installation" section.

4 ♦ If you have an earlier version of Progress and the PROMSGS environment variable is set in the `autoexec.bat` file on Window 98, or on the Environment tab of the System settings in the Windows NT or Windows 2000 Control Panel, you must remove the PROMSGS environment variable before installing Progress Version 9. If PROMSGS points to an old or nonexistent PROMSGS file, the InstallShield utility will not write all the necessary data to the Windows registry.

5 ♦ Truncate the before-image (.bi) file using the PROUTIL TRUNCATE BI utility and back up your Version 8.*x* database using the PROBKUP utility. For more information on these utilities, see the "Administration Utilities" chapter in the *Progress Database Administration Guide and Reference*.

CAUTION: Progress Version 9 requires that your databases be converted to a multi-volume structure. If you are using single-volume databases with Progress Version 8, you must convert your Version 8 databases to a multi-volume structure before converting the databases to Version 9. (Note that you must truncate your BI file in Version 8 before converting to Version 9.) If you plan to replace Progress Version 8 files with Version 9 files, complete this step before installing Version 9 and avoid erasing your Progress Version 8 files.

6 ♦ Make copies of your `ubroker.properties`, `conmgr.properties`, and `proxygen.preferences` files to another directory. The new installation automatically upgrades the files in the *`install-path`*`\properties` directory. After you have finished your new installation, replace the newly installed versions of these files with your copies. When you start the AdminServer, your older files will be updated to match the current standards for these files.

When you uninstall an existing Progress Version 9 product, the process copies `ubroker.properties`, `conmgr.properties`, and `proxygen.preferences`, the three files in the *`install-path`*`\properties` directory, to `\TEMP`. After installing a new Progress Version 9 product, you can manually copy back the files from `\TEMP`.

2.4 Beginning the Installation

How you install Progress depends on what products, components, and subcomponents you choose to install. You can tailor your Progress installation so that only those products and components essential to your business operation are installed. See your specific product documentation for information about which mandatory components you must install and where to install them.

For example, if you want to install Progress/400 or the MS SQL Server, ORACLE, or ODBC DataServer products, you must choose either the Complete installation option or the Custom installation option. If you choose the Custom option, you then select specific DataServer clients from the components list.

As you proceed through the installation, you will specify one of the installation options described in Table 2–2.

Table 2–2: Installation Options

Installation Option	Purpose
Typical	Installs the mandatory components and their mandatory subcomponents of the Progress products you are installing. This default installation option is recommended for most users. If you want to perform a Typical installation, continue with the "Performing a Typical Or Complete Installation" section. If you want to see which product components and subcomponents are installed during a Typical installation, see Appendix F, "Progress Version 9 Typical and Complete Installation Components."
Complete	Installs the mandatory and optional components and subcomponents of the Progress products you are installing. If you want to perform a Complete installation, continue with the "Performing a Typical Or Complete Installation" section. If you want to see which product components and subcomponents are installed during a Complete installation, see Appendix F, "Progress Version 9 Typical and Complete Installation Components."
Custom	Distributes Progress components on different machines and works around disk space limitations. This method is recommended for advanced users. If you perform a Custom installation and do not choose any optional components or subcomponents, only those components and subcomponents that are mandatory for the products you are installing are installed (as in a Typical installation). If you want to perform a Custom installation, continue with the "Performing a Custom Installation" section.

2.4.1 Other Considerations Before You Begin the Progress Installation

Review the following additional points before you attempt to install Progress 9.1D:

- You must have administrative privileges on the machine on which you are installing Progress. For more information, see your Windows documentation.
- You can install Progress products in any order.

- During the install of Progress Version 9.1D on Windows NT, Windows 2000, or Windows XP with products that use the AdminServer, you will be presented an option to set user and/or group AdminServer security. Set this option to require an individual user and/or groups of users to provide valid values during the AdminServer startup process. The products that use the AdminServer are the AppServer, WebSpeed, Sonic MQ Adapter, and the Progress Explorer.

  In Progress Version 9.1C and earlier versions, any valid user can access and execute operations within the AdminServer. The introduction of the AdminServer user-group authorization feature allows you to require a level of security that enables only authenticated operating systems users and groups access to and use of the Admin Service.

  To effectively set up this security option for your AdminServer use, review your security needs and current authenticated operating system users and groups to determine how you will set up this option during the Progress Version 9.1D installation process.

- If you are installing Version 9 and you have an existing Progress installation, you must perform some additional steps before you begin the Version 9 installation. See the "Performing Preinstallation Tasks" section.

Shared Network Installation Utility

You can provide multiple-client access to a single copy of Progress that is preinstalled on a network-accessible drive (server). If you want to do this, you must choose either a Complete installation (in which the Shared Network Installation component is automatically installed) or a Custom installation (which allows you to choose to install the Shared Network Installation component).

For more information about sharing a Progress installation between a server and clients, see Chapter 3, "Sharing a Progress Installation On a Network."

NOTE: Citrix MetaFrame does not support the shared network installation.

2.5 Installing Required Applications During Progress Installation

The following sections describe applications that Progress will install if you do not already have the minimum required versions of these applications on your system. These applications are installed during the Progress installation.

2.5.1 Installing the Microsoft Management Console

Unless you have the Microsoft Management Console (MMC) Version 1.1 or later installed on your system, the Progress Installation utility installs it. You must have one of the following language versions of Windows for the Progress Installation utility to successfully install the MMC: English, French, German, Spanish, Dutch, and Brazilian Portuguese. If you do not have the MMC Version 1.1 or later installed, the Progress Explorer graphical administration tools cannot function. In this case you receive a warning message to install MMC Version 1.1 in the appropriate language.

You can obtain information about acquiring different language versions of MMC Version 1.1 from Microsoft's Web site. You can continue with the installation after viewing the warning message, but the Progress Explorer cannot function until you have installed MMC Version 1.1 or later.

The Progress Installation utility installs MMC under the following conditions:

- If MMC is not already installed on your system and you have one of the language versions of Windows listed previously, then the installation utility installs MMC Version 1.1.
- If MMC Version 1.1 or later is already installed, then the installation utility does not install MMC.
- If MMC Version 1.0 is already installed and you have one of the six language versions of Windows listed previously, then the installation utility installs MMC Version 1.1.

The Progress Installation utility calls the Microsoft-supplied install program to install the MMC. This installation also requires Microsoft Internet Explorer Version 4.01 or later and Microsoft Java 3177 or later.

2.5.2 Installing Microsoft HTML Help

Unless you have Microsoft HTML Help Version 1.22 or later installed on your system, the Progress Installation utility installs it. You must have one of the following language versions of Windows for the Progress Installation utility to successfully install Microsoft HTML Help: English, French, German, Spanish, Dutch, and Brazilian Portuguese. If you are using a different language version of Windows, then the Installation utility cannot install Microsoft HTML Help Version 1.22. In this case you receive a warning message to install Microsoft HTML Help Version 1.22 in the appropriate language.

You can obtain information about acquiring different language versions of Microsoft HTML Help Version 1.22 from Microsoft's Web site at the following URL: `http://www.microsoft.com/`. You can continue with the installation after viewing the warning message.

The Progress Installation utility will install Microsoft HTML Help under the following conditions:

- If Microsoft HTML Help is not already installed on your system and you have one of the six language versions of Windows listed previously, then the installation utility installs Microsoft HTML Help Version 1.22.
- If Microsoft HTML Help Version 1.22 or later is already installed, then the installation utility does not install Microsoft HTML Help.
- If an earlier version of Microsoft HTML Help is already installed and you have one of the six language versions of Windows listed previously, then the installation utility installs Microsoft HTML Help Version 1.22.

2.5.3 Installing Microsoft Java

You must have Microsoft Java (MS JVM) Build 3177 or later for this installation. If you do not have MS JVM 3177 or later already installed on your system, then the Progress Installation utility will automatically install MS JVM.

2.5.4 Installing the MERANT SQL-92 ODBC Drivers and the MERANT ODBC Branded Drivers

If you are installing Progress Version 9 on a Windows operating system you must have the following Microsoft Installers for this installation:

- You must have the MDAC 2.6 Installer from Microsoft to install Microsoft Data Access Components (MDAC) on Windows operating systems. The Progress Installation utility contains the MDAC 2.6 Installer, which it automatically launches during installation. The MDAC 2.6 Installer is located at `installation-path\%DLC%\odbc\install\mdac_typ.exe`. You can find more information about the MDAC at Microsoft's Web site at the following URL: `http://www.microsoft.com/data/prodinfo.htm`.
- You must have the DCOM98 Installer from Microsoft to install the Distributed Component Object Model (DCOM) on the Win95 operating system. The Progress Installation utility installer contains the DCOM98 Installer, which it automatically launches during installation. The DCOM98 Installer is located at `installation-path\%DLC%\odbc\install\dcom98.exe`. You can find more information about DCOM98 at Microsoft's Web site at the following URL: `http://www.microsoft.com/Com/DCOM/Dcom98/redis.asp`.

NOTE: The DCOM98 Installer is the correct version for installing DCOM on the Win95 operating system.

You must reboot your system after installing either the MDAC or the DCOM98 Installer.

Installing the MERANT SQL-92 ODBC Drivers

The Installation utility installs the MERANT SQL-92 ODBC driver files to the *installation-path*\bin directory. The Installation utility also installs the files to %windir%\system32 for Windows NT and Windows 2000 systems, and to %windir%\system for Windows 98 systems. Table 2–3 lists the SQL-92 driver files installed for the Progress database.

Table 2–3: Driver Files For Windows Operating Systems For SQL-92

Database	Driver Files
Progress SQL-92	pgodbc.lic pgutl15.dll pgutl15R.dll pgbas15.dll pgbas15R.dll pgpro915.dll pgpro915R.dll pgpro915S.dll procli92.dll (Progress Driver File)

Installing the MERANT Branded ODBC Drivers

The Installation utility installs the MERANT branded ODBC driver files to the *installation-path*\bin directory. Table 2–4 lists the MERANT branded ODBC driver files installed with the Enterprise DataServer for ODBC. You can refer to the odbcref.pdf file installed in the *installation-path*\bin directory for information on how to configure datasources to connect to the different databases that Progress supports.

Table 2–4: Driver Files For Windows Operating Systems With Enterprise DataServer For ODBC

Database	Driver Files
All Enterprise ODBC data sources	P1BAS16.dll P1BAS16R.dll P1UTL16.dll P1UTL16R.dll IVP1.LIC P1WNSCK.dll P1MSNMP.dll
DB2 Common Server	P1UDB16.dll P1UDB16R.dll P1UDB16S.dll QECSV1.bnd QECSWHV1.bnd QERRV1.bnd QERRWHV1.bnd QEURV1.bnd QEURWHV1.bnd
Informix 7	P1INF16.dll P1INF16R.dll P1INF16S.dll P1INFDTC.dll
Microsoft SQL Server 6.5	P1SS616.dll P1SS616R.dll P1SS616S.dll
Microsoft SQL Server 7.0 and later	P1MSSS16.dll P1MSSS16R.dll P1MSSS16S.dll DBMSSOCN.dll
Sybase	P1ASE16.dll P1ASE16R.dll P1ASE16S.dll

Additional DataServer Information For ODBC-related Components

To use dataservers for ODBC, note the following points:

- Progress recommends that you use MERANT DataDirect Version 3.7.
- Progress requires that you install either the Enterprise DataServer or the Personal DataServer. (For the Dataserver for ODBC, Windows NT is the supported server platform.)

Also, you will need the specific data-source software components and version numbers installed for the specific data sources you are using. Table 2–5 presents these software requirements.

Table 2–5: Data-source Components and Version Numbers

If you are using a dataserver for ODBC product from this data source...	Then you must install...
Informix	Informix Server Software Version 7.x for the data-source server software, and Informix Connect Version 2.x for data-source client software
MicroSoft Access	MicroSoft Access 97 for the data-source software.
MicroSoft SQL Server (Version 6.5 or lower)	MicroSoft SQL Server Version 6.5 for the data-source server software and MERANT DataDirect ODBC Driver Version 3.01 or later for SQL Server 6.5.
DB2	One of the following software requirements: • DB2/MVS Versions 3.1, 4.1, or 5.1. (These versions can include DDCS Multi-user Gateway Version 2.3 or higher.) Also, DB2/MVS Versions 3.1 and 4.1 require IBM Client Access Enabler Version 2.11 or later. • DB2/NT Version 5.0 or 6.1. (Either version can also include DB/2 Connect Version 5 or later.) • DB2/6000 Version 5.0 or 6.1 (Either version can also include DB/2 Connect Version 5 or later.)
Sybase	Sybase adaptive Server System 10 or System 11 for data-source server software and Open Client Library Version 11.1.

2.6 Installing Progress On Your System

Follow these steps to begin installing your Progress products:

1 ♦ See your *License Addendum* for additional information that you need to complete the installation. The *License Addendum* includes the serial number and control numbers for your installation. It also provides volume information, the media-ID number, and the licensed Agent limit (WebSpeed products only). Once you complete the installation process, keep the *License Addendum* in a secure area.

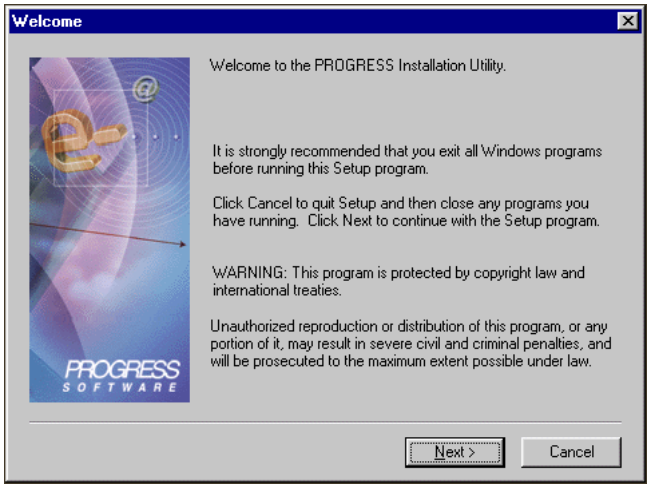
NOTE: When you install a client networking license, the ADM/ADM2 directories are not installed in the `$DLC/GUI` directory. This .r code is considered part of your application and should be deployed as a module of your application.

2 ♦ If you have not done so already, read the hard-copy *Progress Release Notes* for the installation. The *Progress Release Notes* might replace or supplement the installation instructions in this book.

3 ♦ Close all other applications before beginning the installation process.

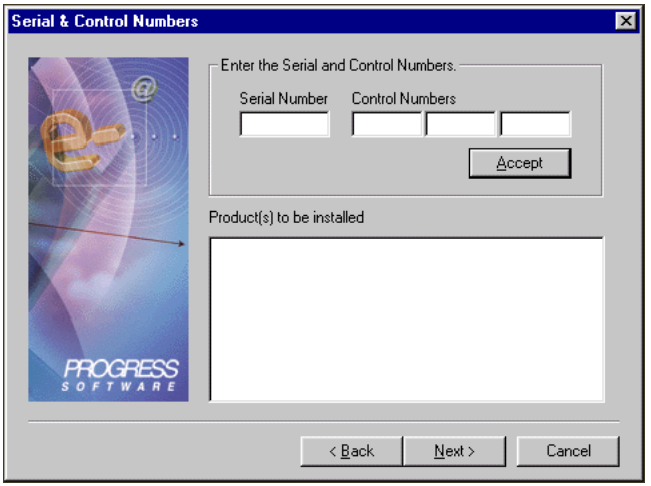
4 ♦ Insert the installation CD into your CD-ROM drive. If the CD does not run automatically, double-click `setup.exe` in the root directory of the CD.

5 ♦ The **Welcome** dialog box appears:



The **Welcome** dialog box reminds you to close any other running Windows programs.

6 ♦ Choose **Next** to continue. The **Serial & Control Numbers** dialog box appears:



7 ♦ For the product you are installing, type the nine-digit serial number and each of the five-digit control numbers from your *License Addendum*.

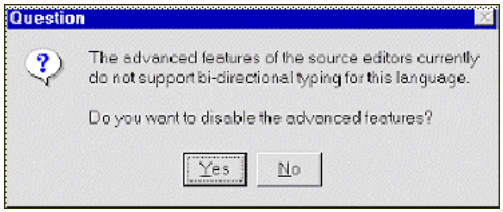
NOTE: This information is not case sensitive. The installation process automatically converts all letters to uppercase. The cursor advances to the next field automatically when you type the correct number of characters.

Choose **Accept**. The product name appears in the **Product(s) to be installed** list.

To install another product, type its serial and control numbers, then choose **Accept**.

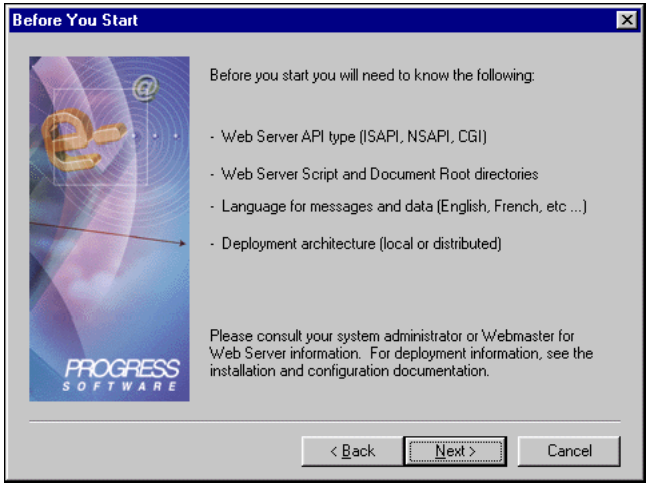
8 ♦ Once you choose all the products you want to install, choose **Next** to continue. Depending on which products you chose the following dialog boxes might appear:

- If you have a Hebrew or Arabic language operating system, you will be prompted to choose whether you want to disable the Progress Advanced Editor features. The Progress Advanced Editor is a new feature in Version 9.1 that is included with each of the following four products: ProVision, ProVision Plus, 4GL Development, and WebSpeed Workshop. The Progress Advanced Editor does not support bidirectional typing on Hebrew and Arabic Windows operating systems. If you install one of the four products that include the Progress Advanced Editor, the **Question** dialog box appears:



  You can answer Yes to disable the Progress Advanced Editor.

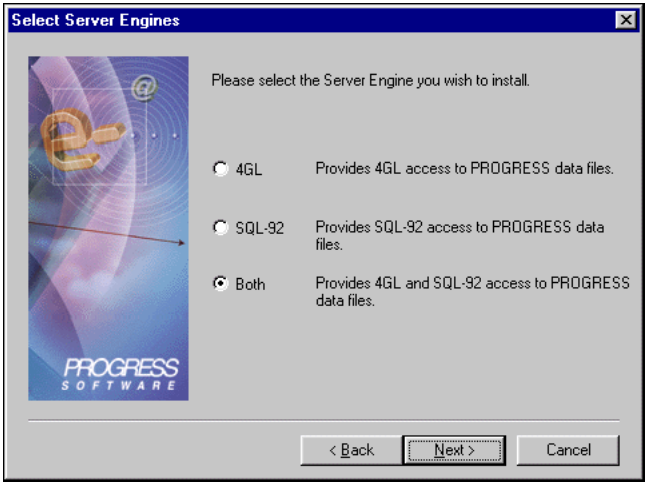
- If you are performing a Complete installation of ProVision Plus, the **Before You Start** dialog box appears, identifying information you must provide to complete the installation:



  Before you continue with a WebSpeed installation, see the “Installing WebSpeed” chapter of the *WebSpeed Installation and Configuration Guide*.

Choose **Next** to continue. If you are installing a database product, the **Select Server Engines** dialog box appears; otherwise, the **Choose Destination And Working Path Directories** dialog box appears.

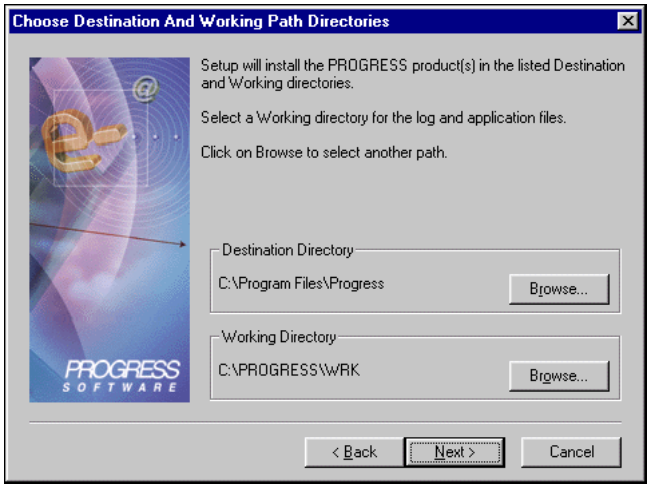
- If you are installing a Progress database product, the **Select Server Engines** dialog box appears:



Choose the **Server Engine** option. (Note that the **Both** option is selected by default.)

Choose **Next** to continue. The **Choose Destination And Working Path Directories** dialog box appears.

9 ♦ After you have entered any necessary information in the dialog boxes from the previous step, or if none of those dialog boxes applies to your installation, the **Choose Destination And Working Path Directories** dialog box appears:



The **Destination Directory** identifies where Progress will be installed. The **Working Directory** contains your applications, your databases, and your log files. The **Working Directory** also serves as the default working directory for Progress clients and WebSpeed Agents.

The default Destination Directory is `C:\Program Files\PROGRESS`, and the default Working Directory is `C:\PROGRESS\WRK`.

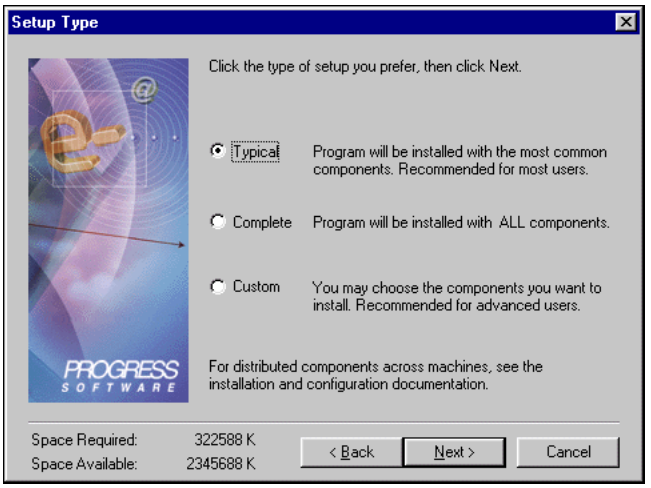
NOTE: If you are an international user, your default destination will be equivalent to the `\Program Files` directory for your language.

10 ♦ Accept the **Choose Destination And Working Path Directories** defaults, or choose **Browse** to select alternate directories, keeping in mind the following important considerations:

- Do not make your Progress working directory a subdirectory under the Progress installation path. If you uninstall Progress, the working directory is deleted along with the destination directory. Additionally, you cannot start a server for a database that has been installed in a subdirectory of Progress.
- If you have other versions of Progress products installed, **do not** enter the pathname where these products reside. You must specify a different Destination Path for Version 9. This ensures that you do not overwrite any of your existing Progress product directories and files.
- If you want to add Progress products to the most recently installed version, the install program assumes that you are adding products to that install location. For more information, see the "Installing Additional Products" section.

CAUTION: Never run Progress products from the directory in which you installed them. If you do, you could damage the Progress software files.

11 ♦ Choose **Next** to continue. The **Setup Type** dialog box appears:



To perform a Typical or Complete installation, see the "Performing a Typical Or Complete Installation" section. To perform a Custom installation, see the "Performing a Custom Installation" section.

2.7 Performing a Typical Or Complete Installation

You can choose the installation method that best suits your company's unique product and resource requirements. You can choose one of following installation options:

- **Typical** — Perform a Typical installation when you want to install most (but not all) of the Progress components. A Typical installation saves disk space by installing only the most widely used Progress components and subcomponents. If you want to see which product components and subcomponents are installed during a Typical installation, see Appendix F, "Progress Version 9 Typical and Complete Installation Components."

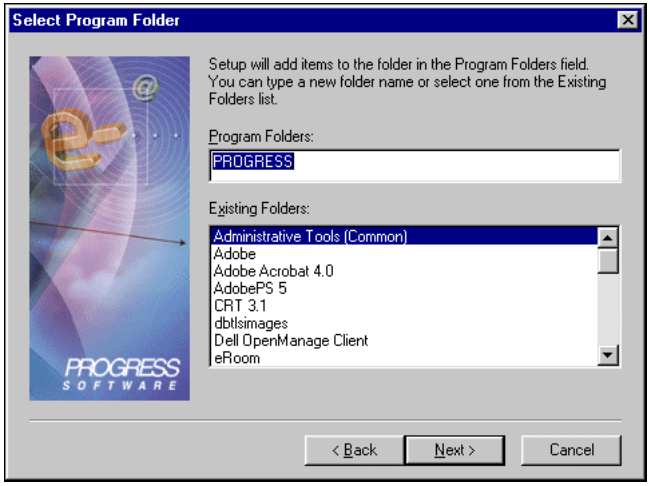
- **Complete** — Perform a Complete installation when you want to install all Progress components and subcomponents. For example, choose this option if you intend to install the WebSpeed, ProVision, or ProVision Plus development servers, or if you want to automatically install the Shared Network Installation utility. For more information, see the details about the Shared Network Installation utility in the "Beginning the Installation" section.

If you want to see which product components and subcomponents are installed during a Complete installation, see Appendix F, "Progress Version 9 Typical and Complete Installation Components."

Follow these steps to perform either a Typical or Complete installation:

1 ♦ Follow Steps 1 through 10 in the "Installing Progress On Your System" section.

2 ♦ In the **Setup Type** dialog box, choose **Typical** or **Complete**, then choose **Next** to continue with the installation. The **Select Program Folder** dialog box appears:

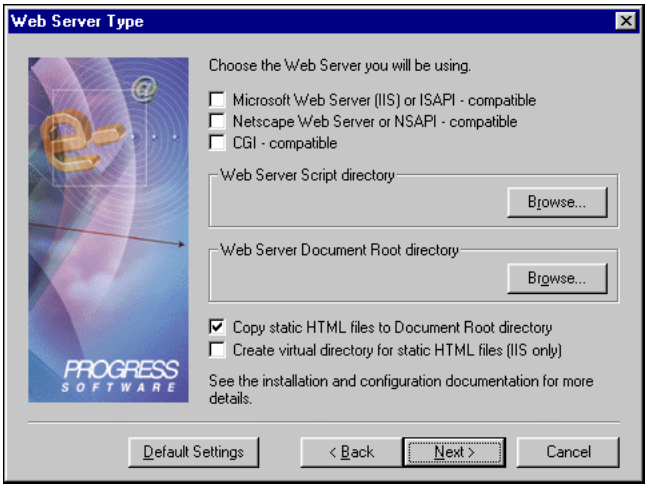


3 ♦ Accept the default Program Folder, type the name of a new Progress Program Folder, or select an existing folder.

If you have an existing Progress installation, make sure that the Program Folder name you type here is not the same one that contains the installed product.

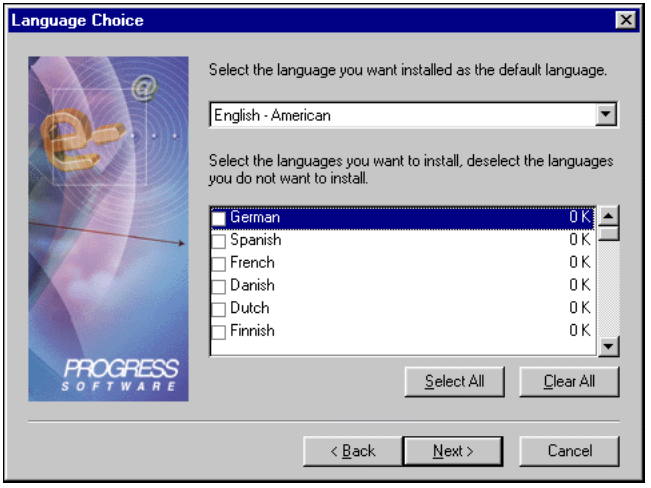
Then choose **Next** to continue.

If you are performing a Complete installation of ProVision Plus, the **Before You Start** dialog box appears, identifying information you must provide to complete the installation:



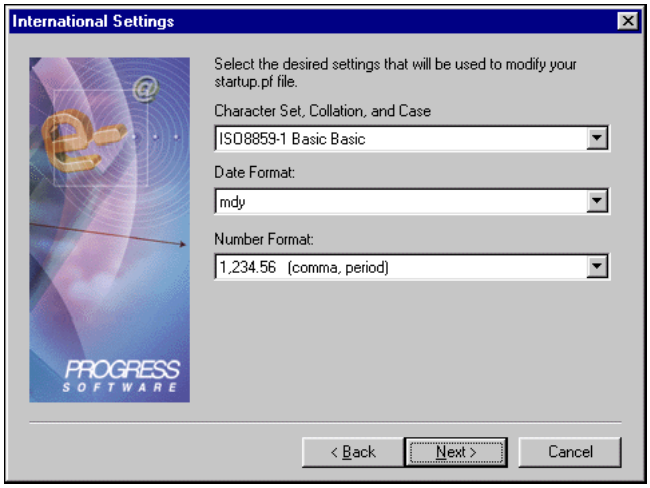
To reset the dialog box settings to those chosen in a previous installation, choose **Default Settings**. For information about choosing the other WebSpeed options, refer to the "Installing WebSpeed" chapter in the *WebSpeed Installation and Configuration Guide*.

4 ♦ Choose **Next** to continue. The **Language Choice** dialog box appears:



5 ♦ Select the languages you want to use for error and informational messages. Progress can use these languages for run-time messages. The initial value for the default language is determined by the operating system on which you are installing Progress. However, you can set the result to any of the provided languages.

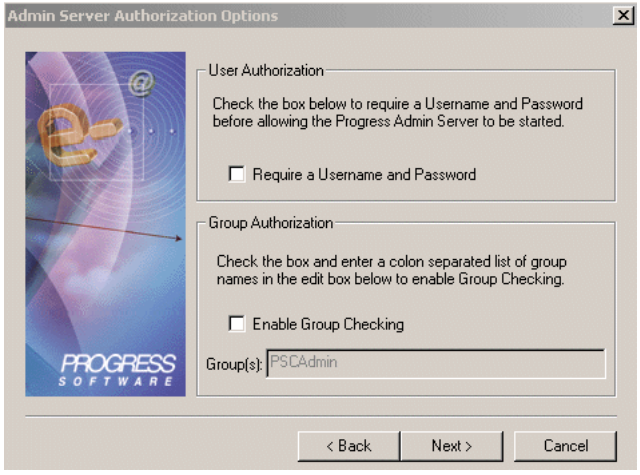
6 ♦ Choose **Next**. The **International Settings** dialog box appears:



The **International Settings** dialog box allows the installation program to tailor your Progress startup parameter (`startup.pf`) file to specify international conventions. (For more information on international conventions, see the *Progress Internationalization Guide*.)

7 ♦ From each drop-down list, select the international Character Set, Collation, and Case; Date Format; and Number Format to tailor the default `startup.pf` file.

8 ♦ Choose **Next**. The **Admin Server Authorization Options** dialog box appears:



When you install Progress Version 9.1D on Windows NT, Windows 2000, or Windows XP and install Progress products that use the AdminServer, the installation program prompts you to define optional security options. You can indicate whether individual

and/or group authorization are required to start the AdminServer. The products that use the AdminServer are the AppServer, WebSpeed, Sonic MQ Adapter, and the Progress Explorer.

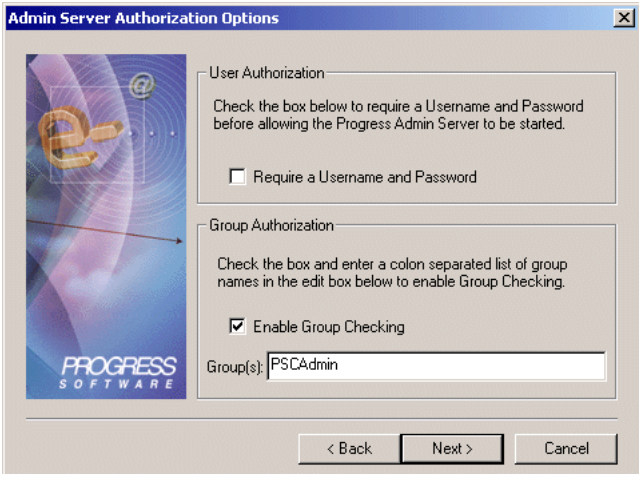
NOTE: The **User Authorization** and **Group Authorization** options are established independently of each other. You can elect to use one option, both options, or none. To bypass this option, select **Next**.

9 ♦ To require individual users to provide a valid user name and password before accessing the Progress AdminServer, click the check box next to the **Require a Username and Password** option.

During the separate process used to start the AdminServer, the user will be prompted to enter user name and password information. Only users who supply valid user name and password values will be allowed access to the Progress AdminServer.

NOTE: Specific user name and password information values are established by an administrator at the operating system level. Refer to your operating system documentation for information on how to set up users.

10 ♦ To require the **Group Authorization** option to secure Progress AdminServer access and operational privileges at a group level, click the check box next to the **Enable Group Checking** option.



Note the following:

- An initial default group value, called PSCAdmin, becomes active in the **Group(s)** field. You can elect to identify this default group as one of your authorization groups.

If you do not want to use this default group, you can delete this default value from the **Group(s)** field now.

- You can define additional user-defined group names. Separate each group name that you enter with a colon as noted on the **Admin Server Authorization Options** dialog box.

Note the following restrictions on the values and characters that you can use to identify a user-defined group name:

- A group name cannot consist solely of periods or spaces.
- Any spaces you enter prior to entering the group name will be considered part of your group name's definition. For example, if you enter a colon and then a space before you type **Group1**, the Group1 name will only be valid when you include this leading space.
- The name can contain a maximum of 256 uppercase or lowercase characters, with the exception of the characters identified in Table 2–6.

Table 2–6: Illegal Characters

Illegal Character	Character Format
quotation marks	"
left hash mark	/
right hash mark	\
colon	:
semicolon	;
pipe	|
equal sign	=
plus sign	+
asterisk	*
question mark	?

11 ♦ For each group you enter in the **Group(s)** field, including the default group value, a unique **Question** dialog box displays once you select the **Next** button. (If the group already exists, this dialog box does not display.) This step actually creates the group.

For example, Figure 2–1 shows the **Question** dialog box that displays for the AdminGroup1 entry.

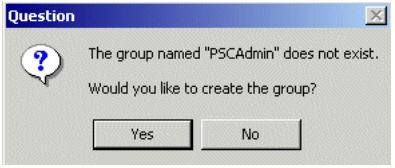


Figure 2–1: Question dialog box

12 ♦ To create a unique group with the group name as it is identified, select **Yes**. Select **No** to reconsider creating this specific group at this time.

Figure 2–2 shows the **Warning** dialog box that appears if you select **No**:

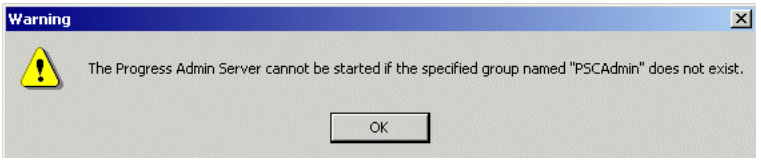
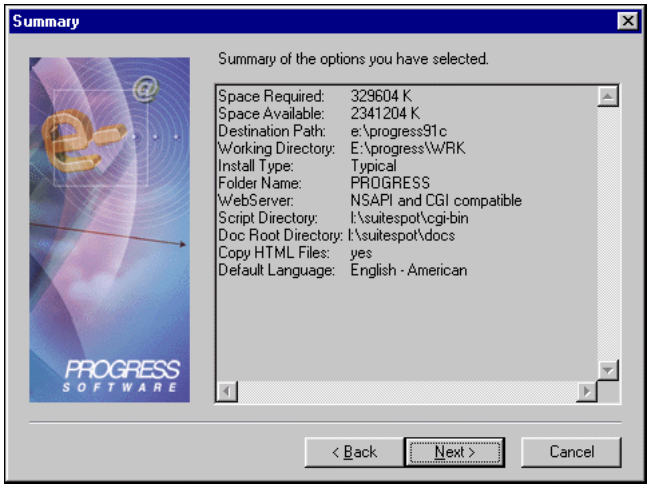


Figure 2–2: Warning dialog box

NOTE: If you select the **No** option at this time, but determine after you complete this installation that you actually do want to create a group, or create additional new groups, you need to perform these tasks:

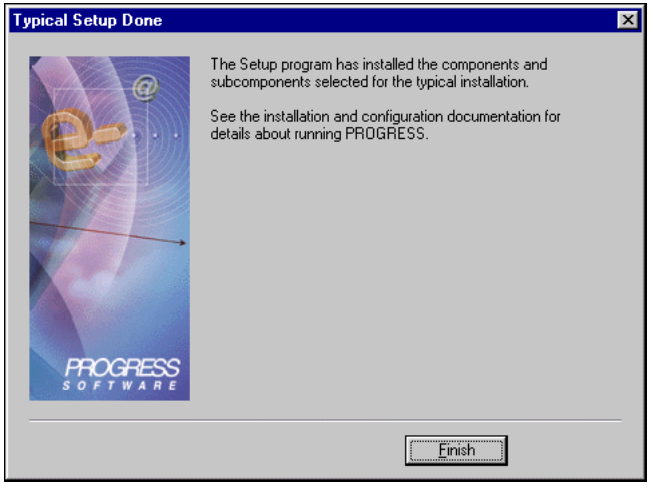
- Re-install Progress using the installation program.
- Add new groups when the **Admin Server Authorization Options** dialog box appears.
- Repeat Step 10 through Step 12 for each new group you enter.

13 ♦ The **Summary** dialog box appears, displaying a summary of the installation options you selected:



14 ♦ Choose **Next**. The setup program installs the Progress files.

When the installation finishes, either the **Typical Setup Done** or **Complete Setup Done** dialog box appears, corresponding to the type of installation you chose:



15 ♦ Choose **Finish**.

16 ♦ Reboot your machine if prompted to do so.

NOTE: Before you can start Progress, the AdminServer must be running.

On Windows NT, the AdminServer is configured and started as an NT service by default. After you launch the Progress Explorer and connect to the AdminServer, you will see a directory tree comprised of your installed Progress tools and databases.

For more information on the AdminServer and Progress Explorer, see the "Managing the AdminServer" section in Chapter 5, "Setting Up Your Environment," and the Progress Explorer online help.

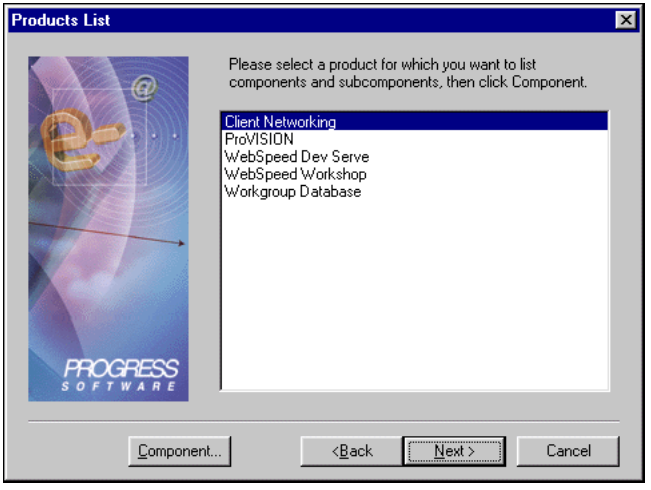
2.8 Performing a Custom Installation

Perform a Custom installation when you want to install individual Progress product components and subcomponents. Choosing a Custom installation allows you to install only the mandatory components required for each Progress product in order to help you work around possible disk space limitations.

Follow these steps to perform a Custom installation:

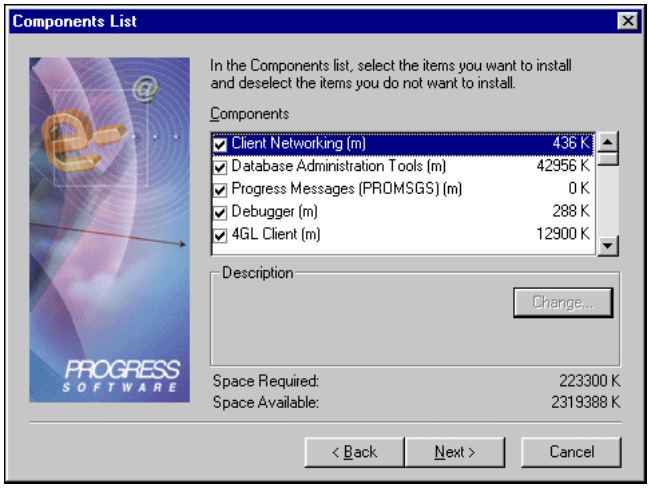
1 ♦ Follow Steps 1 through 11 in the "Installing Progress On Your System" section.

2 ♦ In the **Setup Type** dialog box, choose **Custom**, then choose **Next** to continue with the installation. **The Products List** dialog box appears:



3 ♦ Select the product whose components and subcomponents you want to install.

4 ♦ Choose **Component** to view a list of product components. The **Components List** dialog box appears:

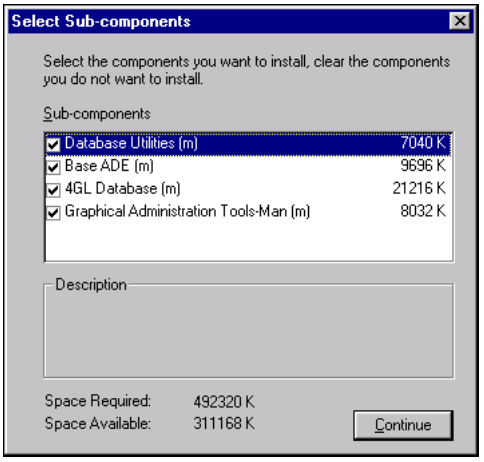


Some of the Progress components are marked as mandatory. You must include these components in your Progress installation, though not necessarily on the machine you are currently setting up. You can ignore the warning messages about installing mandatory components only if you are an experienced Progress user.

5 ♦ Select the components for a Custom installation. If you intend to use the Shared Network Installation option, be sure to choose to install the NetSetup component.

CAUTION: If you previously installed the required version of the JavaSoft (InstallShield) JDK on your system and want Progress to use this utility, you should contact Progress Software Technical Support for assistance.

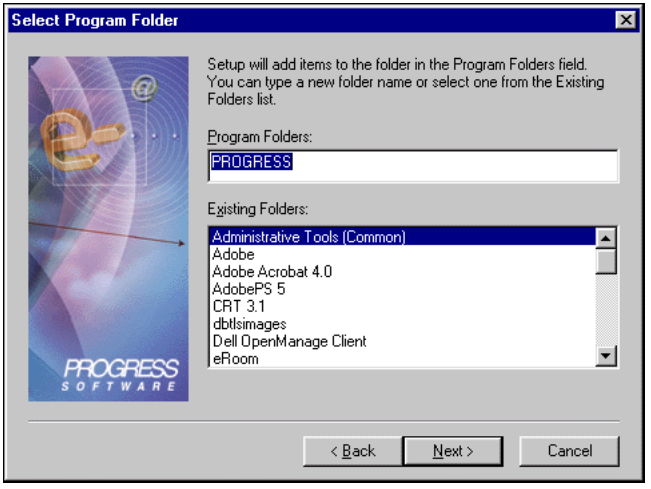
6 ♦ To select a component's subcomponents, highlight the component in the list and choose **Change**. The **Select Sub-components** dialog box appears:



Some of the Progress subcomponents are marked as mandatory. You must include these subcomponents in your Progress installation, though not necessarily on the machine you are currently setting up. You can ignore the warning messages about installing mandatory subcomponents only if you are an experienced Progress user.

7 ♦ Select the subcomponents that you want to install, and choose **Continue**. The **Components List** dialog box reappears. You can install additional subcomponents for any component listed. Choose **Next** when you are done. The **Products List** dialog box reappears.

8 ♦ You can choose components or subcomponents for another product, or you can choose **Next**. When you select **Next**, the **Select Program Folder** dialog box appears:

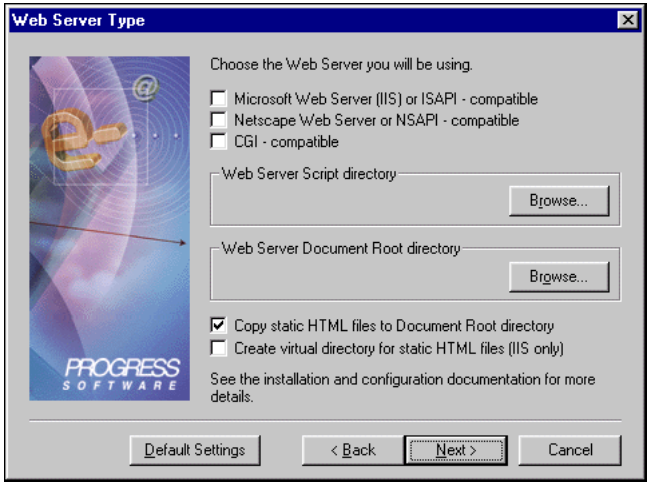


9 ♦ Accept the default Program Folder, type the name of a new Progress Program Folder, or select an existing folder.

If you have an existing Progress installation, make sure that the Program Folder name you type here is not the same one that contains the installed product.

Choose **Next** to continue.

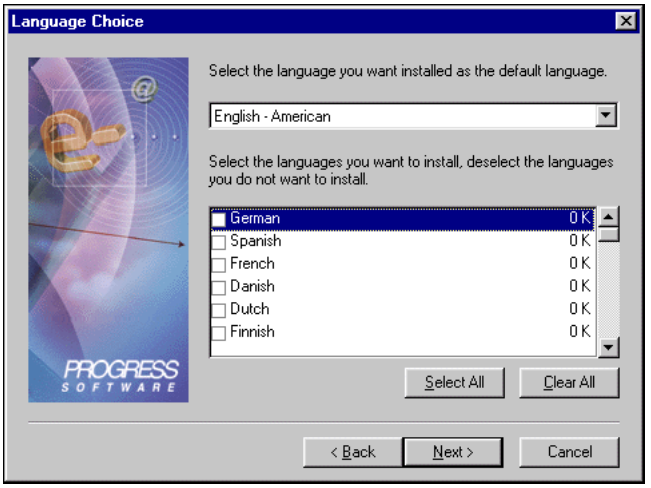
If you are installing the WebSpeed components of ProVision Plus, the **Web Server Type** dialog box appears:



10 ♦ Select the Web server you will use to install Messengers. Choose **Next**.

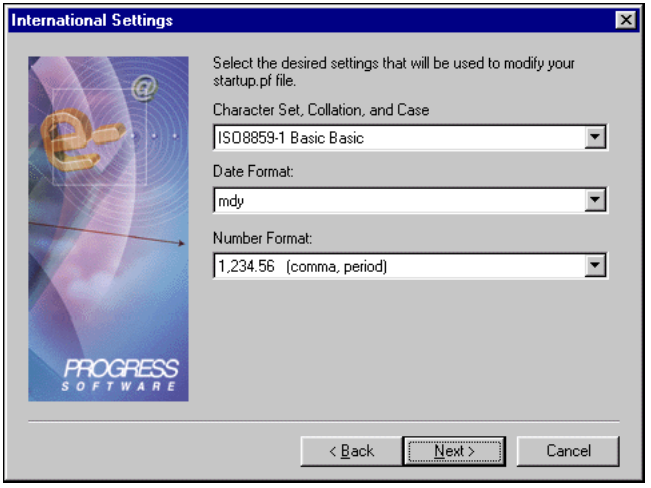
For more information on WebSpeed components and Web servers, see the “Installing WebSpeed” chapter of the *WebSpeed Installation and Configuration Guide*.

The **Language Choice** dialog box appears:



11 ♦ Select the default language and any other languages you want to install. Progress can use these languages for run-time messages. The initial value for default language is determined by the operating system on which you are installing Progress. However, you can set the result to any of the provided languages.

12 ♦ Choose **Next**. The **International Settings** dialog box appears:

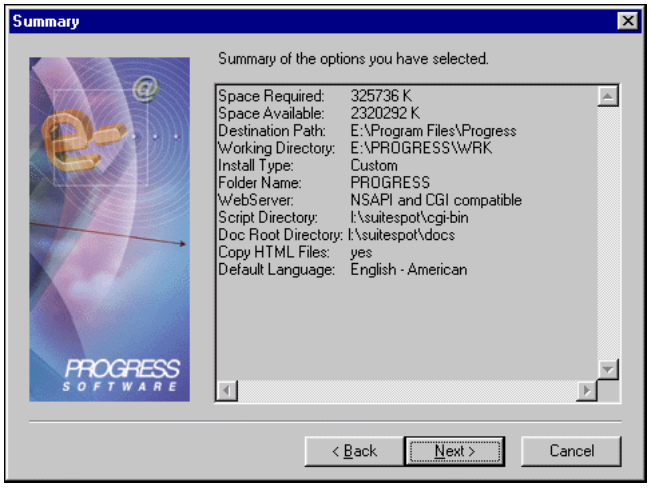


The **International Settings** dialog box allows the installation program to tailor your Progress startup parameter (`startup.pf`) file to specify international conventions. (For more information on international conventions, see the *Progress Internationalization Guide*.)

13 ♦ From each drop-down list, select the International Character Set, Collation, and Case; Date Format; and Number Format to tailor the default Progress `startup.pf` file.

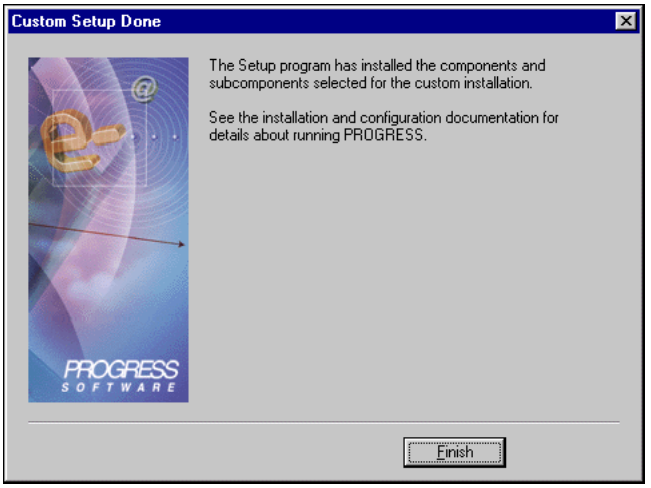
14 ♦ Follow Step 8 through Step 12 in the "Performing a Typical Or Complete Installation" section if you are installing any Progress products that use the AdminServer and you want to set up the optional AdminServer User-Group Security Authorization feature.

15 ♦ Choose **Next** to continue the setup program. The **Summary** dialog box appears, displaying a summary of the installation options you have selected:



16 ♦ Choose **Next**. The setup program installs the Progress files.

The **Custom Setup Done** screen appears:



17 ♦ Choose **Finish**.

18 ♦ Reboot your machine if prompted to do so.

NOTE: Before you can start Progress, the AdminServer must be running.

On Windows NT, the AdminServer is configured and started as an NT service by default. After you launch the Progress Explorer and connect to the AdminServer, you will see a directory tree comprised of your installed Progress tools and databases.

For more information on the AdminServer and Progress Explorer, see the "Managing the AdminServer" section in Chapter 5, "Setting Up Your Environment," and the Progress Explorer online help.

2.9 Finishing the Installation

If you saved information in a `progress.ini` file in a previous version of Progress that you want to continue using, follow these steps to add that information to the new `progress.ini` file:

1 ♦ Copy the information you want to save from your previous `progress.ini` file to the Version 9 `progress.ini` file.

2 ♦ If you are copying information from a Version 8 installation, rename any PROUIB section to PROAB before running `ini2reg`. (The PROUIB section of the Version 8 `progress.ini` file is referred to as PROAB in Version 9.)

3 ♦ Run `ini2reg` to update the information in the registry with the information you added from your previous `progress.ini` file.

4 ♦ Restart your system.

If you saved property information (such as `ubroker.properties`, `conmgr.properties`, or `proxygen.preferences`) that you want to continue using, copy the saved property files from `%TEMP%` to *`install-path`*`\properties`.

2.10 InstallShield Silent (Batch Mode) Utility

A normal (nonsilent) installation receives the necessary information by prompting and recording your input in each dialog box. By contrast, a silent installation does not prompt you for input. Rather, a silent installation reads user response information from a response file called `setup.iss`. Silent installation is only supported for Typical and Complete Installations. You cannot use this utility for a Custom Installation.

The InstallShield Silent (ISS) allows you to install Progress in silent or batch mode by recording user input information contained in a response (`setup.iss`) file.

NOTE: You must create a response file before running a silent or batch installation.

2.10.1 Running the InstallShield Silent Utility

Follow these steps to run the InstallShield Silent utility:

1 ♦ Create a Response file. You can produce this file either automatically (recommended method) by using the InstallShield Response (-r) command-line parameter or manually by editing a `setup.iss` file. For more information on automatically creating a response file, see the "Creating the Response File Automatically" section. For more information on manually creating a response file, see the "Creating the Response File Manually" section.

2 ♦ Play back the Silent Installation using the Silent (-s) command-line parameter. The InstallShield runs on its own without your intervention:

```
drive:\setup.exe -s -f1C:\<path-to-file>\setup.iss
        [-f2C:\<path-to-file>\setup.log]
```

In the above sample, *`path-to-file`* indicates the location of the setup file. The default location of the `setup.iss` file is the directory where you placed the Progress `setup.exe`. If you have moved this file to a different directory, you must specify the path to that directory here.

The setup.log file is by default created in the same directory as the setup.iss file. You can use the optional -s -f2 command-line parameter to specify a different location for the setup.log file.

NOTE: You must launch the InstallShield Silent installation utility by using the setup.exe -s -f1 command-line parameters.

Creating the Response File Automatically

InstallShield creates a response file for you by recording all your installation choices in the setup.iss file and places the file in the Windows directory. Use this syntax to initiate an InstallShield response file:

```
drive:\setup.exe -r -f1C:\<windows directory>\setup.iss
```

Creating the Response File Manually

You can create and edit a response file (setup.iss), which is similar to a .ini file. Note that this file requires you to add the necessary sections in a predefined order.

Follow these steps to create a response file manually:

1 ♦ Create a new text file named setup.iss using any text editor.

2 ♦ Enter InstallShield response file (setup.iss) required sections in the following order, noting that Section names are contained in square brackets, for example, [InstallShield Silent]:

a) Silent Header Section [InstallShield Silent]

b) Dialog Sequence Section [DlgOrder]

c) Application Header Section [Application]

d) Dialog Data Sections, for example [Welcome Dialog-0]

3 ♦ Edit the necessary data within each section name. Data entries consist of *name=value* pairs, such as Dlg0=Welcome-0.

4 ♦ Save and close the setup.iss file and place the file in the Windows directory.

2.10.2 Sample setup.iss File

The following sample shows a typical `setup.iss` file, which can include the following sections:

setup.iss *(1 of 2)*

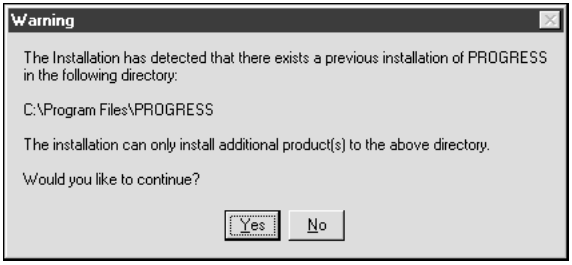
```
[InstallShield Silent]
Version=v5.00.000
File=Response File

[File Transfer]
OverwriteReadOnly=NoToAll

[Application]
Name=PROGRESS
Version=9.1D
Company=PSC
Lang=0009

[DlgOrder]
Dlg0=Welcome Dialog-0
Count=10
Dlg1=SerialControlNumber Dialog-0
Dlg2=TargetDirectory Dialog-0
Dlg3=UserInstallationType Dialog-0
Dlg4=SdSelectFolder-0
Dlg5=LanguageChoice Dialog-0
Dlg6=International Settings Dialog-0
Dlg7=GetAuthorizationInfoDlg-0
Dlg8=SdShowInfoList-0
Dlg9=RebootDialog-0

[Welcome Dialog-0]
Result=1

[SerialControlNumber Dialog-0]
nvCount=1
svSerialNumber-0=xxxxxxxxx
svControlNumber_1-0=xxxxx
svControlNumber_2-0=xxxxx
svControlNumber_3-0=xxxxx
_bInstallingDB=0
_bMessengerInstalled=0
_bProgressExplorerLocalOption=1
_nDbProdnumBitmask=0
_bInstallingColorEditor=1
_szUseSourceEditor=yes
Result=1
```

setup.iss *(2 of 2)*

```
[TargetDirectory Dialog-0]
szDirDest=C:\Program Files\Progress
szDirWrk=C:\PROGRESS\WRK
Result=1

[UserInstallationType Dialog-0]
_bQuickInstallRadioButton=0
_bCustomInstallRadioButton=0
_bTypicalInstallRadioButton=1
Result=1

[SdSelectFolder-0]
szFolder=PROGRESS
Result=1

[LanguageChoice Dialog-0]
_szDefaultLanguage=English - American
Languages-type=string
Languages-count=1
Languages-0=English - American
Result=1

[International Settings Dialog-0]
_szNlsCpinternal=ISO8859-1
_szNlsCpcollation=Basic
_szNlsCpcase=Basic
_szNlsDateFormat=mdy
_szNlsNumberFormat=1,234.56   (comma, period)
_szNumsep=44
_szNumdec=46

[GetAuthorizationInfoDlg-0]
GroupList=PSCAdmin
UsernamePassword=1
GroupChecking=1
Result=1

[SdShowInfoList-0]
Result=1

[RebootDialog-0]
Result=0
Choice=602
```

2.11 Installing Additional Products

You can add other Progress products to your current installation by following the steps outlined in the "Installing Progress On Your System" section of this chapter. When the installation process detects the existing version of Progress, a **Warning** dialog box appears, notifying you of the existing version's location:



NOTE: When you add products to an existing installation, you can use the installation utility in batch mode as long as you are performing a Typical or Complete installation of the products you are adding.

Follow these steps to continue with the installation:

1 ♦ Choose **Yes** to continue with the installation. The **Welcome** dialog box appears.

2 ♦ Choose **Next** to continue. The **Serial & Control Numbers** dialog box appears.

3 ♦ Enter the control numbers and choose **Accept** for each product you want to add to your current installation.

4 ♦ Choose **Next** to continue. If you are installing a WebSpeed product, the **Before You Start** dialog box appears.

5 ♦ Review the information and choose **Next**. The **Choose Destination And Working Path Directories** dialog box appears. The install program deactivates (grays out) the **Choose Destination And Working Path Directories** dialog box modifier functionality and adds your Progress products to Version 9 directories automatically.

6 ♦ Accept the default directories and continue with the installation.

2.11.1 Changes In the Progress Core Installation Program For Progress Messages Files

Progress Version 9.1D incorporates specific file protection measures to accommodate files associated with Progress add-on products. For purposes of this document, the term add-on products refers to Progress products released independently of a point or major Progress product release. Add-on products provide functionality that enhances and is adjunct to the Progress Core software product set.

In Progress Version 9.1D, these file protection and detection procedures within Progress Core have been formalized to ensure that you have the most recent `promsgs` files. All Progress products use one centralized method to display Progress messages contained in the `promsgs` file. With each Progress add-on product you install to the Version 9.1D core installation environment, an updated `promsgs` file is installed into the Progress Core product's destination directory. Add-on installation processes have been modified such that if the add-on product contains a newer `promsgs` file than the associated core release, the following activities occur:

- The add-on product's `promsgs` file is compared with the core product's `promsgs` file to determine which of the two files is newer.
- The newer file is copied to the Version 9.1D core directory.

To support this process, Progress Version 9.1D also introduces procedures to ensure that message files are not incorrectly overwritten, and files such as `messages.jar` and Prohelp\`msgdata` that are associated with `promsgs` files are also protected.

The information in this section describes

- Details About the Installation and Update of Promsgs Files
- Procedures to Protect Files from Being Overwritten
- Procedures to Ensure Promsgs Files are Synchronized

Details About the Installation and Update Of Promsgs Files

The Progress Core Virtual Machine (PCVM) displays Progress-specific messages that are contained in promsgs files. During the Progress Core Installation process, the user selects the language(s) that can be used during the product's execution. It is possible to have several translated promsgs files installed into the progress destination path\prolang subdirectory due to this selection process. During the Core installation process, the promsgs files for the language identified as the default language are copied from the progress destination path\prolang subdirectory to the DLC directory.

Promsgs files contain the most up-to-date messages at the time the Progress core installation is released. However, the promsgs files are constantly being updated. Consequently, add-on products and Progress install service packs that are released after the Progress Core product release date can contain even more recently updated promsgs files. As each Progress add-on product is installed, the installation program checks to ensure that the newest copy of the promsgs file is being used by all products; all products use the centrally-located copy of the promsgs file stored in the DLC directory.

Procedures To Protect Promsgs Files From Being Overwritten

During the Progress Core Installation process prior to Progress Version 9.1D, certain functions have the potential to incorrectly overwrite or update a promsgs file in the local directory. To eliminate the possibility of these activities occurring, Progress Version 9.1D introduces two changes to protect promsgs files and any associated files, and to ensure that you always have the most recent promsgs files:

- A file protection mechanism has been added to the Version 9.1D Progress Core Product installation program which prohibits overwriting any Progress-associated promsgs file that already exists.

  In Progress Version 9.1D, the Progress Core Installation program default behavior determines that if a promsgs file exists in the local directory, it is the latest version. Therefore, there is no need to perform any file changes.

- The Progress Core Installation program supports a versioning scheme that adds date information to the header of the promsgs file. The install program uses this date information to help determine the latest version of a promsgs file.

If the user installs an add-on product using the Progress Add-on Installation program, this program automatically reads and compares the header information of the source promsgs file (specifically, the file associated with the add-on product) with the header information of the destination promsgs file (the file already installed in the Progress Core Installation program). Based on details noted in the header, the Add-on Installation program determines whether or not to install the promsgs file. These details include a date stamp, the single criterion for determining the most up-to-date promsgs file. The promsgs file with the latest date is always selected to be installed.

NOTE: Header information provides important information to keep promsgs files synchronized. For information about file synchronization, see the section "Procedures To Ensure Promsgs Files Are Synchronized" later in this section.

These measures also prohibit the following files that are associated with promsgs files from being overwritten by the Progress Core Installation program:

- Messages.jar
- Promsgs
- Prohelp\msgdata

Procedures To Ensure Promsgs Files Are Synchronized

Beginning in Progress Version 9.1D, promsgs files are considered to be either in-synchronization or out-of-synchronization. These terms reflect the status of the date stamp associated with a promsgs file when the date in the header of the promsgs files located in the add-on directory is compared with the date in the header of the promsgs files currently installed in the Core installation's DLC directory.

In Progress Version 9.1D, the Progress Core Installation and the Add-on Installation processes are designed to compare and evaluate the date stamp information. A promsgs file is considered synchronized if, at the conclusion of any product installation process, the Progress Core installation contains the promsgs file with the most current, or latest, date stamp. A promsgs file is considered out-of-synchronization, and therefore invalid, when the date stamp associated with the promsgs file does not display the most current date.

Table 2–7 identifies the general installation sequence that can occur at a customer site when Progress Core products and add-on products are installed. It illustrates how the promsgs files are compared, evaluated, and updated to ensure that the promsgs files are always synchronized.

Table 2–7: Promsgs File Synchronization Process *(1 of 2)*

Install Sequence	When. . .	Then. . .
1	Progress Version 9.1D core products are initially installed.	The promsgs files associated with the languages selected by the user during the install process are installed to the DLC directory.
2	A Progress add-on product is installed.	Date stamp information in the header of the existing promsgs file in the DLC directory is compared with the date stamp information in the header of the add-on product's promsgs file. If the core promsgs file's date is *later* than the add-on product's promsgs file's date, the file is already synchronized and no changes occur. If the core promsgs file's date is *earlier* than the add-on product's promsgs file's date, the add-on promsgs file replaces the existing promsgs file.

Table 2–7: Promsgs File Synchronization Process *(2 of 2)*

Install Sequence	When. . .	Then. . .
3	Progress Version 9.1D core products are re-installed to add a new product and a new promsgs file.	Two comparisons and their associated activities occur: If the re-installation process finds that a promsgs file exists, the existing promsgs file is not overwritten. If, during the re-installation process, a new language is added, the promsgs file associated with that new language is installed into the DLC directory.
4	Another Progress add-on product is installed.	The date stamp information in the header of the existing promsgs file in the DLC directory is compared with the date stamp information in the header of the add-on product's promsgs file. If the core promsgs file's date is *later* than the add-on product's promsgs file's date, the file is already synchronized and no changes occur. If the core promsgs file's date is *earlier* than the add-on product's promsgs file's date, the add-on promsgs file replaces the existing promsgs file.

Table 2–8 illustrates another example of how this process works, using more detailed data for you to review.

The first column of Table 2–8 elaborates on the typical installation sequence outlined earlier in this section. In Step 1, the user initially installs ProVision Plus with a promsgs file for American English. The file header date of this newly installed promsgs file is 04/14/2002. In Step 2, when the user installs an add-on product, the Add-on product installation compares the header date of its American English promsgs file, 04/15/2002, with the header date of the existing American English promsgs file, 04/14/2002. Since the header date of the promsgs file associated with the add-on product is later than the existing promsgs file, the promsgs file is updated, or synchronized.

This example helps to illustrate the criterion for updating promsgs files; Only promsgs files associated with languages that are currently installed in the Progress Core will be updated by the Add-on installation process.

In Step 3, when the user installs another Progress Core product, the Progress AppServer, and identifies the Spanish promsgs file, the promsgs file with the date of 04/14/2002 is installed. This latter part of the example illustrates how the promsgs files can become out of sync per the date information in the respective headers.

Table 2–8: Example Of Promsgs Files Being Out Of Sync

Installation step order	Install...	And the promsgs file is...	Which contains this header date...
1.	A core product such as ProVision Plus, Version 9.1D	Installed for American English	04/14/2002
2.	An add-on product.	Updated for American English	04/15/2002
3.	A core product such as AppServer, Version 9.1	Installed for Spanish promsgs	04/14/2002

As Table 2–8 indicates, the installation of previously non-existing Spanish promsgs file dated 04/14/2002 into the Progress core installation is now out of synchronization with the updated American English promsgs file dated 04/15/2002, that updated the original American English promsgs file.

Addressing Synchronization Issues

When an additional Progress installation is performed and the Progress Core Installation program detects that a promsgs language has been installed that did not previously exist as illustrated by Step 3 in Table 2–8, the Progress Core Installation program displays a message. This message indicates the following information:

- The add-on product name that contains the latest promsgs
- Indicates the destination path of the add-on product

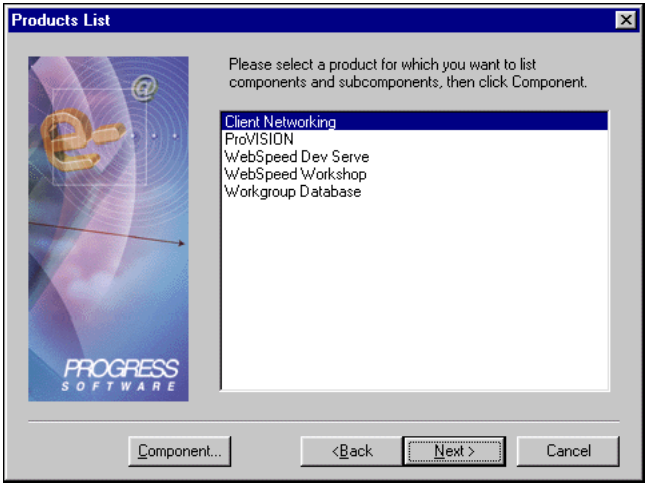
The Progress Core installation message only displays this message when it detects that add-on products have been installed and it reads a new file called `addons`. The `addons` file is a text file defined as a Windows initialization (.ini) file. This file is created and/or updated in the Progress core destination directory by the Add-on installation program. To resynchronize your `promsgs` file, you must re-install your add-on product.

2.12 Installing Additional Components To Previously Installed Products

You can add components and subcomponents to existing Progress Version 9.1 and later installations without entering any data other than the required components or subcomponents. In earlier Progress versions it is necessary to reinstall Progress and execute the "Custom Install" setup type; these steps are removed with this Add feature.

Follow these steps to add components or subcomponents to a previously installed product:

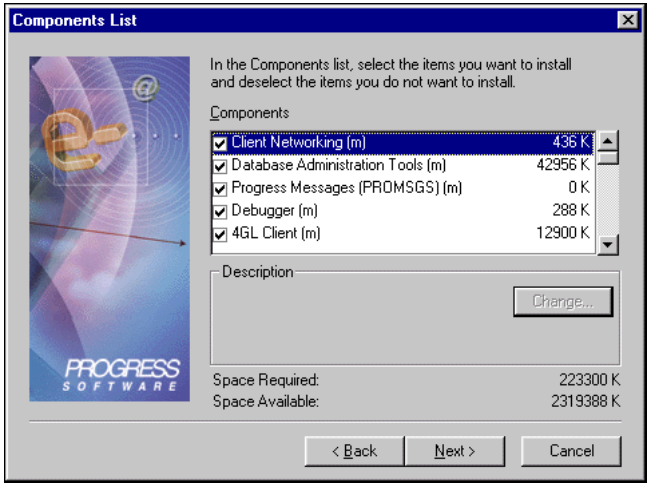
1 ♦ From the **Start** menu, select **Programs**, then select **Progress**. Select **Add Components** from the Progress menu. The **Products List** dialog box appears:



All previously installed products appear on the **Products List**.

2 ♦ Select the product to which you want to add components or subcomponents from the **Products List**.

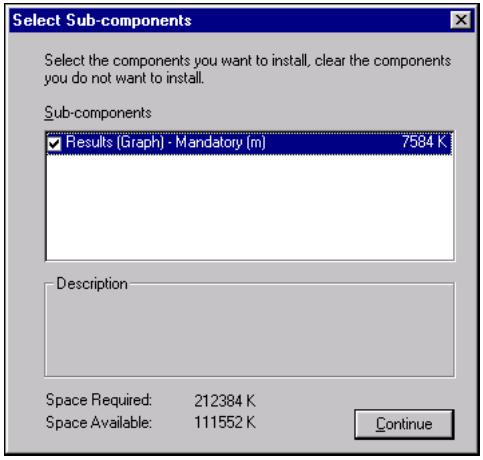
3 ♦ Choose **Component** to add components or subcomponents to the already installed product. The **Components List** dialog box appears:



Only components that you have not previously installed appear on the **Components List**.

4 ♦ Select the components you want to add. If the selected component contains subcomponents, the **Change** button becomes enabled.

5 ♦ Choose **Change** to add subcomponents. The **Select Sub-components** dialog box appears:



Only subcomponents that have not previously been installed appear on this list.

6 ♦ Select the subcomponents you want to add, then choose **Continue**.

7 ♦ Choose **Finish** to update the existing install. The **Progress Meter Bar appears**; the installation status is displayed in this meter bar.

NOTE: If a system file in a newly added component or subcomponent is locked or busy during installation, a **Reboot** dialog box appears to prompt you to reboot your system.

2.13 Viewing Registry Information

Applications running on Windows platforms rely on the registry for startup information, such as color, font, and key bindings. The Progress installation adds the required `progress.ini` file information into the registry as entries under the following keys:

```
HKEY_CURRENT_USER\SOFTWARE\PSC\PROGRESS\9.1D
HKEY_LOCAL_MACHINE\SOFTWARE\PSC\PROGRESS\9.1D
```

NOTE: There is different `progress.ini` file information in each key.

The installation also automatically adds entries when you install an ODBC driver. For example, if you install the ODBC Personal DataServer, the following entries appear:

```
HKEY_CURRENT_USER\SOFTWARE\ODBC
HKEY_LOCAL_MACHINE\SOFTWARE\ODBC
HKEY_LOCAL_MACHINE\SOFTWARE\PSC\ODBC Personal DS\91D
HKEY_LOCAL_MACHINE\SOFTWARE\PSC\Progress ODBC\91D
```

To add `progress.ini` file information into the registry, run the `ini2reg` utility. (The `ini2reg` updates the `HKEY_CURRENT_USERS` key.)

Follow these steps to review registry information:

1 ♦ On your desktop, choose **Start→Run**. The **Run** Dialog box appears.

2 ♦ Type **REGEDIT** (or **REGEDT32**) in the **Open** field:



3 ♦ Choose **OK**. A directory tree appears.

4 ♦ Double-click the `HKEY_CURRENT_USER\SOFTWARE\PSC\PROGRESS\9.1D` key or the `HKEY_LOCAL_MACHINE\SOFTWARE\PSC\PROGRESS\9.1D` key to view its contents.

2.14 Performing Postinstallation Tasks

Before you run Progress Version 9, there are a few postinstallation tasks you must complete:

- **Set Environment Variables** — For more information on setting environment variables (including SQL), see Chapter 5, “Setting Up Your Environment.”
- **Create Customized Executables** — To create customized product executables, see the “Building Progress Executables” appendix in *Progress Client Deployment Guide*. Creating executables might be required for certain product configurations.

- **Convert Existing Databases** — After your Progress Version 9 installation is complete, you must convert your Version 8 databases to Version 9 using the PROUTIL CONV89 utility. Note that if you have a Version 6 or Version 7 database, you must convert it to a Version 8 database first. For instructions on converting your Version 8 databases to Version 9, see the "Administration Utilities" chapter in the *Progress Database Administration Guide and Reference*.

- **Validate and Populate Group Names Created for the AdminServer Security Option** — The tasks to verify that your groups were created, and to identify the specific members of these groups, are accomplished outside of the Progress Version 9.1D installation. The installation process only allows you to creation the groups that you specify.

  NOTE: For detailed information about how to verify that groups have been created, and how to access and set up group members for each group on a Windows platform, refer to your operating system-specific documentation. Also, the criteria you use to set up users within each group is determined by your company.

For example, the following generalized steps outline suggest one procedure you can use to verify that your groups exist and to create users for each group if you have successfully installed Progress Version 9.1D on Windows 2000:

1 ♦ From the Control Panel, choose **Administrative Tools** and select **Computer Management**.

2 ♦ Choose **System Tools** to display a list.

3 ♦ Select **Local Users and Groups**, and choose **Groups**.

Verify that the groups you created during the installation process are among those listed.

4 ♦ To populate a given group, double click the group name and complete the **Group Properties** dialog box.

2.15 Downloading Executables For Heterogeneous Environments

The distributed architecture of Progress Version 9 allows you to optimize your hardware and network resources by installing components across networked machines, specifically when you are installing the WebSpeed Transaction Server or the Open AppServer. Although some of these products' components must reside together on the same machine, you can, in some cases, distribute components to different machines, even if the machines run on different platforms. For example, you can install a WebSpeed Messenger or the NameServer on a Windows NT platform and install a WebSpeed Transaction Server on a UNIX platform.

If you need either the WebSpeed Messenger executable or the NameServer executable for a platform other than Windows, you can download the executables free of charge from `http://techweb.progress.com/services/support/ts-dnlod.htm`.

2.16 Uninstalling Progress

When you delete files from the Progress directory tree, you only partially remove a Progress installation. By contrast, the Uninstall utility or the Remove Program utility removes all Progress files as well as configuration information from the registry, which prevents conflict with subsequent Progress installations. If for some reason the Uninstall utility cannot completely uninstall Progress, you must manually remove the installation.

If you want to upgrade or remove an installation, choose one of the following:

- Run the Uninstall utility from the Progress program group, as described in the "Using the Uninstall Or Add/Remove Programs Utility" section.
- Run the Add/Remove Programs utility from the Microsoft Windows Control Panel, as described in the "Using the Uninstall Or Add/Remove Programs Utility" section.
- Manually remove the installation, as described in the "Manually Removing Progress Version 9.1D" section. For additional details, see the Progress Software Corporation online Knowledge Base at `http://techweb.progress.com/services/support/kb.htm`.

  NOTE: You can uninstall Progress on a Citrix MetaFrame platform in the same way as on Windows NT. Citrix MetaFrame operates in a client/server environment and is not the same as a shared network environment. Therefore, you cannot share a Progress installation on a network that operates on Citrix MetaFrame.

2.16.1 Using the Uninstall Or Add/Remove Programs Utility

You can run the Uninstall utility (or use the Add/Remove Programs utility located in the Windows Control Panel) to automatically remove Progress from your system. Running the Uninstall or the Remove Program utility removes configuration information from the registry and prevents conflict with subsequent Progress installations.

CAUTION: When uninstalling, do not delete any of the following Microsoft system files: `mfc40.dll`, `mfc42.dll`, `mfcans32.dll`, `msvcrt.dll`, `msvcrt20.dll`, `msvcrt40.dll`, `olepro32.dll`. These system files are common to other applications, and deleting them might adversely affect the operation of the other applications that use them. To avoid deleting these system files while running the Uninstall utility, answer **NO** to the prompts at the end of the uninstall process.

Follow these steps to run the Uninstall utility:

1 ♦ Log in under the same domain and user name you used when you installed Progress.

2 ♦ Make sure that Progress is not running in an open DOS window (or that the current directory is not in any Progress-related file or directory).

3 ♦ Stop all Progress processes and close all Progress Help files. You can use the Task Manager to ensure that you stop all processes and close all help files.

4 ♦ Using the Progress Explorer, shut down all Progress services (brokers, NameServers, and databases).

5 ♦ Shut down the AdminServer by following these steps:

a) From the desktop, select **Start→Settings→Control Panel→Services**.

b) Highlight the **AdminService for Progress 9.1D**, and choose **Stop**.

c) Choose **Yes** to confirm that you want to stop the service.

d) When the service stops (in approximately 30-60 seconds), choose **Startup**. The **Service** dialog box appears.

e) Change the Startup from Automatic to Manual, and choose **OK**. (This step is necessary if you reboot your machine before completing the uninstall so that the AdminServer does not start up automatically.)

NOTE: If these same services will be required for a new installation, be sure to note any configuration settings, agent parameters, etc.

6 ♦ For WebSpeed product installations, make sure that all Progress services and your ISAPI or NSAPI Web server are shut down. For more information, see the "Installing WebSpeed" chapter in the *WebSpeed Installation and Configuration Guide*.

7 ♦ Choose the **Uninstall** icon in the Progress Program Group.

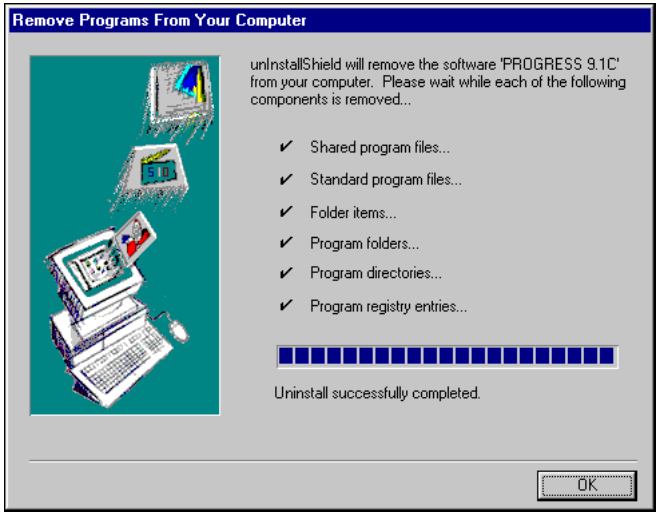
OR

From the desktop, select **Start→Settings→Control Panel→Add/Remove Programs**. From the **Install/Uninstall** tab, select **Progress** and choose **Add/Remove**.

The **Confirm File Deletion** dialog box appears:



8 ♦ Choose **Yes**. The **Remove Programs From Your Computer** dialog box appears:



When the usage count on a shared system file reaches 0, a **Shared File** warning dialog box appears.

9 ♦ Follow the instructions in the **Shared File** warning dialog box to either remove shared files or leave them on your system. Do not remove any of the following Microsoft system files, because their removal might affect other applications that share them: `mcf40.dll`, `mfc42.dll`, `mfcans32.dll`, `msvcrt.dll`, `msvcrt20.dll`, `msvcrt40.dll`, `olepro32.dll`.

10 ♦ Choose **OK** when the deletion process is complete.

NOTE: If an installation terminates abnormally, you must remove the `ProgressUninstall` file before reinstalling. This file records the initial installation and appends additional installations to the file. You can locate the `ProgressUninstall` file in the Windows directory folder; for example, on a Windows NT machine the file `ProgressUninstall9.1D` is located in `drive:\Winnt40` (where `drive` represents the drive on which you installed Progress).

2.16.2 Manually Removing Progress Version 9.1D

This section provides guidelines to follow when manually uninstalling or reinstalling Progress Version 9.1D on Windows NT.

Follow these steps to successfully uninstall or reinstall Progress Version 9.1D:

1 ♦ Log in under the same domain and user name you used when you installed Progress.

2 ♦ Make sure that Progress is not running in an open DOS window (or that the current directory is not in any Progress-related file or directory).

3 ♦ Stop all Progress processes and close all Progress Help files. You can use the Task Manager to ensure that you stop all processes and close all help files.

4 ♦ Using the Progress Explorer, shut down all Progress services (brokers, NameServers, and databases).

5 ♦ Shut down the AdminServer by following these steps:

a) From the desktop, select **Start→Settings→Control Panel→Services**.

b) Highlight the **AdminService for Progress 9.1D**, and choose **Stop**.

c) Choose **Yes** to confirm that you want to stop the service.

d) When the service stops, choose **Startup**. The **Service** dialog box appears.

e) Change the Startup from Automatic to Manual, and choose **OK**. (This step is necessary if you reboot your machine before completing the uninstall so that the AdminServer does not start up automatically.)

NOTE: If these same services will be required for a new installation, be sure to note any configuration settings, agent parameters, etc.

6 ♦ You might need to reboot your system. Read the following two cases to decided whether or not to reboot:

- If you are using the Netscape Web server (`wsnsa.dll`), you are not required to reboot your system. You only have to shut down the Web server and the WebSpeed Transaction server.
- If you are using the WebSpeed Messenger (`wsisa.dll`) with the Microsoft IIS Web server, you must reboot your system. In order to do this on some production machines, you might have to plan the Progress/WebSpeed reinstallation when you can perform a shutdown of the system.

NOTE: If you use a WebSpeed IIS Messenger and do not reboot your system before uninstalling, the uninstall fails.

7 ♦ Remove the `ProgressUninstall 9.1D` file from the Windows directory (for example, `C:\Winnt40\ProgressUninstall 9.1D`).

8 ♦ Run `regedit.exe` (or `regedt32.exe`) to edit the Windows registry by deleting the 9.1D keys under the following:

```
HKEY_CURRENT_USER\SOFTWARE\PSC\product name(s)
HKEY_LOCAL_MACHINE\SOFTWARE\PSC\product name(s)
```

Then remove the 9.1D key under the following:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\
Uninstall
```

9 ♦ If you have installed a Progress product for which the ODBC driver was also installed, run `regedit.exe` (or `regedt32.exe`) to edit the Windows registry as follows:

a) Go to HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI.

b) Remove the following key (and the values it contains):

```
MERANT 3.60 32-BIT PROGRESS SQL92 v9.1D
```

10 ♦ If you have installed a Progress product for which the MMC was also installed, run `regedit.exe` (or `regedt32.exe`) to edit the Windows registry as follows:

a) Go to HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\MMC\SnapIns.

b) Remove the following key (and the values it contains):

{ED06106E-40A4-11d4-81B3-00C04F6B9729}

11 ♦ If you have installed a Progress product for which the Progress Explorer COM Object was also installed, run `regedit.exe` (or `regedt32.exe`) to edit the Windows registry as follows:

a) Go to HKEY_CLASSES_ROOT\CLSID\.

b) Remove the following key (and the values it contains):

{B970C27F-CA68-11D2-AF3A-00C04FB97F7D}

12 ♦ Depending on the products you have installed, the following files might have been registered during the install and should be unregistered:

dzocx32.ocx	pstimer.ocx
dzstat32.ocx	prox.dll
duzocx32.ocx	sstree.ocx
cscomb32.ocx	cihttp.ocx
csspin32.ocx	cslist32.ocx

NOTE: The registered version of some of these files might not be under the Progress directory. You should also check the Windows `system` and `system32` directories for these files.

The following example shows how you can unregister one of these OCX files, which, in the example, is located in the Windows NT `system32` directory:

```
%DLC%\bin\regsvr32.exe /u c:\winnt40\system32\cscomb32.ocx
```

NOTE: If the Progress Uninstall utility does not complete successfully, you must perform all or some of Step 13 through Step 17 before the new installation can take place. For details, see the Progress Software Corporation online Knowledge Base at `http://techweb.progress.com/services/support/kb.htm`.

13 ♦ Delete the Progress program directory, including all of its subdirectories. The default Progress directory is `C:\Program Files\Progress`.

14 ♦ Delete the Progress folder/group from the Windows NT Start menu.

15 ♦ Shut down your Web server and delete the `cgiip.exe` and `wsisa.dll` files from the Web server `cgi-bin/scripts` directory:

NOTE: If you are uninstalling WebSpeed and using the Netscape Web server (`wsnsa.dll`), you must return the `obj.conf` file to its pre-WebSpeed state. If you are upgrading WebSpeed to the same directory, you need not modify the `obj.conf` file. However, if you intend to change the directory location, then you must modify the `obj.conf` file to reflect the correct location.

16 ♦ Depending on the installation options you chose, such as a Web server type, WebSpeed virtual directory, or having static HTML files copied to your Web server document root directory, you might need to take one or more of the following steps:

a) Delete the WebSpeed directory from your Web server document root directory; for example, on MSIIS the default is: `\InetPub\wwroot\webspeed`.

b) Delete any virtual directories defined for WebSpeed in your Web server.

17 ♦ Reboot your machine and install Progress.

3

Sharing a Progress Installation On a Network

This chapter describes how you can use the Shared Network Installation utility to provide multiple client access to a single copy of Progress that is preinstalled on a network-accessible drive (server). The Shared Network Installation utility is available to you provided you performed either a Complete installation (which automatically installs the utility) or a Custom installation (in which you install the NetSetup component) on the server machine.

This chapter also describes the Silent Installation option for the Shared Network Installation utility.

The chapter contains the following sections:

- Networking Overview
- Preinstallation Considerations
- Using the Shared Network Installation Utility
- Uninstalling the Shared Network Installation Utility
- New Silent Installation Option For the Shared Network Installation Utility

NOTE: Citrix MetaFrame does not support a shared network installation.

3.1 Networking Overview

Typically, a network consists of the following:

- Application workstations
- Database server machine
- Network file server machine

You can install the Progress server and client software on each machine where it will run, or you can install one copy of each on the network file server. For more information on networking with Progress, see Chapter 6, “Configuring Progress.”

3.1.1 Application Workstation

An application workstation is a computer on your network that executes the Progress Client or single-user product. This software allows one or more workstations to access the database server machine.

3.1.2 Database Server Machine

The Progress database server machine is a computer on your network that executes the Progress Server software. This software allows the database server machine to manage one or more Progress databases.

3.1.3 Network File Server Machine

The network file server is a computer that manages file sharing and system security, coordinates station-to-station communications, and controls any attached peripherals, such as printers, disk drives, modems, and plotters.

3.2 Preinstallation Considerations

Before you begin a shared network installation, you must:

1. Uninstall Progress from your client machine if you have previously installed a version of Progress.
2. Be sure that Progress Version 9.1D is installed on your server machine and that you have access to this server.
3. Map a network-accessible drive to the server where Progress is installed.
4. Make sure that you have enough disk space available. The complete set of files necessary to run the Shared Network Installation utility uses 3.5MB of disk space. The files are copied to your `\Windows\System32` directory or `\Windows\System` directory. Note that some of these files might already exist in your system directory.

3.3 Using the Shared Network Installation Utility

The Shared Network Installation utility updates each client machine with all system files, icons, and registry entries that are necessary for launching Progress locally. Each client shares the networked copy of Progress.

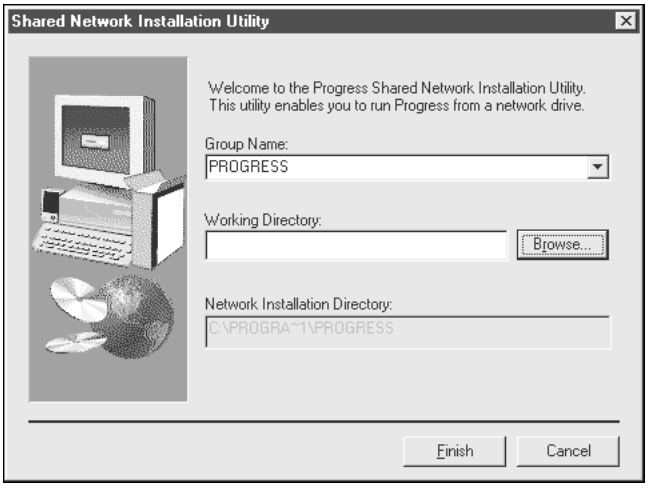
Follow these steps to perform a Progress shared network installation on your client machine:

1 ♦ From the **Start** menu, choose **Run**. The **Run** dialog box appears. Enter this command in the **Run** dialog box:

```
drive:\destination path\netsetup\setup.exe
```

where the *destination path* is the same path where Progress is installed on the server machine.

2 ♦ Choose **OK**. The **Shared Network Installation Utility** dialog box appears:



A network installation directory path appears.

3 ♦ Choose the default group name or type a different name in the **Group Name** field. If the group name does not already exist, the install utility adds the group name to your system's Program menu.

4 ♦ Choose a working directory. The working directory is where Progress places the databases and programs you create.

5 ♦ Choose **Finish**. You can access Progress from the network.

3.4 Uninstalling the Shared Network Installation Utility

You can use an uninstall utility to quickly uninstall a Progress shared network installation. All the products that you previously installed for this Progress version are removed.

Follow these steps to uninstall the Progress shared network installation:

1 ♦ From the desktop, choose **Start→ Settings→Control Panel**.

2 ♦ In the **Control Panel Group**, double-click the **Add/Remove Programs** icon. The **Add/Remove Program Properties** dialog box appears.

3 ♦ Select the **Install/Uninstall** tab. The **Uninstall utility** dialog box appears.

4 ♦ From the list of installed programs, select the **Progress 9 Shared Network Installation**. Choose **Add/Remove**. The **Confirm File Deletion** dialog box appears.

5 ♦ Choose **Yes** to confirm that you want to delete the Progress 9 Shared Network Installation from your client machine. The **Remove Programs From Your Computer** dialog box appears.

6 ♦ Choose **OK** to exit the Uninstall utility.

NOTE: When the usage count on a shared system file reaches 0, a **Shared File warning** dialog box appears. Follow the instructions in the dialog box.

3.5 New Silent Installation Option For the Shared Network Installation Utility

In Progress 9.1D, the Shared Network Installation Utility (NetSetup) supports a new installation option. This option, referred to as a silent installation mode, uses the Installshield SILENTMODE mechanism to enable the Progress installation to occur in response to values supplied by a user-created file rather than by values interactively entered through a keyboard.When the Progress 9.1D installation is initiated using this option, there is no user involvement required to advance through and complete the installation process.

NOTE: The NetSetup utility program is an optional component that allows users to share an existing installation of Progress Core. This program is located in DLC\netsetup folder.

The information in this section describes

- How to run the NetSetup utility in an interactive installation mode to create a user-defined response file
- How to execute the NetSetup to run the silent installation
- A sample response file and sample log file

3.5.1 Creating a User-Defined Response File

Before you can run the NetSetup utility in silent installation mode, you must create the user-response file. This file records the values that the NetSetup utility needs to successfully complete the silent installation process.

To create this file, you must perform an initial interactive installation, providing the required values.

To create the user-defined response file, follow these steps:

1 ♦ Enter the following command on the command line:

```
setup.exe -r
```

`setup.exe`

The command to run the NetSetup program interactively.

`-r parameter`

The Installshield mechanism to create the response file and to identify it as setup.iss. The setup.iss file is a text file that is modifiable. The `-r` parameter is the recommended method to ensure that you create a complete response file.

2 ♦ Press **ENTER**. NetSetup runs interactively.

As you enter values through the keyboard, NetSetup simultaneously creates the response file, setup.iss, to record particular values that you enter. Values specific to your installation are read and stored in the response file.

Figure 3–1 shows the typical contents of a sample response file.

```
[InstallShield Silent]
Version=v5.00.000
File=Response File
[File Transfer]
OverwriteReadOnly=NoToAll
[DlgOrder]
Dlg0=DLG_SHARED_INSTALL-0
Count=2
Dlg1=RebootDialog-0
[DLG_SHARED_INSTALL-0]
ProgramFolder=PROGRESS 9.1D1B Shared Network Installation
WorkingDir=c:\temp files\cortezwrk
[Application]
Name=PROGRESS 9.1D1B Shared Network Installation
Version=9.1D1B
Company=Progress Software Corporation
Lang=0009
[RebootDialog-0]
Result=0
Choice=0
```

Figure 3–1: Contents Of the Sample Response File - Setup.iss

The values entered for the ProgramFolder and WorkingDir during the interactive installation are recorded in the response file. The shortcut information identified in the Program Folder and the user's work files identified in the WorkingDir are read during a silent installation.

3.5.2 Executing NetSetup With the Silent Installation Option

Once the sertup.iss file exists, the installation process using the silent mode can be initiated.

To initiate NetSetup with the silent installation option, enter the following command on the command line to run NetSetup in silence:

```
DLC\netsetup\setup.exe -psclogC:\temp -s
```

`DLC\netsetup`

The path to where the NetSetup utility resides in the Progress Version 9.1D product file structure.

`setup.exe`

The command to run the NetSetup program.

`-psclog`

Required parameter to run NetSetup in silent installation mode. This parameter is also used to identify a path to a log file that contains information about the status of the silent installation.

In this example command entry, the new log file will be created in the temp directory on the C: drive as indicated in the sample command entry.

The log file created by the installation program is called PscNetSetupMsg.log.

`-s`

Required parameter to run an installation without requiring user interaction. This parameter is executed with the setup.exe to run a silent installation.

Figure 3–2 shows the typical contents of the PscNetupMsg.log file.

```
[Progress NetSetup Messages]
Type=SEVERE
Date=12-4-2001
Time=14:40:48
File=Setup.Rul
Line=1541
Message=A path to the log file must be supplied using the -psclog
parameter.
**********
Type=INFORMATION
Date=12-4-2001
Time=14:52:40
File=Setup.Rul
Line=364
Message=Completed Successfully.
```

Figure 3–2: Contents of the Sample Log File - PscNetSetupMsg.log

Details About the PscNetSetupMsg.log File

Review the following points concerning the log file PscNetSetupMsg.log:

- You must supply the absolute path to the log file because the installation program is shared by other users. Without this explicit path information, the standard Installshield log file might be incorrectly written to by more than one instance of the setup.exe file which would corrupt the file.
- The PscNetSetupMsg.log file automatically logs all error messages and all successful installation messages as reported by the installation code.

4

Administration Utilities

This chapter provides step-by-step instructions for running the Product Update, SHOWCFG, ProControl, and Progress Explorer utilities. You can use the Product Update utility to change the number of users or the expiration date for a Progress product and then display product license information. You can use the SHOWCFG utility to display license information and to manage user licenses.

Specifically, this chapter contains the following sections:

- Using the Product Update Utility
- Displaying License Information Using the SHOWCFG Utility
- Managing User Licenses
- Using the Progress Explorer Tool
- Using the ProControl Utility
- Using Progress Resources

4.1 Using the Product Update Utility

Use the Progress Update utility to change the number of users or the expiration date for a Progress product. Contact your Progress Software Sales Representative for a new *License Addendum* if you need to use this utility.

Follow these steps to use the Product Update utility to update your license:

1 ♦ To review your licensing information, use the Show Configuration utility (SHOWCFG) to display the product license information for each Progress product installed on your system. See the "Displaying License Information Using the SHOWCFG Utility" section for instructions.

2 ♦ Choose the **License Update** icon from your Progress group. After a welcome message appears, the **Serial & Control Numbers** dialog box appears.

3 ♦ Type the new serial number and control numbers Progress Software Corporation supplies when you upgrade your license.

4 ♦ Choose **Accept**. The **Product(s) Updated** dialog box displays the products you want to update.

5 ♦ When you are finished updating the products, choose **Done**.

4.2 Displaying License Information Using the SHOWCFG Utility

After you install Progress, the installation program prompts you to enter product information contained in the *License Addendum*, which is included in your software package. The installation program records the license information in the Progress configuration file (`progress.cfg`). You can use the SHOWCFG utility to display this licensing information.

4.2.1 Using the SHOWCFG Utility

The SHOWCFG utility displays product installation and configuration information for each Progress product installed on your system. To run SHOWCFG from the Start menu, choose **Progress→Config**; to run SHOWCFG from the Progress Program Group, double-click the Config icon. The Config icon looks like this:



You can also start the SHOWCFG utility by typing the following command at the command line of the PROENV window:

```
showcfg
```

The SHOWCFG utility opens the **Product Configuration Information** dialog box to display the product configuration information stored in the Progress configuration file `progress.cfg`, which is created and modified during product installation. Figure 4–1 shows a typical display of the **Product Configuration Information** dialog box.

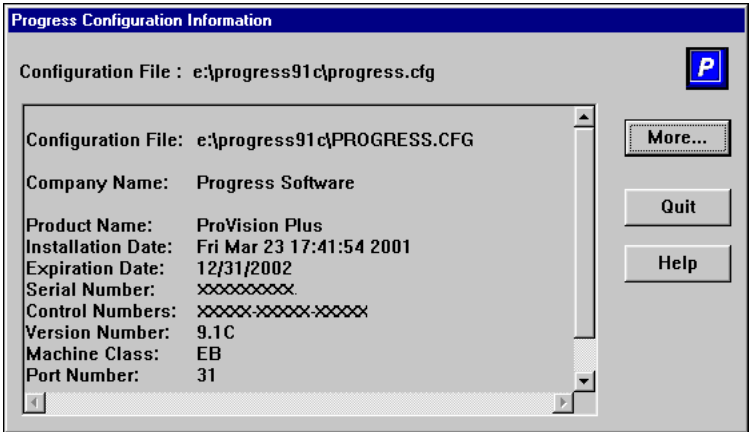


Figure 4–1: Product Configuration Information From SHOWCFG Utility

The display contains the following information for each Progress product installed on the system:

Product Name — Name of the installed product.

Installation Date — Date the product was installed.

Expiration Date — Date the license expires.

Serial Number — Number associated with the license agreement.

Control Numbers — Numbers used by the Progress installation software.

Version Number — Software product version.

Machine Class — Tier code associated with the license agreement.

Port Number — Platform on which the software product is installed.

SHOWCFG searches for the `progress.cfg` file in the locations defined as PROCFG and DLC in the `progress.ini` file. To find the `progress.ini` file, SHOWCFG searches the following locations, in the order specified:

1. The current working directory.
2. The special users directory (set using the Properties option).
3. The Windows directory.

If the utility finds the `progress.cfg` file, it displays the contents in the **Progress Configuration Information** dialog box.

If SHOWCFG cannot find the `progress.cfg` file, the **Open** dialog box appears so you can specify the file's location. You can also use the **Open** dialog box to specify a different Progress configuration file to display. To display the **Open** dialog box, choose the **More** button in the **Progress Configuration Information** dialog box.

Progress does not accept optional qualifiers that point to a `.cfg` file other than `DLC:PROGRESS.CFG`.

4.2.2 Displaying License Information

Follow these steps to display your current product license information, the number of users, and the platforms for each Progress product installed on your system:

1 ♦ From the desktop, choose **Start→Run**. The **Run** dialog box appears.

2 ♦ Type **SHOWCFG** in the **Open** field, and choose **OK**.

OR

At the command line, type the following command and press **ENTER**:

```
showcfg %DLC%\progress.cfg
```

The Progress Configuration Information dialog box appears and displays the information you entered.

The "Managing User Licenses" section contains more information about licensing.

4.3 Managing User Licenses

The Progress license you purchase determines how many concurrent users are allowed to run Progress. You are responsible for making sure users comply with your license agreement. Progress provides reporting capabilities to help you do this.

This section describes:

- The Progress license information that is shipped with your Progress product.
- How to read the Progress usage log file.
- How to produce a report of current licensed user connections.

4.3.1 Progress License Information

The *License Addendum* that is shipped with your Progress product provides specific information about the product license you purchased, including:

- A serial number
- A control number
- The maximum number of concurrent Progress users allowed by the license

When you install Progress, the installation procedure prompts you to enter product information from the *License Addendum*. The installation procedure records the license information in the Progress configuration file `progress.cfg`. Use the SHOWCFG utility to display the product license information for each Progress product installed on your system.

For more information on the SHOWCFG utility, see the "Using the SHOWCFG Utility" section in this chapter.

4.3.2 Using the Progress License File

Progress creates a license file that records license-related information about Progress database users. If the log file does not already exist, the broker creates it and places it in the same directory as the database (.db) file. The broker creates the file in the format *databasename*.lic, where *databasename* is the name of the database to which the user connects.

NOTE: If Progress encounters an error while trying to open or write to the license file, the error is recorded in the database .lg file and no more entries are written to the license (.lic) file.

Reading the License File

Use a text editor to display the license file contents. The contents appear in the following order:

1. Current date
2. Current time
3. Number of licensed users specified by the configuration file
4. Current number of total connections
5. Maximum number of total connections
6. Minimum number of total connections
7. Current number of interactive connections
8. Maximum number of interactive connections for the past hour
9. Minimum number of interactive connections for the past hour
10. Current number of batch connections
11. Maximum number of batch connections for the past hour
12. Minimum number of batch connections for the past hour

For example, the following sample file entry illustrates the log format:

```
4/26/01 9:00 25 18 23 11 17 20 11 1 5 0
```

When Progress writes to the license file, the maximum and minimum values are reset for the next hour.

Maintaining the License File

The database or system administrator should consider archiving license files periodically. In one year, a license file accumulates 8,760 entries. These entries occupy about 440,000 bytes of disk space.

Since the license file must be closed before the administrator archives it, the administrator must first shut down the database. At that point, the license file can be either archived immediately or renamed and archived later.

Creating a Usage Report

To produce a report of license-related information about current Progress database users, run the `licrpt.p` procedure file. The report generator input screen appears:

```
Enter Date Range:      To:        Enter Start Time (hours 0 to 23):
Enter Stop Time (hours 0 to 24):
Enter time division (in hours, or 0 for complete range):
Database Name:
```

This is a sample output from the `licrpt.p` procedure file:

```
                        Database Connection Counts
-------------------------------------------------------------------------
| Date   Period  LcnUsers MaxTot Excptns MinTot AveTot MaxBat MinBat AvBat
|
|------- ------- -------- ------ ------- ------ ------ ------ ------ -----
|
| 4/11/01 8-17      100     20       0      0    10.      0      0     0.
|
| 4/13/01 8-17      100     20       0     18    19.      0      0     0.
|
| 4/16/01 8-17      100     23       0     17    20.      0      0     0.
|
| 4/20/01 8-17      100     33       0     17    25.      0      0     0.
|
| 4/24/01 8-17      100     32       0     26    29.      0      0     0.
|
| 4/26/01 8-17      100     26       0     17    22.      0      0     0.
```

4.4 Using the Progress Explorer Tool

Progress Explorer is a graphical administration tool that runs on Windows platforms.

Using the Progress Explorer tool, you can:

- Create new instances of Progress servers and configure their property settings
- Modify property settings of existing Progress server instances
- Start and stop Progress servers
- Monitor the status of Progress servers

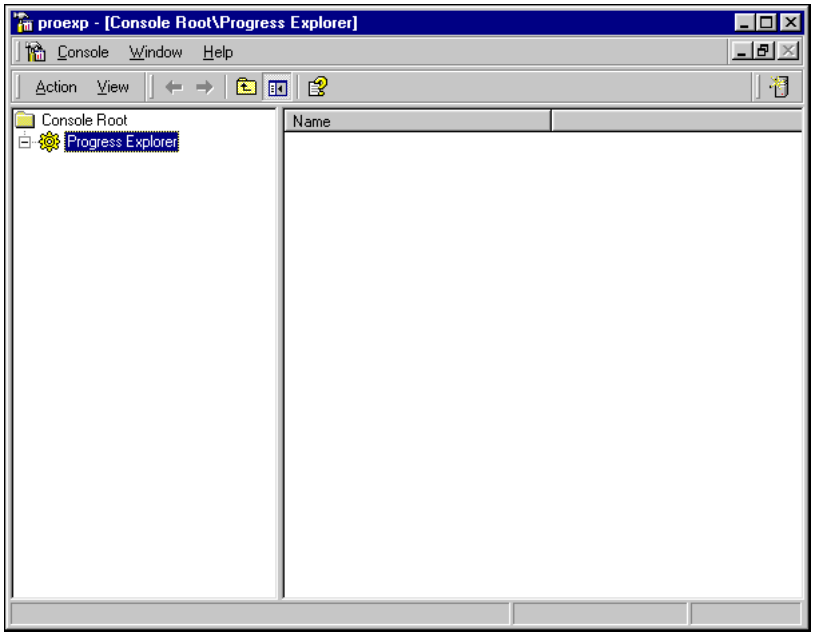
To use the Progress Explorer configuration tools, you must first connect to the AdminServer on the same host where the Progress products you want to start, stop, or query are installed. When you establish a connection to the AdminServer, Progress Explorer presents a tree view of all the Progress products to which the AdminServer grants you administrative access. For complete instructions on using Progress Explorer, see the Progress Explorer online help.

Follow these steps to launch the Progress Explorer:

1 ♦ From the **Start** menu choose **Progress→ Progress Explorer Tool**, or from the Progress Program Group, double-click the Progress Explorer Tool icon. The Progress Explorer Tool icon looks like this:



2 ♦ When you start Progress Explorer, the main window appears:



3 ♦ Follow the instructions in the Progress Explorer online help to establish a connection to an AdminServer.

NOTES:

- Database configurations you create with the Progress Explorer Database Configuration Tool are saved in the `conmgr.properties` file. Do not edit the `conmgr.properties` file directly.
- Configurations you create with any other Progress Explorer configuration tool are saved in the `ubroker.properties` file. You can edit the `ubroker.properties` directly with any text editor. However, you must be certain that you are not running Progress Explorer simultaneously. When you want to edit the configuration of a NameServer, AppServer, AppServer Internet Adapter, SonicMQ Adapter, WebSpeed Transaction Server, or WebSpeed Messenger, use only one configuration tool at a time.

4.5 Using the ProControl Utility

The ProControl utility is a tool with a graphical user interface and is supplied with the Progress NT Server. The Progress NT Server permits Progress processes to run as registered NT services.

ProControl lets you perform the following tasks on a local or remote NT host:

- Start and stop ProService
- Start and stop Progress databases
- Start and stop Progress DataServer processes
- Set Progress environment variables
- Start Progress batch processes

NOTE: Using ProControl to perform these tasks ensures that Progress processes run as registered NT services. All of these tasks result in changes to the registry.

Follow these steps to access the ProControl Utility and start ProService:

1 ♦ Access the ProControl utility by double-clicking the ProControl icon in the Control Panel, or by double-clicking the ProControl icon in the Progress program group. The ProControl icon looks like this:



Alternatively, you can access ProControl by typing the following command at the command line:

```
proctrl
```

2 ♦ Follow the instructions in the "Starting ProService" section of Appendix B, "The ProControl Utility," to start ProService.

4.6 Using Progress Resources

Progress uses several operating system resources such as shared memory and memory locks, processes, and client memory. You can plan Progress operations more effectively if you understand these resources.

4.6.1 Shared Memory

Shared memory is an area in system memory that multiple users can access concurrently. Progress stores shared resources in the shared-memory area; more than one user and server can access each database. Progress uses semaphores and spin locks to synchronize the activities of server and self-service client processes that are connected to a database. Each process uses its semaphore or relies upon the spin lock when it must wait for a shared resource.

You can tune Progress performance by reconfiguring the size of the following shared-memory buffers:

- **Database buffers** — Progress reads database blocks into the database buffer pool. Larger buffers usually result in less disk I/O.
- **Before-image (BI) buffers** — Progress stores BI notes in memory before writing them to disk.
- **After-image (AI) buffers** — Progress stores AI notes in memory before writing them to disk.

Progress also creates shared-memory tables to provide essential information on the status of each process, server, transaction, and lock. These tables enable you to control all of the database activities from one shared area.

See the *Progress Database Administration Guide and Reference* for more information about managing and improving Progress performance.

4.6.2 Processes

Progress provides the following optional processes to improve performance:

- **Asynchronous Page Writer (APW)** — The APW improves database performance by performing overhead operations in the background. These operations provide available buffers, reduce the number of buffers that Progress reads before writing to disk, and reduce the overhead associated with before-image checkpointing (the process of synchronizing the buffer pool of modified blocks to the database).
- **Before-image Writer (BIW)** — The BIW improves performance by continually writing before-image buffers to disk. These writes occur in the background.
- **After-image Writer (AIW)** — The AIW improves performance by continually writing AI buffers to disk soon after Progress fills the buffers.
- Progress **Watchdog (PROWDOG)** — The watchdog process cleans up after improperly terminated processes by releasing locks, backing out any live transactions and releasing shared-memory locks, and disconnecting and cleaning up the server's remote clients.

See the *Progress Database Administration Guide and Reference* for more information about managing and improving Progress performance.

PART II

Configuration

Setting Up Your Environment

Configuring Progress

Configuring Progress Unified Broker Products

Starting and Running Progress

Progress Event Logging

5

Setting Up Your Environment

This chapter describes additional steps you must perform to run Progress Version 9.1. This chapter also provides steps for maintaining Progress Version 8 on your system.

Specifically, this chapter includes the following sections:

- Setting System Environment Variables
- Setting Java Environment Variables
- Setting Progress Program Item Properties
- The Windows Registry and the progress.ini File
- The Progress Working Directory
- Using the Proenv Utility
- Managing the AdminServer
- Creating a Version 9 Database
- Creating and Configuring a Version 9 Database Server
- Running Progress
- Maintaining Multiple Versions Of Progress

5.1 Setting System Environment Variables

By default, the Progress installation program tailors all the necessary Progress and Java environment variables to the directories where they are installed. For example, the installation automatically sets the %DLC% environment variable to your Progress installation path.

NOTE: The %DLC% environment variable is not set at the system level. After installing Progress, however, you can set environment variables to suit your own preferences. You can use Proenv to set the %DLC% environment variable to the directory where Progress is installed.

CAUTION: Although editing environment variables is an option, this procedure is not recommended if more than one version of a Progress product exists on the same system.

For more information on environment variables, see the chapter on maintaining user environments in the *Progress Client Deployment Guide*, or see your specific product documentation.

NOTE: Before you continue, consult the *Progress Release Notes*. These notes contain the latest information about the current release that the Progress documentation set might not include. Progress Software Corporation ships release notes in Microsoft Write (readme.wri) format. Click the Release Notes icon in your Progress program group, or access Readme.pro with any text editor.

5.2 Setting Java Environment Variables

Progress Version 9 bundles the Java Runtime Environment (JRE) component and the Java Development Kit (JDK) component with certain products that you install. For more information, see the "System Requirements" section in Chapter 1, "Installation Requirements."

5.2.1 JDKHOME

Java is used by some products, such as WebSpeed, the AppServer, and SQL-92, for product functionality. After you install any of these products, you should verify that the JDKHOME value is set correctly in the registry. The value must be set to the directory where the JDK included in the Progress installation resides (for example, C:\Program Files\PROGRESS\jdk).

You can verify the JDKHOME value in the following location in the registry:

```
HKEY_LOCAL_MACHINE\SOFTWARE\PSC\PROGRESS\version\JAVA
```

5.3 Setting Progress Program Item Properties

Although the Progress Installation Utility creates a Progress Group with Program Items, you must set item properties such as startup parameters and buffer pools.

For information on Progress startup parameters, buffer pools, and related topics, see the Progress DataServer Guides (*Progress DataServer for ODBC Guide*, *Progress DataServer for ORACLE Guide*, and *Progress/400 Product Guide*) and the *Progress Client Deployment Guide*.

For example, to change the properties for a Progress Program Item in Windows NT and Windows 2000, highlight the item and choose **File→Properties** in Windows Explorer. See the appropriate Windows documentation for more information about setting Program Item properties.

5.4 The Windows Registry and the progress.ini File

Applications running on Windows NT, Windows 98, Windows 2000, Windows Me, Windows XP, or Citrix MetaFrame rely on the registry for startup information, such as color, font, and key bindings. Progress keeps this information in the initialization file, `progress.ini`, located in your `Progress\bin` directory. During the tailoring phase of the installation, the Installation Utility creates registry entries from the `progress.ini` data.

The information from the `progress.ini` file resides under the `HKEY_CURRENT_USER\SOFTWARE\PSC\PROGRESS\`*`version`* and the `HEKY_LOCAL_MACHINE\SOFTWARE\PSC\PROGRESS\`*`version`* registry keys.

NOTE: If you modify the `progress.ini` information, you must run the `ini2reg` utility. This utility updates the information in the registry.

See the *Progress Client Deployment Guide* for more information on the `progress.ini` file and the `ini2reg` utility.

5.4.1 Environment Variables

Progress supports some environment variables for graphical user interface (GUI) clients in the [Startup] section of the `progress.ini` file. Progress supports environment variables for character clients, such as the AppServer and WebSpeed Agents, in the [WinChar Startup] section of the `progress.ini` file.

Table 5–1 lists the supported environment variables.

Table 5–1: Supported Environment Variables *(1 of 5)*

Variable	progress.ini File Section	Description	Example
DLC[1]	[Startup] [WinChar Startup]	The relative pathname of the directory where you installed your Progress system software. By default, the installation utility sets this variable.	`set DLC=C:\Program Files\Progress`
EVTLEVEL	–	NT and 2000 only. Specifies the level of information that Progress writes to the NT or 2000 Application Event Log. You can specify one of the following cases: **None:** No Progress events are written to the Event Log. **Brief:** Progress Error and Warning messages are written to the Event Log. **Normal:** Progress Error and Warning messages are written to the Event Log along with any Progress message that are normally written to the log file (`.lg`). This is the default. **Full:** Progress Error, Warning, and Informational messages are written to the Event Log along with any messages generated by the Message Statement.	`set EVTLEVEL = NORMAL`

Table 5–1: Supported Environment Variables *(2 of 5)*

Variable	progress.ini File Section	Description	Example
PATH	–	A list of directory paths separated by semicolons. When you run a program or batch file, the system searches for it in the current directory. Then it searches in the directory paths defined in PATH in the order they are mentioned. Your PATH should include any directory pathname that contains a program or batch file you want to run. Also, each directory pathname should include the drive letter of the disk that contains the directory. PATH is a system environment variable, not a Progress environment variable. Set it in a manner appropriate for the operating system instead of in the registry or in progress.ini. Different Progress products require different PATH settings. To set up PATH for your Progress product, follow the instructions provided in Appendix C, "The Progress Directory Structure."	`set PATH=%PATH%; %DLC%\BIN;%DLC%`

Table 5–1: Supported Environment Variables *(3 of 5)*

Variable	progress.ini File Section	Description	Example
PROBUILD	[Startup]	The pathname of the directory that contains items referenced in link scripts produced by the PROBUILD utility. By default, the installation utility sets this variable.	`set PROBUILD= C:\Program Files\ Progress\PROBUILD`
PROCFG	[Startup] [WinChar Startup]	The filename or full pathname of your product's configuration file. The configuration file is a data file that identifies the Progress products and components that you are licensed to use. Reset PROCFG if you have moved your configuration file from the directory where you installed Progress.	`set PROCFG=%DLC%\ PROGRESS.CFG`
PROCONV	–	The filename (or full pathname) of the Progress `convmap.cp` file. The `convmap.cp` file is a binary file that contains all of the conversion tables that are available to Progress. See the *Progress Internationalization Guide* for more information on the `convmap.cp` file.	`set PROCONV=%DLC%\ CONVMAP.CP`

Table 5–1: Supported Environment Variables *(4 of 5)*

Variable	progress.ini File Section	Description	Example
PROMSGS	[Startup] [WinChar Setup]	The full pathname of your Progress error messages file. The default value is %DLC%\promsgs. Set the PROMSGS environment variable only if you want to use an error messages file different from the default PROMSGS file in the %DLC% directory.	set PROMSGS= C:\Program Files\ Progress\PROLANG\ GER\PROMSGS.GER
PROPATH	[Startup] [WinChar Setup]	A list of directory paths separated by commas. By default, the installation utility sets this variable.	set PROPATH=., C:\Program Files\ Progress
PROSTARTUP	–	The pathname of the Progress default startup parameter file, startup.pf. This file is read by all Progress modules at startup; it must exist for Progress to execute properly.	set PROSTARTUP= C:%DLC%\STARTUP.P F
JDKHOME	–	Establishes the top-level directory for the Java Developer's Kit (JDK).	set JDKHOME=%DLC%\jdk
JREHOME	–	Establishes the top-level directory for the Java Runtime Environment (JRE).	set JREHOME=%DLC%\jre
JFCHOME	–	Establishes the top-level directory for the Java Foundation Classes (JFC).	set JFCHOME=%DLC%\jfc

Table 5–1: Supported Environment Variables *(5 of 5)*

Variable	progress.ini File Section	Description	Example
JDKCP	–	Sets the classpath for class.zip; Java Developer's Kit (JDK) only.	set JDKCP= *%variable-name%*/ lib/class.zip
JRECP	–	Sets the classpath for Java Runtime Environment (JRE); if no JRE, then it sets classpath for JDK.	set JRECP= *%variable-name%*/ lib/rt.jar
JFCCP	–	Sets the classpath for Java Foundation Classes (JFC) only.	set JFCCP= *%variable-name%*/ swingall.jar
PROGRESSCP	–	Contains a list of paths, jar files, and zip files for running Progress-specific products.	set PROGRESSCP= *%variable-name%*/ java/progress.zip
CLASSPATH	–	Progress correctly sets the appropriate classpath variable based on the platform in use.	set CLASSPATH= $JDKCP;$JFCCP; $PROGRESSCP
JIT	–	Sets the just-in-time compiler correctly.	set JIT="-nojit"
JVMEXE	–	Sets the Java virtual machine to run correctly.	set JVMEXE=jre

[1] Note that the DLC variable is set in the various command scripts and in the registry; the variable is not set at the system level.

5.5 The Progress Working Directory

A working directory is a directory that contains your Progress database and application files. The Installation Utility prompted you to create a working directory from which to run Progress.

CAUTION: Never run Progress products from the directory in which you installed them. If you do, you could damage the Progress software files.

5.6 Using the Proenv Utility

The %DLC% environment variable is not set at the system level. The Proenv utility can automatically set the %DLC% environment variable to the directory where Progress is installed. It then adds `%DLC%/bin` to your PATH.

To start the Proenv utility from the desktop, choose **Start→Programs→Progress** (or the actual directory where you installed Progress)→ **Proenv**. You can also start it from the command line. Proenv opens a DOS window, sets the environment variables, and then changes the current directory to the working directory you set when you installed Progress.

5.7 Managing the AdminServer

An AdminServer is installed on every system where you install a Progress database, DataServer, AppServer, SonicMQ Adapter, NameServer, or WebSpeed Transaction Server. On Windows NT-based systems, the AdminServer starts automatically and runs as an NT service. The AdminServer must be running in order to use any of the Progress Explorer configuration tools, or any of the following command-line configuration utilities:

- ADAPTMAN and ADAPTCONFIG SonicMQ Adapter utilities
- AIACONFIG AppServer Internet Adapter utility

  NOTE: There is no AIAMAN command, because the Web server's Java servlet engine starts it. You can control the AppServer Internet Adapter through the Progress Explorer. See the chapter on the AppServer Internet Adapter in *Progress on the Web* for more information.

- ASBMAN and ASCONFIG AppServer utilities
- DBMAN database utility
- MSSCONFIG and MSSMAN SQL Server DataServer utilities

- NSCONFIG and NSMAN NameServer utilities
- ODBCONFIG and ODBMAN ODBC DataServer utilities
- ORACONFIG and ORAMAN DataServer utilities
- WSCONFIG and WTBMAN WebSpeed utilities

5.7.1 Starting the AdminServer

To start the AdminServer on your machine, do one of the following from the desktop:

- On Windows NT, Windows 98, and Citrix MetaFrame platforms, choose **Start→Settings→Control Panel→Services**. Highlight the **AdminService for Progress 9.1D**, and choose **Start**.
- On Windows 2000 and XP, choose **Start→ Programs→ Administrative Tools→ Services**. Select the **AdminService for PROGRESS 9.1D**, and double-click. The **AdminService for PROGRESS 9.1D Properties** dialog box appears. Choose **Start**, then choose **OK**.

  NOTE: If Administrative Tools is not available, right-click from the Task Bar. Choose **Properties**, then select the **Advanced** tab. Select the **Display Administrative Tools** checkbox, then choose **OK**.

NOTE: For most product installations, the AdminServer is set to Autostart.

5.7.2 Stopping the AdminServer

To stop the AdminServer on your machine, do one of the following from the desktop:

- On Windows NT, Windows 98, and Citrix MetaFrame platforms, choose **Start→Settings→Control Panel→Services**. Highlight the **AdminService for Progress 9.1D**, and choose **Stop**. Choose **Yes** to confirm that you want to stop the service and allow 30-60 seconds for the process to complete.
- On Windows 2000 and Windows XP, choose **Start→ Programs→ Administrative Tools→ Services**. Select the **AdminService for PROGRESS 9.1D**, and double-click. The **AdminService for PROGRESS 9.1D Properties** dialog box appears. Choose **Stop**, then choose **OK**.

  NOTE: If Administrative Tools is not available, right-click from the Task Bar. Choose **Properties**, then select the **Advanced** tab. Select the **Display Administrative Tools** checkbox, then choose **OK**.

5.7.3 Changing the Default Port

On Windows NT, the AdminServer runs as a service. The AdminServer is configured to start automatically. You can change the listening port for the AdminServer by adding -port *port-number* to the following registry keys:

```
HKEY_LOCAL_MACHINE\SOFTWARE\PSC\AdminService\version\StartupCmd
HKEY_LOCAL_MACHINE\SOFTWARE\PSC\AdminService\version\ShutdownCmd
```

To change the default port, add the port (-port), a space, and the port number to the end of the value, as follows:

```
... com.progress.chimera.adminserver.AdminServer -port 9998
```

5.7.4 Changing the Default Adminport

The adminport specifies the port that will be used for communication between the AdminServer and the services that it starts. You can change the adminport for the AdminServer by adding -adminport *adminport-number* to the following registry keys:

```
HKEY_LOCAL_MACHINE\SOFTWARE\PSC\AdminService\version\StartupCmd

HKEY_LOCAL_MACHINE\SOFTWARE\PSC\AdminService\version\ShutdownCmd
```

To change the default adminport, add the adminport (-adminport), a space, and the adminport number to the end of the value, as follows:

```
... com.progress.chimera.adminserver.AdminServer -adminport 9999
```

5.7.5 Changing the Startup Setting

For most product installations, the AdminServer is set to Autostart. You can change this setting to Manual by doing one of the following from the desktop:

- On Windows NT, Windows 98, and Citrix MetaFrame platforms, choose **Start→Settings→Control Panel→Services**. Highlight the **AdminService for Progress 9.1D**, and choose **Startup**. The **Service** dialog box appears, allowing you to change the **Startup Type** setting.
- On Windows 2000 and Windows XP, choose **Start→ Programs→ Administrative Tools→ Services**. Select the **AdminService for PROGRESS 9.1D**, and double-click. The **AdminService for PROGRESS 9.1D Properties** dialog box appears. Choose **Manual** in the **Startup type** field, then choose **OK**.

  NOTE: If Administrative Tools is not available, right-click from the Task Bar. Choose **Properties**, then select the **Advanced** tab. Select the **Display Administrative Tools** checkbox, then choose **OK**.

5.7.6 Running Multiple AdminServers

If you are running multiple AdminServers or two versions of the Progress Explorer, you need to override both the default port and the default adminport settings. You can change these settings using the commands detailed in the "Changing the Default Port" and the "Changing the Default Adminport" sections.

5.7.7 Querying the AdminServer

On Windows, you can use the PROADSV utility with the `-query` and `-port` parameters:

SYNTAX

```
proadsv -query -port 9998
```

`-query`

Displays the AdminServer status.

`-port`

Specifies the listening port for the AdminServer. This is needed if you specified a port other than the default port when you started the AdminServer.

5.8 Creating a Version 9 Database

To use Version 9 databases, you must do one of the following:

- Create a new Progress database in Version 9 by using the PRODB command, the Data Administration Tool, or the Data Dictionary.
- Convert an existing Version 8 database to Version 9.

For more information on creating databases, see the *Progress Database Administration Guide and Reference*.

5.9 Creating and Configuring a Version 9 Database Server

Follow these steps to use Version 9 features with a server:

1 ♦ From the desktop, do one of the following:

  – On Windows NT, Windows 98, and Citrix MetaFrame platforms, choose **Start→Settings→Control Panel→Services**, and verify that the AdminServer is running.

  – On Windows 2000 and Windows XP, choose **Start→ Programs→ Administrative Tools→ Services**. Verify that the status for the AdminService for PROGRESS 9.1D is Started.

  NOTE: If Administrative Tools is not available, right-click from the Task Bar. Choose **Properties**, then select the **Advanced** tab. Select the **Display Administrative Tools** checkbox, then choose **OK**.

2 ♦ Use the Progress Explorer to add a database configuration.

  For more information on the AdminServer and the Progress Explorer, see the "Managing the AdminServer" section and the Progress Explorer online help.

5.10 Running Progress

Select an icon from the Progress Program Group to begin running your Version 9 applications. Note that WebSpeed products might need additional setup requirements.

CAUTION: Never run Progress products from the directories in which you installed them. Doing so could result in changes to the software that affect its proper operation.

For complete information about starting Progress or WebSpeed products, see either the *Progress Application Development Environment — Getting Started* guide or the *Getting Started with WebSpeed* guide.

NOTE: Before you start a WebSpeed or an AppServer application, you must start the AdminServer. The AdminServer must be running before you can use the Progress Explorer to configure and manage your applications. For details, see the "Managing the AdminServer" section.

5.11 Maintaining Multiple Versions Of Progress

To maintain more than one Progress version on your system, install Version 9.1D as described in Chapter 2, "Installing Progress,"with the following exceptions:

- When you are prompted for a Destination Directory, make sure the directory you specify is not the same as for other installed versions. Type the pathname of a separate directory in which to install Version 9.1D.
- Redefine your PATH. For more information on modifying your directory paths, see the "Environment Variables" section.
- To run AdminServers for more than one version of Progress, you must set both `-port` and `-adminport` as described in the "Managing the AdminServer" section.

NOTE: To access previous version tools or utilities, you must use complete pathnames.

6

Configuring Progress

This chapter provides information about shared-memory configurations, multi-tier configurations, client/server configurations, and preparing to run Progress on a TCP/IP network.

Specifically, this chapter contains the following sections:

- The Progress Explorer Framework
- Shared-memory Configurations
- Multi-tier Configurations
- Client/Server Configurations
- Client/Server and Progress AppServer In the Network Environment
- Preparing To Run Progress In a TCP/IP Network

6.1 The Progress Explorer Framework

The Progress Explorer Framework is a system administration framework that provides a consistent interface for managing all Progress products installed on your network. The framework consists of the following elements:

- **AdminServer** — Provides secure administrative access to Progress products
- **Progress Explorer Configuration Tool** — A configuration tool with graphical user interfaces for local and remote system administration
- **Command-line configuration utilities** — Character versions of the Progress Explorer configuration tools

6.1.1 AdminServer

An AdminServer is installed on every system where you install a Progress database, DataServer, AppServer, SonicMQ Adapter, NameServer, or WebSpeed Transaction Server. The AdminServer must be running in order to use the Progress Explorer configuration tools or the command-line configuration utilities to manage your Progress installation. The AdminServer grants access to each instance of an installed Progress product.

For the complete syntax of the PROADSV utility, see the "Querying the AdminServer" section in Chapter 5, "Setting Up Your Environment." For more information about the AdminServer, see the "Managing the AdminServer" section in Chapter 5, "Setting Up Your Environment."

6.1.2 Progress Explorer

Progress Explorer is a graphical user interface that provides an easy way for you to manage Progress servers. Progress Explorer is one of the primary components of the Progress administration framework.

The Progress Explorer tool runs as a Windows client and works with another administration component, the Progress AdminServer, to provide a client/server framework for managing the following:

- AppServer
- AppServer Internet Adapter
- Database
- DataServer for MicroSoft SQL Server
- DataServer for ODBC

- DataServer for Oracle
- NameServer
- SonicMQ Adapter
- WebSpeed Transaction Server
- WebSpeed Messenger

Using the Progress Explorer tool, you can:

- Create new instances of Progress products and configure their property settings
- Modify property settings of existing Progress product instances
- Start and stop Progress products
- Monitor the status of Progress products

For more information about working with the Progress Explorer tool, see the Progress Explorer online help. Also see the "Using the Progress Explorer Tool" section in Chapter 4, "Administration Utilities," for information on launching the Progress Explorer.

6.1.3 Command-line Configuration Utilities

The command-line configuration utilities allow you to start, stop, and configure installed Progress products. The command-line configuration utilities that are part of the Progress Explorer Framework include:

- **ADAPTMAN** — Helps manage the operation of a configured SonicMQ Adapter
- **ADAPTCONFIG** — Helps configure an existing SonicMQ Adapter
- **AIACONFIG** — Helps configure an existing AppServer Internet Adapter

  NOTE: There is no AIAMAN command because the Web server's Java servlet engine starts it. You can control the AppServer Internet Adapter through the Progress Explorer. See the chapter on the AppServer Internet Adapter in *Progress on the Web* for more information.

- **ASBMAN** — Helps manage the operation of a configured AppServer
- **ASCONFIG** — Helps configure an existing AppServer

- **DBMAN** — Starts, stops, and queries the current configuration of a Progress database
- **MSSMAN** — Helps manage the operation of a configured DataServer for Microsoft SQL Server
- **MSSCONFIG** — Helps configure an existing DataServer for Microsoft SQL Server
- **NSMAN** — Helps manage the operation of a configured NameServer
- **NSCONFIG** — Helps configure an existing NameServer
- **ODBMAN** — Helps manage the operation of a configured DataServer for ODBC
- **ODBCONFIG** — Helps configure an existing DataServer for ODBC
- **ORAMAN** — Helps manage the operation of a configured DataServer for ORACLE
- **ORACONFIG** — Helps configure an existing DataServer for ORACLE
- **PROADSV** — Starts, stops, or queries the AdminServer
- **WTBMAN** — Helps manage the operation of a configured WebSpeed Transaction Server
- **WSCONFIG** — Helps configure an existing WebSpeed Transaction Server or Messenger

For the complete syntax of each utility, see *Building Distributed Applications Using the Progress AppServer*, the Progress DataServer Guides, *Progress on the Web*, and the *Progress Database Administration Guide and Reference*.

For more information on starting and managing the NameServer using the Progress Explorer and NSMAN utility, see the "Starting Progress Unified Brokers" section in Chapter 7, "Configuring Progress Unified Broker Products."

6.1.4 Managing Progress Product Configurations

The `ubroker.properties` file stores all the configuration definitions for each instance of the following Progress products:

- NameServer
- AppServer
- AppServer Internet Adapter

- DataServer
- SonicMQ Adapter
- WebSpeed Transaction Server

Each configuration definition contains environment variables, registry entries, and property settings for each product instance. Progress Explorer and the associated command-line configuration utilities use this file to store, validate, and manage the configurations for the products.

The file consists of a hierarchical structure of configuration entities, where parent entities provide configuration information that you can override or extend in each child entity. Each configuration entity has a name that begins the entity definition, and the definition contains configuration settings for one or more product instances. For example, the AppServer configurations in `ubroker.properties` can include:

Configuration Entity Name	Configuration Entity Function
`[UBroker]`	Defines default property settings for all NameServer, AppServer, DataServer, and WebSpeed Transaction Server brokers.
`[UBroker.AS]`	Defines default property settings for all instances of an AppServer.
`[UBroker.AS.`*`product-instance-name`*`]`	Defines property settings for this instance of an AppServer. The `ubroker.properties` file can contain several of these entities each with a unique *`product-instance-name`*.

Parent entities provide default values for all of their child entities. However, at any child level, a redefinition of any value supersedes the default value of its parent. All children from the redefinition level down inherit this new value.

The `ubroker.properties` file is installed in the `properties` subdirectory of the Progress installation directory: for example, `%DLC%\properties\ubroker.properties` on Windows NT.

A similar file, `conmgr.properties`, stores all the properties for each instance of a Progress database.The `conmgr.properties` file is installed in the `properties` subdirectory of the Progress installation directory: for example, `%DLC%\properties\conmgr.properties` on Windows NT.

Working With the Supported Products

For the definitions and usage of all properties supported in the properties file, see the comments in the installed `ubroker.properties` file. For more information on:

- Each configuration management and validation utility, see the *Progress Client Deployment Guide* and the *Progress Database Administration Guide and Reference*.
- Maintaining `ubroker.properties` for AppServer installations, see the *Building Distributed Applications Using the Progress AppServer* manual.
- Maintaining `ubroker.properties` for DataServer installations, see the Progress DataServer Guide for your DataServer product.
- Maintaining `ubroker.properties` for WebSpeed installations, see the *WebSpeed Installation and Configuration Guide*.
- Maintaining `ubroker.properties` for NameServer installations, see "The Progress Unified Broker Administration Framework" section in Chapter 7, "Configuring Progress Unified Broker Products," and the configuration manual for each product that includes the NameServer in its configuration.
- Maintaining `ubroker.properties` for AppServer Internet Adapter installations, see *Progress on the Web*.
- Maintaining `ubroker.properties` for SonicMQ Adapter installations, see *Progress on the Web*.

Editing and Validating the Content Of the ubroker.properties File

If you do not have access to Progress Explorer and must configure these products within a Windows NT environment, you must edit the `ubroker.properties` file using a text editor such as Notepad.

To ensure proper run-time processing, the file must be named `ubroker.properties` and must be located in the `properties` subdirectory of the Progress installation directory. Follow these steps to edit the file and create or modify a product configuration:

1 ♦ Make a backup copy from the installed `ubroker.properties` file (and name it, for example, `test.properties`).

2 ♦ Make changes to `test.properties` with a text editor.

3 ♦ Run the appropriate validation utility on `test.properties` using the `-propfile` option to validate your changes. Table 6–1 lists the commands to run the validation utility for each supported product. The command you run must have local access to the `ubroker.properties` file whose configurations you want to validate.

Table 6–1: ubroker.properties File Validation Utilities

Product	Utility
AppServer	`asconfig`
AppServer Internet Adapter	`aiaconfig`
DataServer for Microsoft SQL Server	`mssconfig`
DataServer for ODBC	`odbconfig`
DataServer for Oracle	`oraconfig`
NameServer	`nsconfig`
SonicMQ Adapter	`adaptconfig`
WebSpeed	`wsconfig`

4 ♦ Shut down any running brokers or NameServers using the `-stop` option of the appropriate configuration management utilities (for example, `nsman` and `asbman`).

5 ♦ Copy `test.properties` to `ubroker.properties` in the `properties` subdirectory of the Progress installation directory. If you might return to the previous configuration, store a backup copy of the `ubroker.properties` file before replacing it with your modified version.

6 ♦ Restart your brokers and NameServers using the `-start` option of the appropriate configuration management utilities. Table 6–2 lists the commands to run the validation utility for each supported product.

Table 6–2: ubroker.properties File Management Utilities

Product	Utility
AppServer	`asbman`
DataServer for Microsoft SQL Server	`mssman`
DataServer for ODBC	`odbman`
DataServer for Oracle	`oraman`
NameServer	`nsman`
SonicMQ Adapter	`adaptman`
WebSpeed	`wtbman`

For more information on editing and validating the `ubroker.properties` file to configure a NameServer, see the "Editing the Properties File" section in Chapter 7, "Configuring Progress Unified Broker Products." For more information on editing and validating the file for each Unified Broker product, see your product documentation.

6.2 Shared-memory Configurations

Shared memory is an area in system memory that multiple users can access concurrently. Shared-memory systems run a version of Progress that uses the shared-memory architecture to provide superior performance and additional features.

On shared-memory systems, more than one server can access a single database, service local or remote clients, and handle a variety of network protocols. Progress keeps resources shared by all database users in shared memory and lets multiple servers access those resources efficiently. Optionally, additional asynchronous I/O processes can offload I/O operations from each server, further improving resource utilization.

Local clients running multi-user Progress can access database resources directly, rather than through a database server. This eliminates client/server message exchange and task-switching overhead. Database requests do not have to be queued until a server can process them. Local direct-access clients are known as *self-service clients*.

To run shared-memory versions of Progress over a network, you need information regarding network-related system files, network configuration, and the startup parameters required to start remote clients. For more information about the network files and configuration, see the "Client/Server and Progress AppServer In the Network Environment" and the "Preparing To Run Progress In a TCP/IP Network" sections in this chapter. For information about starting remote clients, see Chapter 8, "Starting and Running Progress."

Figure 6–1 shows a sample shared-memory configuration.

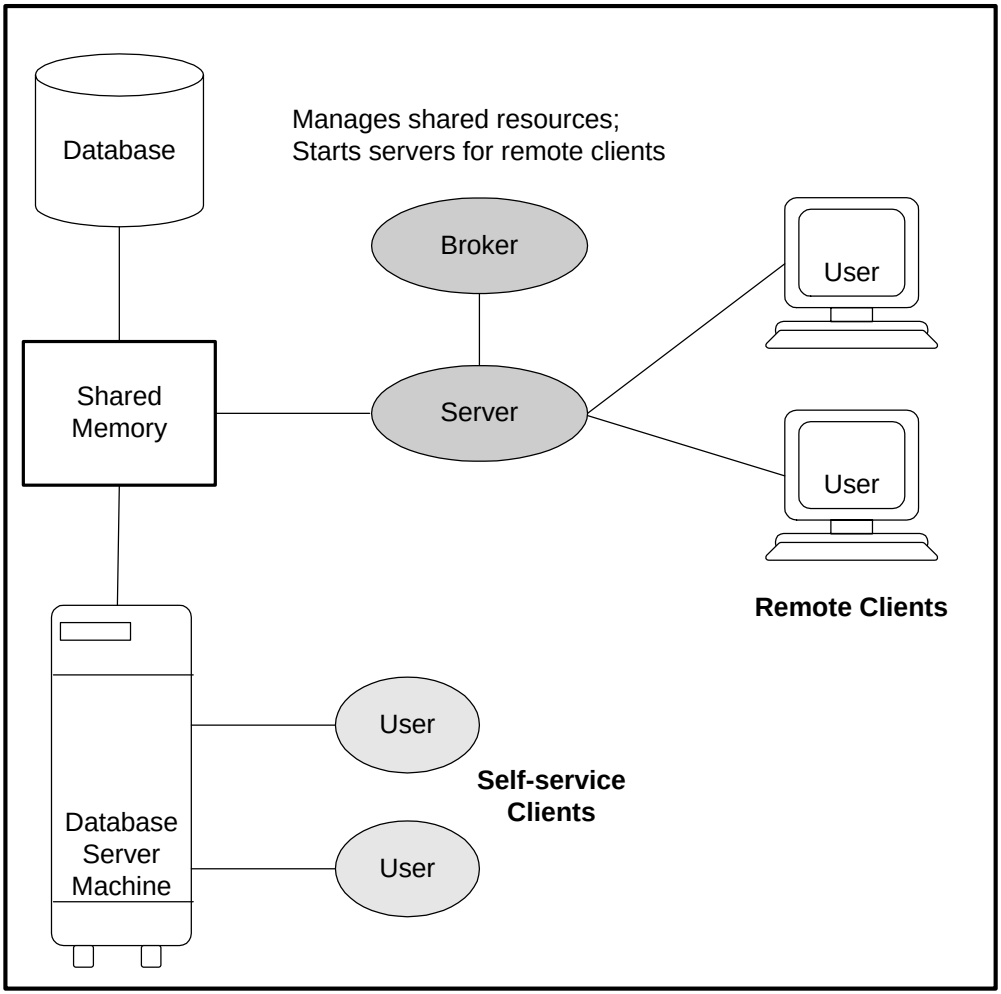


Figure 6–1: Database Access On a Shared-memory System

6.3 Multi-tier Configurations

For large installations Progress recommends a multi-tier configuration to improve system performance. Figure 6–2 illustrates a three-tier configuration that consists of a database-tier that supports self-service clients, an application tier that supports remote clients, and a thin-client tier.

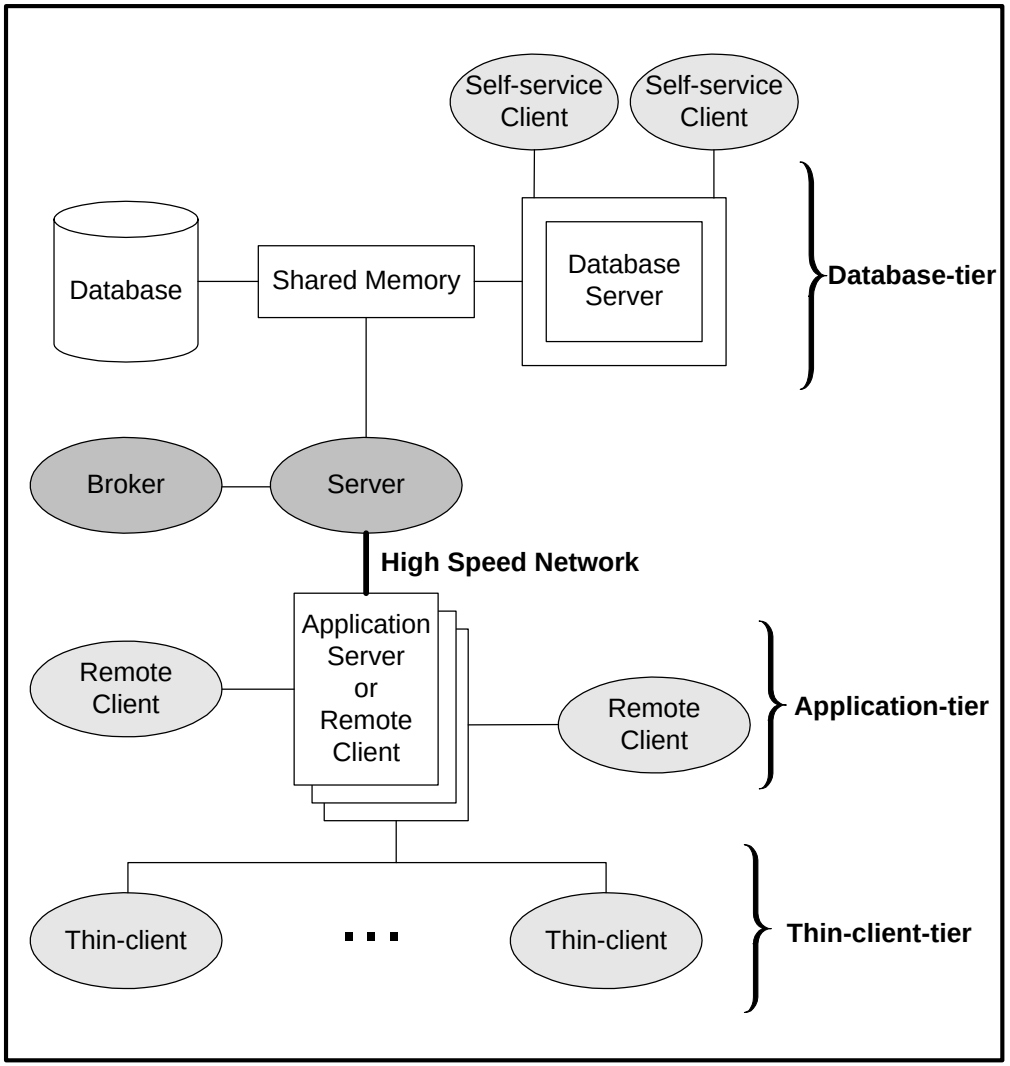


Figure 6–2: Sample Multi-tier Configuration

6.3.1 Shared-memory Architecture

Figure 6–3 shows the shared-memory Progress architecture.

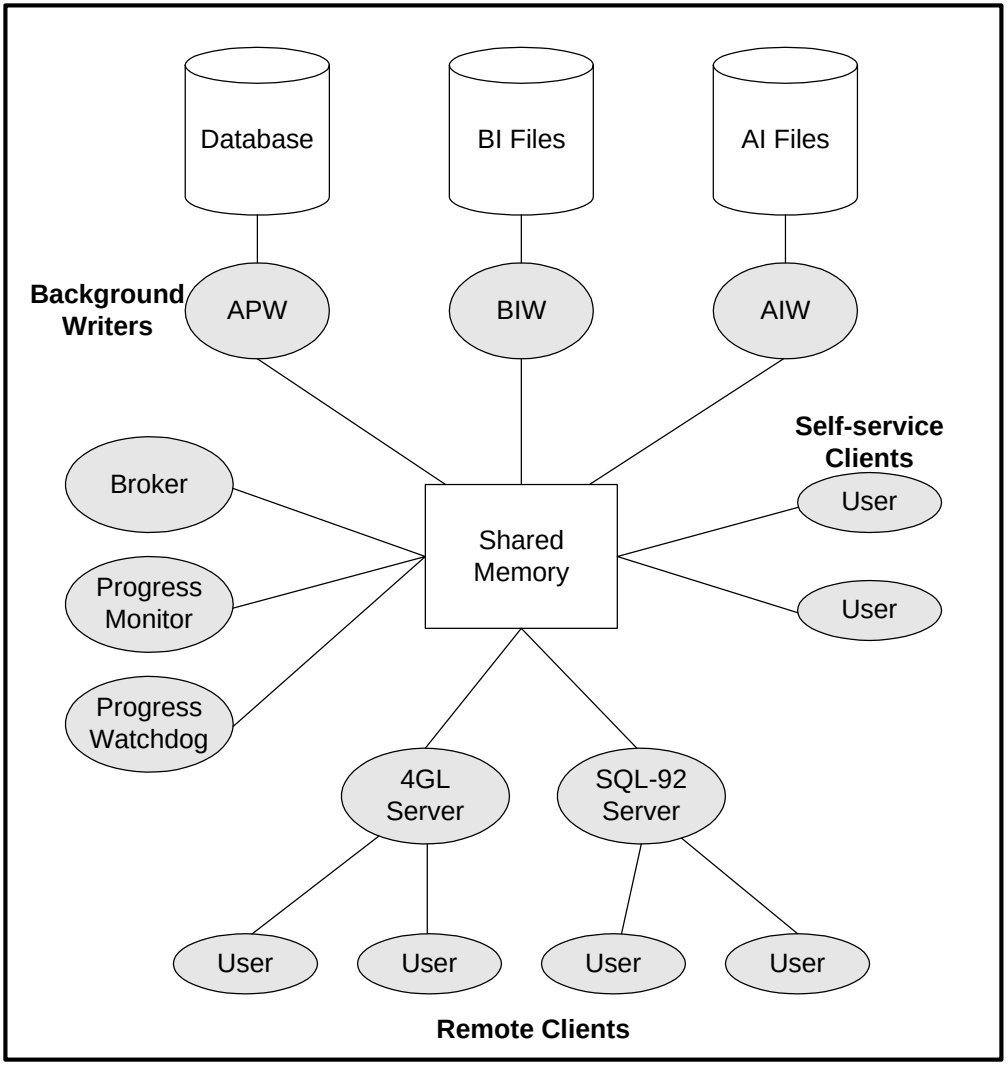


Figure 6–3: Shared-memory Progress Architecture

The following sections explain the components of the architecture that are specific to shared-memory systems.

Progress Database Monitor Utility

The Progress Database Monitor utility displays performance and usage information about database status and activity.

For more information about the Database Monitor utility, see the description of the PROMON utility in the *Progress Database Administration Guide and Reference*.

Progress NT Performance Monitor Utility

The Progress NT Performance Monitor utility provides a graphical interface that you can use to monitor performance and usage information about database status and activity.

For more information about the NT Performance Monitor utility, see the *Progress Database Administration Guide and Reference*.

Progress Watchdog Utility

If a process terminates improperly, it can maintain a lock on a record or shared-memory structure. This can impact database concurrency. The Progress Watchdog utility detects processes that have terminated improperly and cleans up after them.

At regular intervals, the Watchdog utility checks for processes that have terminated unexpectedly. If it finds one, it releases any locks or shared-memory structures that the process might hold.

The Watchdog utility checks for inactive processes approximately once every 10 seconds. It also checks for self-service clients that are no longer active, releases all the appropriate record locks, backs out of any live transactions, and releases any shared-memory locks. If a server process terminates unexpectedly, the Watchdog utility disconnects and cleans up the server's remote clients.

For more information about the Watchdog utility, see the description of the PROWDOG utility in the *Progress Startup Command and Parameter Reference*.

Background Writers

Shared-memory Progress offers three background writer processes that improve performance. These processes continually perform certain housekeeping functions in the background. Because these functions are performed regularly by the dedicated background writer processes, client and server processes rarely have to wait for these functions to be performed.

The three types of background writers, asynchronous page writers, before-image writers, and after-image writers, are described in the "Processes" section in Chapter 4, "Administration Utilities."

The AdminServer starts the background writers if the AdminServer has been configured to do this by Progress Explorer. For more information about background writers, see the *Progress Database Administration Guide and Reference*.

6.4 Client/Server Configurations

Wherever it runs, multi-user Progress functions in a client/server architecture. On a single machine, Progress provides multi-user access to a database by using a separate client process for each user. In a client/server configuration, one or more clients access the database through a server. The server provides a connection to the database through the shared memory. While separate and distinct, the Progress client and server processes compete for the same machine resources.

In client/server configurations, the client application and the database server are separate processes. Client processes can be local or remote.

The Progress user interface and Progress applications execute in the client session, sending requests to the Progress server. The Progress server accesses the database on behalf of each client session.

6.4.1 Terminology

This section introduces the terminology used to describe client/server configurations.

Application Workstation

An *application workstation* is any node that runs one or more Progress clients. For example, Windows NT is a multi-processor, so you can start more than one client on an application workstation. Depending on its operating system and configuration, an application workstation might run local clients and servers as well.

Database Server Machine

A *database server machine* is any node that runs one or more Progress servers for local or remote Progress clients.

Network File Server

A *network file server* is any node that provides shared services including file, printing, and security services to other nodes, including application workstations and database server machines. A network file server usually provides these services by allowing other nodes to access its local files and printers as if they were local to the other nodes. For example, Progress clients can run application procedures and Progress servers can access database files stored on a remote network file server.

A network operating system (NOS) is a network environment that includes one or more network file servers that provide a common set of resource sharing and security services to other nodes. A network file server usually runs the *kernel* of an NOS, the program that controls access to shared network resources. Depending on its operating system, a network file server might also run one or more Progress database clients and servers.

Although Progress Software recommends that you store the database on a disk locally attached to the database server machine, you can store the database on a network file server. Clients can access shared application code and communicate with the database server. Depending on your application and network environment, however, you might lose database integrity.

Note that Progress often runs in local area networks (LANs) that have no network file servers. On these LANs, application workstations can access only locally stored procedures, and database server machines can access only locally stored databases. However, the application workstations and database server machines can communicate with each other as remote processes.

Single-process Database Server Machine

A *single-process database server machine* is a node that runs only one server process for each database, providing access to that database for self-service clients only.

Multi-process Database Server Machine

A *multi-process database server machine* is a node that runs multiple server processes for each database, providing multiple data paths to the database. Each server queues and runs requests for one or more clients. A separate broker process starts a new server for each additional client (or set of clients, in specified increments) that access the database. All nodes that run operating systems with shared memory run as multi-process database server machines. Progress supports multi-process database server machines on Windows nodes. For more information, see the "Shared-memory Configurations" section.

You can dedicate all the resources of a database server machine to run database servers. However, depending on your application and operating system, you can also run local clients and remote clients for other database server machines.

6.4.2 Simple Client/Server Configurations

Figure 6–4 shows a simple client/server configuration, where the client and server components both run on a single system.

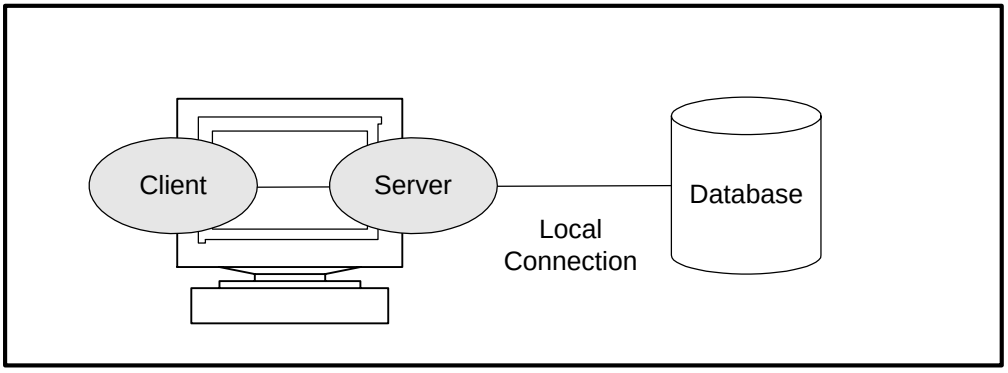


Figure 6–4: Simple Client/Server Configuration

Figure 6–5 shows a multiple system client/server configuration. In this configuration, the server runs on the system where the database resides. The clients run on remote systems, accessing the database through the server system.

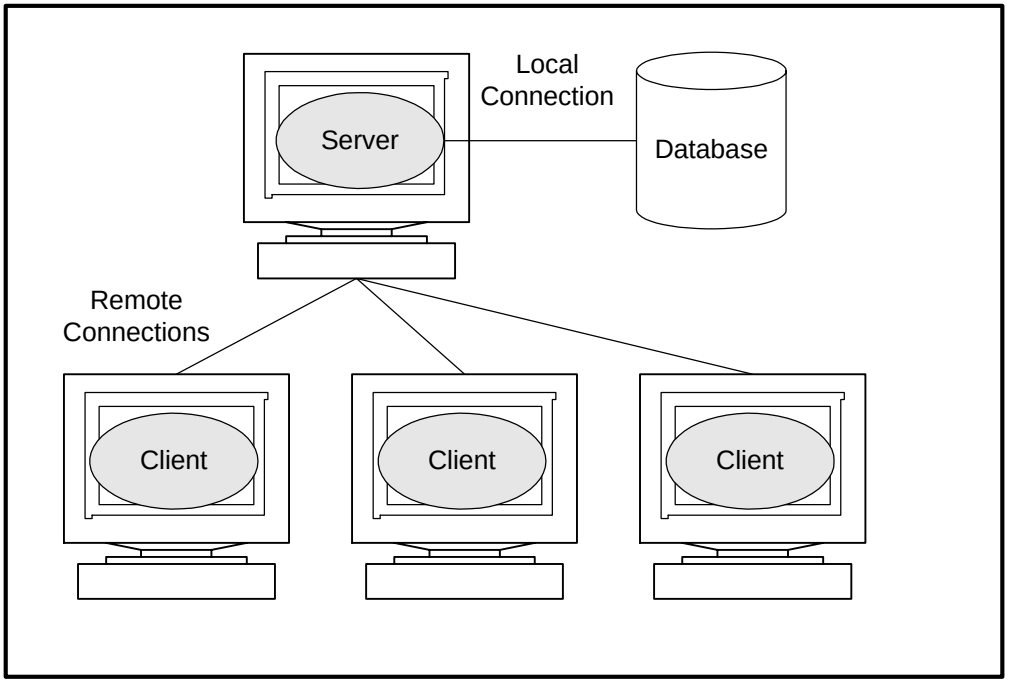


Figure 6–5: Multiple System Client/Server Configuration

6.5 Client/Server and Progress AppServer In the Network Environment

The Progress client/server architecture fits naturally into a network environment, allowing clients and servers to run together in many different (heterogeneous) hardware and operating system environments. On a network, the Progress client and server processes are distributed to separate nodes where they communicate through a common network protocol. Some nodes run client processes, while others run server processes. One advantage of this is that adding users or databases has minimal impact on the machine resources used by others. Each has its own resources devoted only to its client or server tasks. Another advantage is that a single Progress application can take advantage of the strengths of a multi-machine, multi-operating system environment, without regard to differences in file resources on the separate machines. Remote Progress clients and servers interact transparently, regardless of the type of machine environment in which they run. The result is a cooperative application environment with many more possibilities for expansion.

Progress Network Support

Table 6–3 describes the network types that Progress supports.

Table 6–3: Network Type Options For Progress

Network Type (-N)	Description
LFP	Allows Windows clients to access a Progress AS/400 server using ISO 802.2 raw frames. Thus, LFP is a proprietary protocol implemented at the data link layer of the OSI reference model. It is a high-speed alternative to Systems Network Architecture (SNA) that also uses the SNA session for connection monitoring. For more information, see the *Progress/400 Product Guide*.
SNA	Allows Windows clients to access a Progress AS/400 server using the APPC protocol suite (Advanced Program to Program Communication using SNA Logical Unit (LU) type 6.2). For more information, see the *Progress/400 Product Guide*.
TCP	Allows client and server operation among Windows systems that communicate using TCP/IP. In a Progress AppServer configuration, the client connection to the application server is always TCP. The Progress AppServer supports all of the client/server network types for the connection of the application server to the database server.

Sample Network Configurations

The following figures show sample configurations for Progress on a LAN. Your configuration might differ depending on your requirements.

Figure 6–6 shows the simplest Progress network configuration—a database server machine and an application workstation. Although the figure shows only one database server machine and workstation, there can be more than one of each.

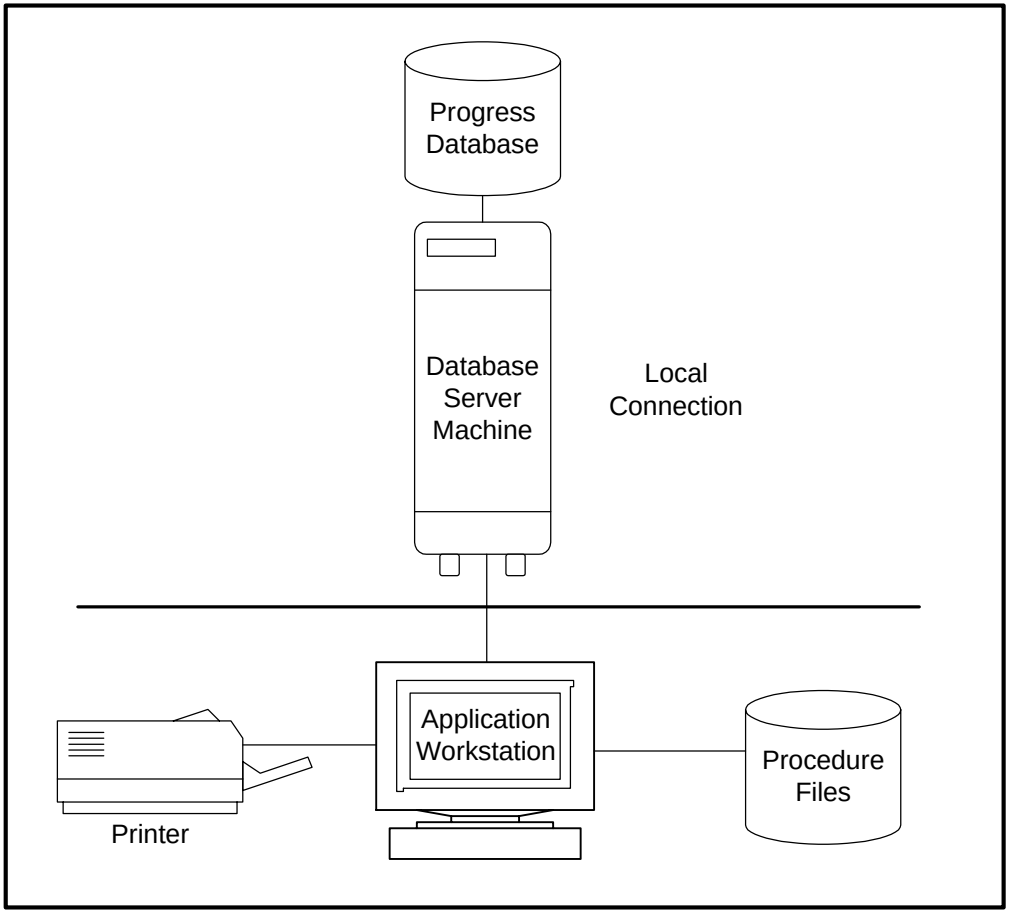


Figure 6–6: Simple Progress Network Configuration

This configuration is typical of TCP/IP networks without file servers. There are no shared resources except the Progress database. The application workstation and database server machine each have a hard disk. A printer is also attached to the application workstation. Progress is installed on each node.

A workstation in this configuration often supports multiple users and clients (for example, a system with multiple terminals) who share the local printer and Progress application. The database server machine is usually a high-performance back-end processor that can also support local self-service clients. This network configuration, with the Progress database local to the database server machine, ensures full database integrity. With all files stored local to each node, it generally (but not always) provides the highest performance on a LAN.

Figure 6–7 shows a dedicated network file server, dedicated Progress database server machine, and application workstations. Although the figure shows a limited number of workstations, file servers, and database server machines, there can be more of each.

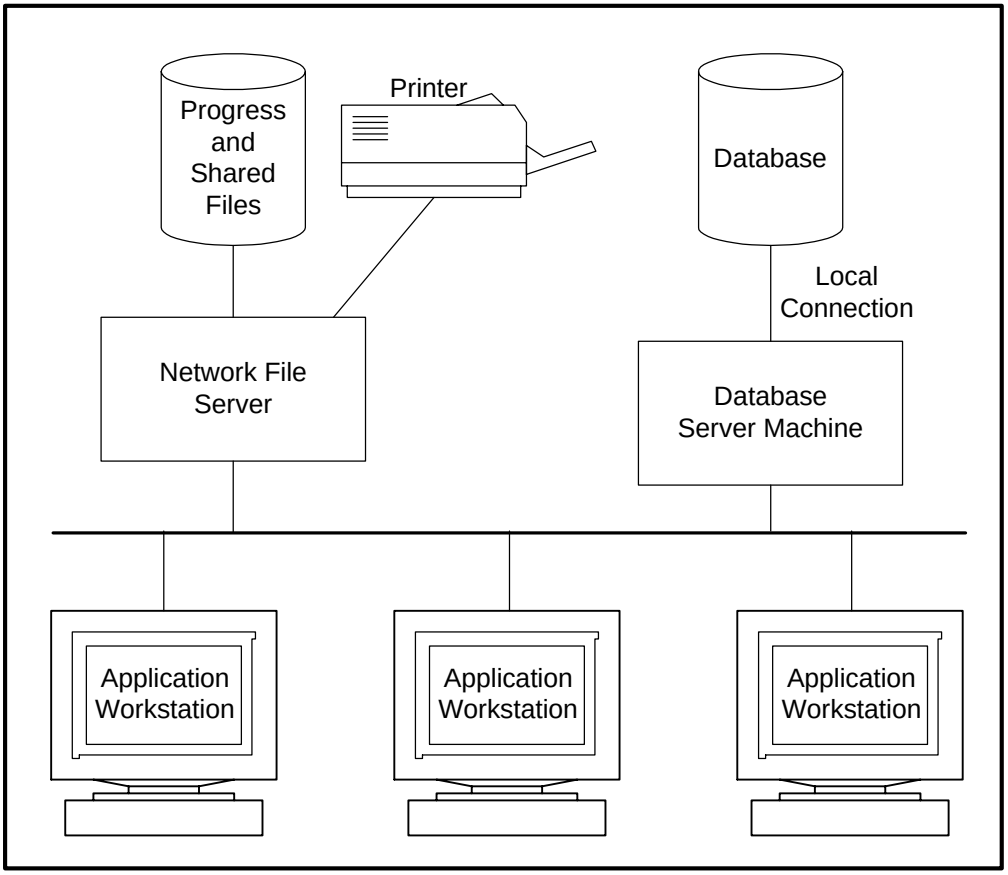


Figure 6–7: Network File Server For Application Files

This is a configuration typical of PC LANs with file servers and network operating systems. A hard disk and a printer are attached to the network file server, and an additional hard disk is attached to the Progress database server machine. The Progress database is on the disk drive that is locally attached to the Progress database server machine. Progress and all application procedures are installed on the file server and shared by all other nodes.

This network configuration ensures full database integrity and high performance, limited only by network and application performance capabilities.

Figure 6–8 shows a network file server doubling as a Progress database server machine and disk-optional application workstations. Although the figure shows a limited number of workstations, file servers, and database server machines, there can be more of each.

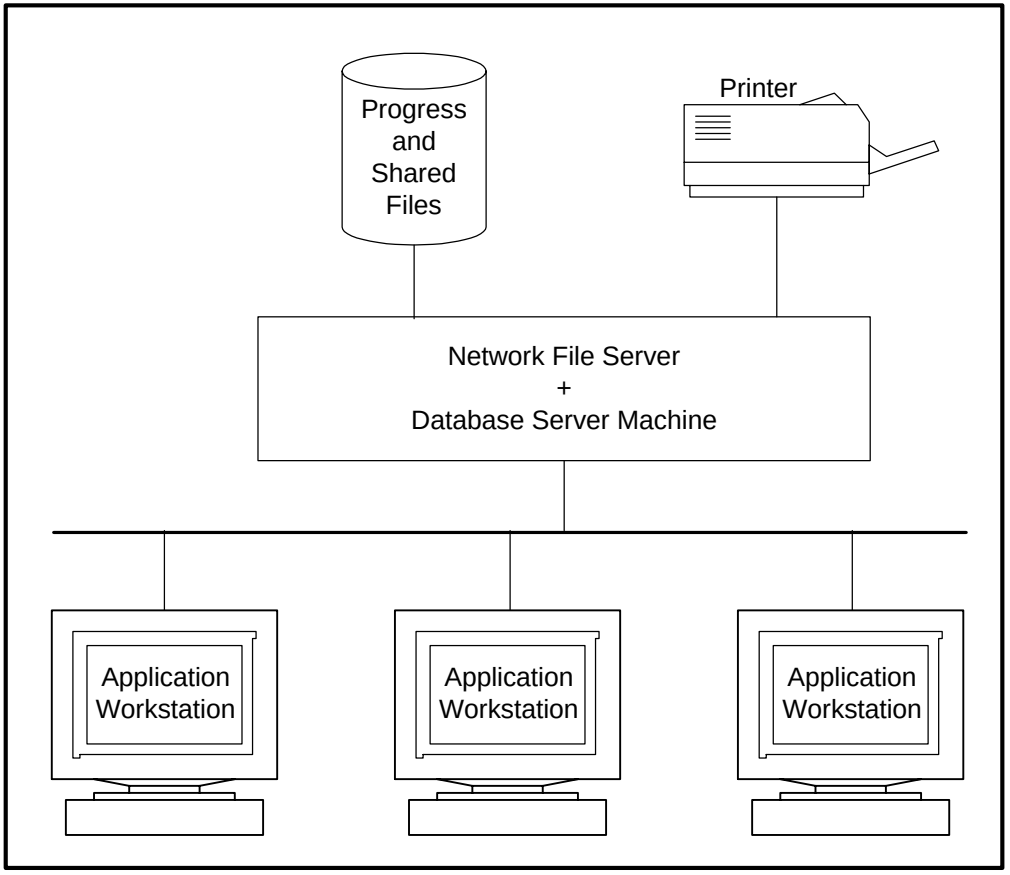


Figure 6–8: Network File Server As a Database Server

This is a configuration you might find on a PC LAN with a powerful file server running a multi-tasking operating system. Progress, application procedures, and the Progress database are all installed on the file server and are shared by the other nodes.

This network configuration provides full database integrity and acceptable performance on a file server with high-speed CPU and I/O resources.

NOTE: Avoid doubling a network file server as a database server machine on low-capacity nodes or on nodes where the database server machine can run only in an emulated environment.

Figure 6–9 shows database files residing on a network file server.

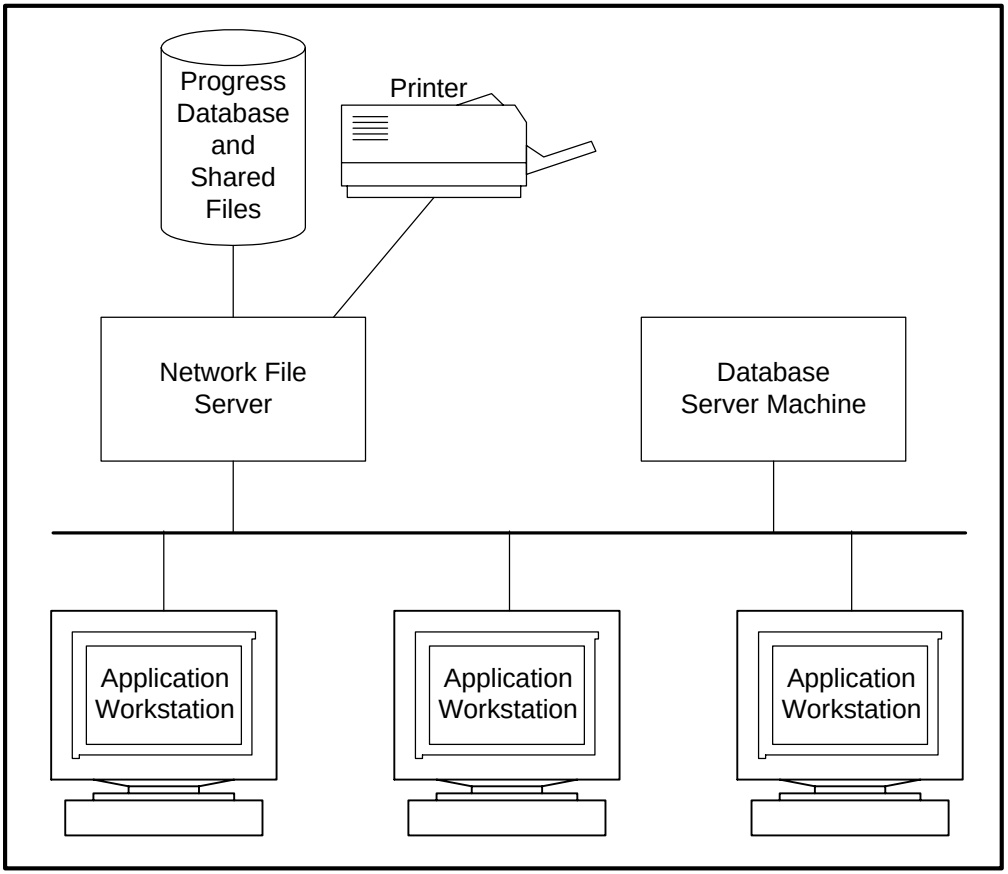


Figure 6–9: Network File Server For Application and Database Files

This network configuration runs the risk of compromising database integrity if the network file server or database server machine crashes, because the before-image (BI) file must be on the network file server, making synchronous writes to it impossible. Performance also depends on whether network file server I/O efficiency compensates for traffic across the network.

Figure 6–10 shows the simplest LAN configuration for Progress on a network.

An application server running on the application server machine connects through shared memory to a Progress database and has access to a set of procedure files. A Progress 4GL application runs at the application workstation, connects to the application server running on the Progress AppServer machine, and sends the requests to the application server to run remote procedures. The procedure execution and database access occur in a remote Progress session context.

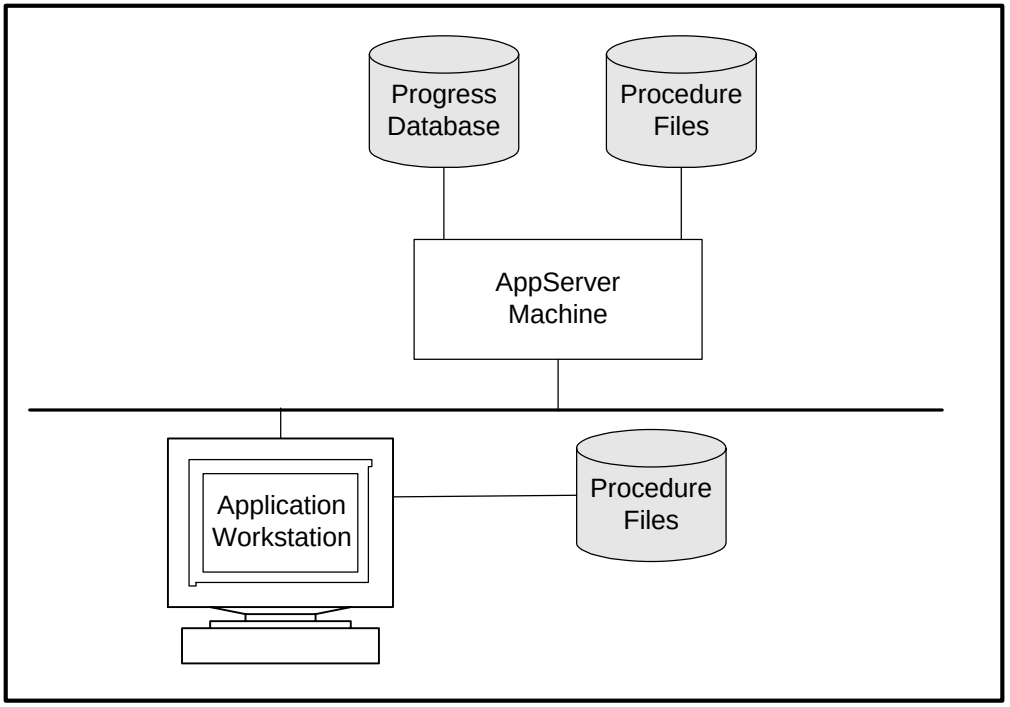


Figure 6–10: LAN Configuration With the Progress AppServer

In more complex implementations of the Progress AppServer, an application server can connect to another application server in order to connect with a database. For more information about the Progress AppServer, see *Building Distributed Applications Using the Progress AppServer*.

6.6 Preparing To Run Progress In a TCP/IP Network

You can make Progress operational in a network environment by following these guidelines:

- Identify and configure the nodes on your network for use as application workstations, database server machines, application server machines, and network file servers.
- Install Progress on each node, or if your network has a network file server, install Progress on the file server. For more information, see Chapter 3, "Sharing a Progress Installation On a Network."
- If any application workstations and database server machines have incompatible processors or operating systems, you must install the appropriate Progress product on each node.
- Set up network system files on each node.
- If you are using a network file server, make its resources, including printers and directories, available to all other nodes that require them.
- If you installed Progress on a network file server, you might want to distribute the appropriate Progress system files to the compatible application workstations and database server machines that use them. This takes advantage of networks where the local file and data access is faster than using the network.
- Set up your Progress databases on each file server, database server, and application server machine.

6.6.1 Installing Progress On Your TCP/IP Network

When installing Progress on your network, keep these two basic considerations in mind:

- Where to place your database
- Where to place your Progress executables and r-code files

Locating Your Database

Place your database on the hard disk of the machine that runs the Progress server. If you place the database on a remote file server, synchronous writes are lost along with your database's integrity, in the event of a system crash.

Synchronous writes ensure database integrity by flushing system buffers directly to disk. This is especially important for maintaining the before-image (BI) file. Therefore, if you must keep your database separate from the database server machine, use the Before-image Filename startup parameter to keep the before-image file local to the database server.

NOTE: Remote Progress clients do not have to be concerned about synchronous writes because they do not write to the database.

6.6.2 Typical TCP/IP Configuration With a Hard Disk On Each Machine

Figure 6–11 shows the configuration for a typical network when there is a hard disk on each machine and no file server is used.

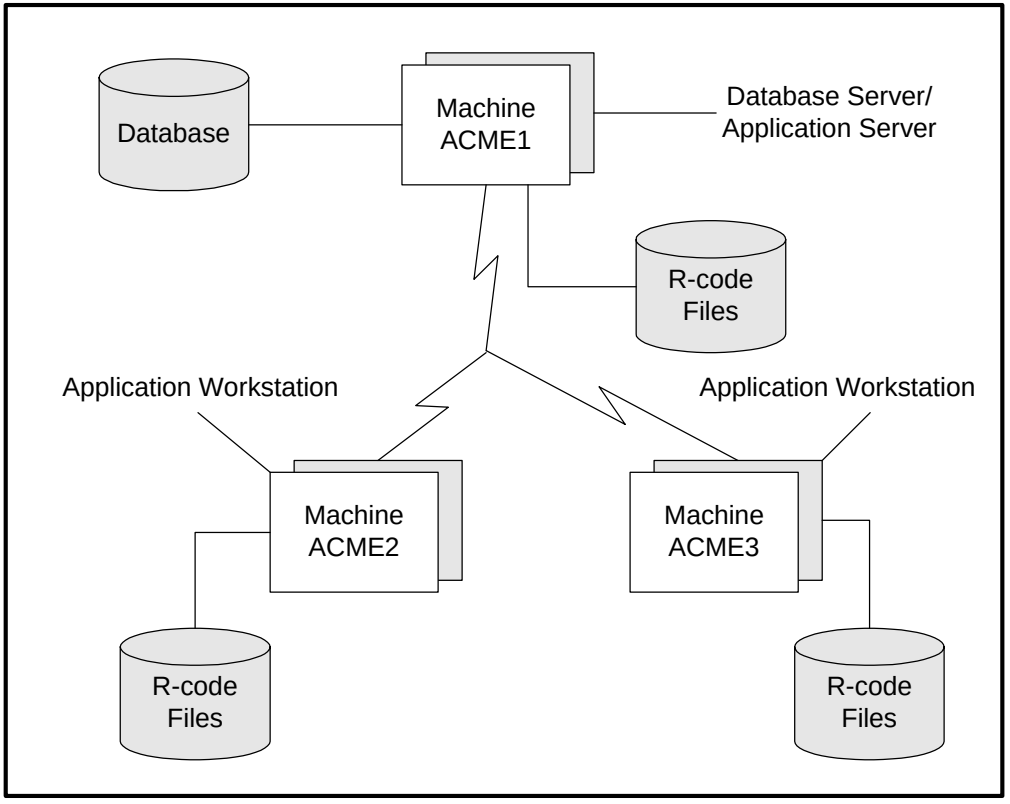


Figure 6–11: Typical TCP/IP Configuration (File Server Not Used)

When you use this configuration, you must install Progress on each machine in the network. In Figure 6–11, the client machines do not have to be running the same operating system.

6.6.3 Setting Up Network Files To Run Progress

There are several files you must check, and modify if necessary, before you can run Progress on your network. The filenames and locations might differ for different operating systems and TCP/IP implementations, but the functional contents are identical. Table 6–4 lists these files.

Table 6–4: TCP/IP Network Files

File	Purpose
`hosts`	Lists machine names/network addresses.
`services`	Lists Progress server/port number.
`protocols`	Defines system protocols.

6.6.4 Configuring Progress On a Network Operating System

This section describes preparations that you can make to promote efficient and reliable Progress operation in a network operating system (NOS) environment, that is, a network environment that includes one or more network file servers that provide a common set of resource sharing and security services to other nodes. This section describes some of the more general considerations.

Making Network Resources Available

Once you have installed Progress, you must make sure that each application workstation and Progress AppServer machine has access to Progress system files, application files, and any other necessary network resources (such as printers). Each NOS provides a set of commands or utilities to make these resources available across the network. In general, you set up pointers to remote resources so that each workstation can access them as though they were local to the workstation. These pointers can be in the form of logical drives (for example, on Windows nodes) or mounted directory paths (for example, on UNIX nodes).

For more information on making network resources available, refer to the documentation for your network and operating system.

Setting Network Resource Attributes

After you have made network resources available, you must make sure that they possess the necessary attributes to allow all application workstations to access them simultaneously. Each NOS provides a different means of setting the attributes to make network resources shareable.

For example, suppose you want to set the attributes of the Progress installation directory on a network file server so that all workstations can access the Progress files stored there. Progress is already loaded onto the network file server and is available to the network. The commands used to set resource attributes vary from network to network.

For further information on how to use these or equivalent commands for your network, refer to the documentation for your network and operating system.

Granting User Access Rights

After making Progress network resources available and setting resource attributes, you might have to grant access rights to client users and Progress AppServer machines in the network. Depending on your network, these access rights can include attributes such as read, execute, or open permissions that you must set for each user. See the network documentation for details about how to grant user access rights.

NOTE: User rights in a Progress AppServer configuration are assigned to the machine where the application server resides, not to the user's client machine.

Remember that a Progress database server can be a user on your network. Like application workstations, it might need user access rights granted to it. If you locate any database files on your network file server, be sure to grant the Progress database server the necessary rights to access the network directory that contains the database.

7

Configuring Progress Unified Broker Products

This chapter describes the framework and tasks for configuring the common components of Progress Unified Broker products that run on Windows. For more information on product-specific configuration, see the documentation for each product as explained in this chapter. These products all support multi-tier or client/server configurations using the Progress NameServer and include the Progress AppServer, WebSpeed, SonicMQ Adapter, and, when configured to use the Progress NameServer, the Progress DataServers for ODBC (Windows only), Microsoft SQL Server, and ORACLE.

Specifically, this chapter contains the following sections:

- The Progress Unified Broker Administration Framework
- Configuring Progress Unified Broker Components
- Setting Up the Progress Unified Broker Environment
- Starting Progress Unified Brokers
- Understanding Load Balancing
- Understanding Server-level and Connection-level Fault Tolerance

7.1 The Progress Unified Broker Administration Framework

Progress integrates all management functions for Unified Brokers in a single administration framework. Figure 7–1 shows an overview of this framework.

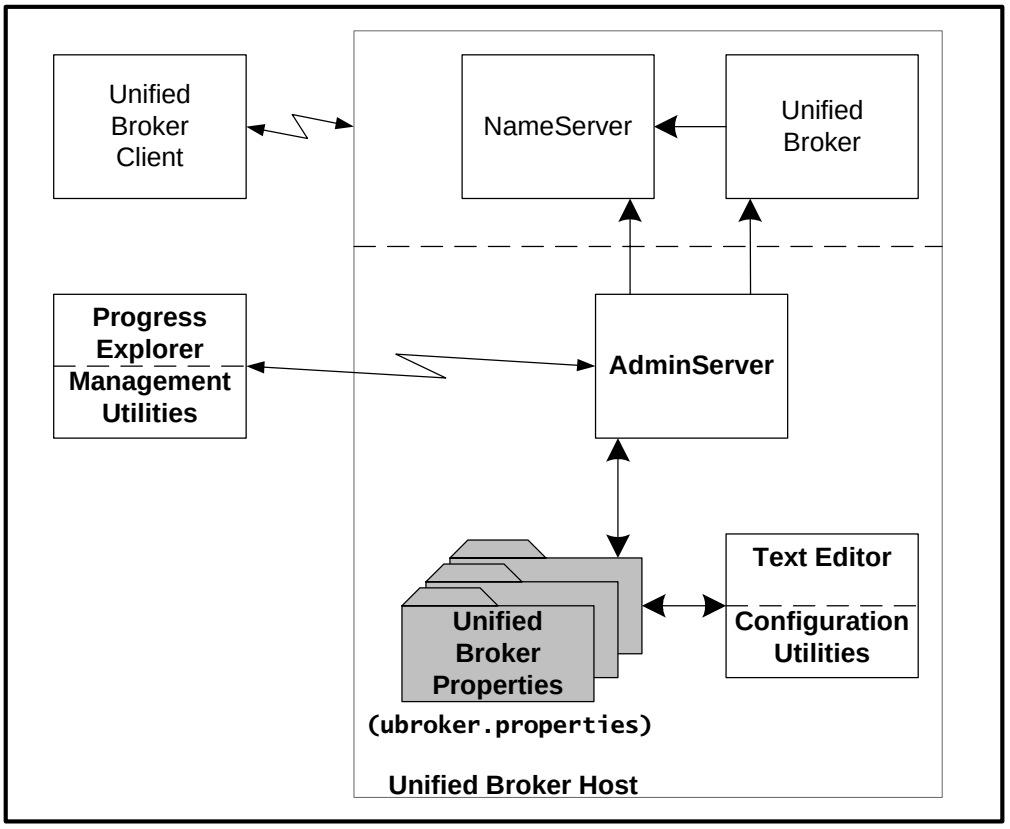


Figure 7–1: Progress Unified Broker Administration Framework

Figure 7–1 shows all the Unified Broker components running on the same machine. You can install some of these components on separate machines connected on the same network. In that case, a separate AdminServer and `ubroker.properties` file exists on each machine for access by the Progress Explorer. For more information, see the "Distributing Machines For a Unified Broker Product" section.

The core of this framework is a common process, the AdminServer process, for managing the AppServer, WebSpeed, SonicMQ Adapter, and DataServer products. The AdminServer supports the Progress Explorer and management utilities to provide similar configuration and management functions for all of these products.

Complete administration for a Unified Broker application potentially involves all of the components in Figure 7–1. However, for some components and administration tasks you can use AdminServer-based management utilities (including the Progress Explorer) or a text editor and configuration validation utilities to accomplish the task; for others you can only use the AdminServer-based utilities. Table 7–1 summarizes the components and possible administration approaches.

Table 7–1: Unified Broker Administration Summary

Components	Configuration	Run-time Management
Unified Broker clients	AdminServer-based utilities or text editor and configuration validation utilities (WebSpeed Messenger only). Otherwise, as specified for your client product.	AdminServer-based utilities (WebSpeed Messenger only). Otherwise, as specified for your client product.
NameServers	AdminServer-based utilities or text editor and configuration validation utilities.	AdminServer-based utilities.
Unified Brokers	AdminServer-based utilities or text editor and configuration validation utilities.	AdminServer-based utilities.
Unified Broker properties file, `ubroker.properties`	AdminServer-based utilities or text editor and configuration validation utilities.	NA

This chapter provides specific information on the NameServer and outlines the general procedures for administration of the remaining components. For specific information on administration of all application components for a Unified Broker product, see your product documentation:

- **AppServer** — *Building Distributed Applications Using the Progress AppServer*
- **AppServer Internet Adapter** — *Progress on the Web*
- **DataServers** — Depending on your product:
  - *Progress DataServer for Microsoft SQL Server Guide*
  - *Progress DataServer for ODBC Guide*
  - *Progress DataServer for ORACLE Guide*
- **SonicMQ Adapter** — *Progress on the Web*
- **WebSpeed** — *WebSpeed Installation and Configuration Guide*

7.1.1 Unified Broker Clients

The clients of Unified Broker products include:

- **AppServer** — 4GL clients (including other AppServers and WebSpeed instances) and Open Clients
- **AppServer Internet Adapter** — 4GL clients (including AppServer and WebSpeed instances)
- **DataServers** — 4GL clients (including AppServer and WebSpeed instances)
- **SonicMQ Adapter** — 4GL clients (including AppServer and WebSpeed instances)
- **WebSpeed** — The WebSpeed Messenger, which directs Web client requests to WebSpeed Transaction Servers

Of these clients, you can use the Unified Broker administration framework to manage only WebSpeed Messengers. For specific information on configuring these clients, see your Unified Broker product documentation.

7.1.2 Progress Explorer and Management Utilities

Progress provides two ways to manage local or remote Unified Brokers:

- Progress Explorer tool
- Command-line management utilities

Progress Explorer Tool

The Progress Explorer is a graphical user interface that provides an easy way for you to manage Progress servers. The Progress Explorer tool runs as a Windows client of the Progress AdminServer to manage the following Progress servers:

- AppServer
- AppServer Internet Adapter
- Database
- DataServer for Microsoft SQL Server
- DataServer for ODBC (Windows only)
- DataServer for ORACLE
- NameServer
- SonicMQ Adapter
- WebSpeed Messenger
- WebSpeed Transaction Server

Using the Progress Explorer tool, you can locally or remotely:

- Create new instances of Progress products and configure their property settings
- Remove existing Progress product instances
- Modify property settings of existing Progress product instances
- Start, stop, manage, and monitor the status of Progress product executions

This chapter describes basic Progress Explorer administration for all of these server products except the database. For information on Progress database administration, see the *Progress Database Administration Guide and Reference*. For an introduction to using the Progress Explorer tool, see Chapter 4, "Administration Utilities," and Chapter 6, "Configuring Progress." For more information on working with the Progress Explorer tool to manage all the Progress products, see the appropriate chapters of this manual and the Progress Explorer online help.

Command-line Management Utilities

The command-line management utilities are a set of utilities for Windows and UNIX that allow you to manage existing Unified Broker configurations, including the:

- **ADAPTMAN utility** — To manage SonicMQ Adapters
- **AIAMAN utility** — To manage AppServer Internet Adapters
- **ASBMAN utility** — To manage AppServers
- **MSSMAN utility** — To manage DataServers for Microsoft SQL Server
- **NSMAN utility** — To manage NameServers
- **ODBMAN utility** — To manage DataServers for ODBC (on Windows only)
- **ORAMAN utility** — To manage DataServers for ORACLE
- **WTBMAN utility** — To manage WebSpeed Transaction Servers

Like the Progress Explorer tool, the command-line management utilities run as clients of the Progress AdminServer to manage the NameServer and Unified Broker products. This chapter describes how to manage the NameServer using the NSMAN utility and introduces how to manage the Progress Unified Brokers using their own command-line management utilities.

Using these utilities, you can locally or remotely start, stop, manage, and monitor the status of Unified Broker execution. Unlike the Progress Explorer, they do not help you create, remove, or modify properties for Unified Broker configurations.

For more information on starting and managing the NameServer using the Progress Explorer and NSMAN utility, see the "Starting Progress Unified Brokers" section. For more information on using the Progress Explorer and management utilities for your Unified Broker product, see your product documentation.

The rest of this chapter describes how to configure NameServers and begin configuration and operation of your Unified Broker product.

7.1.3 NameServers

A *NameServer* is a single process that mediates client connections for a set of Unified Brokers that have registered with it. Any number and type of Unified Broker instance can register with a single NameServer, and each Unified Broker instance can register with exactly one NameServer. The NameServer that a broker instance registers with is the broker's *controlling NameServer*.

When a Unified Broker instance starts up, it registers with its controlling NameServer by sending its location and other configuration information. The NameServer uses this information to help resolve client connection requests. Part of this registration information is the Application Service that the Unified Broker supports. An *Application Service* is a designation for the particular business function that a Unified Broker provides. For more information on Application Services, see the "Unified Broker" section.

NameServers provide the following services for a Unified Broker product:

- **Location Transparency** — A requesting client does not need to know the network location of a Unified Broker instance. When a client attempts to create a connection to a Unified Broker instance, it first requests the connection from a NameServer to a broker that provides a specified Application Service. The NameServer then locates and assigns a broker to complete the connection that provides the specified Application Service.

- **Server-level Fault Tolerance and Load Balancing** — If you have installed the load-balancing option, you can provide server-level fault tolerance, where the NameServer can select from several Unified Broker instances to satisfy a client request. This option also allows you to balance connection load among multiple Unified Broker instances that provide the same Application Service. The NameServer then assigns connections among several Unified Broker instances based on a weight factor that you configure for each instance. For more information on server-level fault tolerance, see the "Understanding Server-level and Connection-level Fault Tolerance" section. For more information on how load-balancing affects Unified Broker operation, see the "Understanding Load Balancing" section.

- **Connection-level Fault Tolerance** — You can also make multiple NameServer instances available to help ensure that at least one NameServer is available even if another fails. In this type of configuration, one of several possible NameServers resolves the connection request. Thus, you can provide connection-level fault tolerance for requesting clients. For more information on connection-level fault tolerance, see the "Understanding Server-level and Connection-level Fault Tolerance" section.

For more information on how your Unified Broker product uses NameServers, see your product documentation.

7.1.4 Unified Broker

Unified Broker products include a Unified Broker process that is the initial point of client connection to a Unified Broker product instance. It is the process responsible for managing other process resources that are part of the product, and making those resources available to clients. As such, the Unified Broker process:

- Registers with a controlling NameServer the following information:
  - The broker's location on the network
  - The weight factor that you specify for load balancing
  - The Application Service that you specify
- Manages connections between clients and the Unified Broker instance.
- Provides other services unique to the Unified Broker product (AppServer, SonicMQ Adapter, WebSpeed, or DataServer). For example, it maintains the status of each 4GL process running on an AppServer and scales the number of processes according to changing demand.

Unified Broker Types

The Unified Broker that manages a particular Unified Broker product defines a unique type of Unified Broker specific to that product (AppServer, SonicMQ Adapter, WebSpeed, or DataServer).The product documentation for each Unified Broker type identifies its Unified Broker process by a unique name associated with that product, as follows:

- **AppServer** — Application Broker
- **DataServers** — DataServer Broker
- **SonicMQ Adapter** — SonicMQ Adapter Broker
- **WebSpeed** — WebSpeed Transaction Broker

NOTE: Each Unified Broker process manages only one Unified Broker product instance of the same type.

For more information on how to configure the Unified Broker process for a product, how clients specify connections, and how the Unified Broker manages connections with clients, see your product documentation.

Application Services

The Application Service that a Unified Broker provides is identified by a list of one or more names that you can optionally specify during broker configuration. Each Application Service name you specify is an arbitrary designation for the business function that the Unified Broker instance provides.

The NameServer maintains a separate Application Service name space for each Unified Broker type. So, an AppServer, SonicMQ Adapter, WebSpeed Transaction Server, and DataServer instance can each register the same Application Service name with the same controlling NameServer without conflict. However with the load-balancing option, if you have multiple Unified Broker instances of the same type register the same Application Service name with the same controlling NameServer, you must guarantee that each Unified Broker instance provides exactly the same functionality. For AppServers and WebSpeed Transaction Servers, this means providing the same application procedures and database resources for all instances. For DataServers, this means accessing the same database for all instances.

If, for example, you use the same Application Service name to identify functionality on several AppServers, each of which support different remote procedures and database connections, multiple requests from the same client application are likely to provide inconsistent results.

The Default Service

You do not have to designate an explicit Application Service for a Unified Broker. Instead, you can specify that the broker supports the *default service*. The default service is a special Application Service designation that supports default client connection requests. Thus, any Unified Broker that supports the default service and is of the appropriate type (AppServer, SonicMQ Adapter, WebSpeed, or DataServer) can satisfy a connection request from a client that does not specify an Application Service name as part of its connection request.

7.1.5 Unified Broker Properties File (ubroker.properties)

Progress stores the configurations for all the component configurations managed by the Unified Broker administration framework in a properties file, `ubroker.properties`. The UNIX and NT `ubroker.properties` files are the same except for platform-specific differences (for example, differences in directory path separators and the differences between environment variable references in UNIX and registry references in NT).

This file includes the configurations for:

- AppServers
- AppServer Internet Adapters
- NameServers

- Oracle DataServers
- ODBC DataServers (on Windows only)
- SonicMQ Adapters
- DataServer for Microsoft SQL Server
- WebSpeed Transaction Servers

There is one copy of this file local to each Progress installation. The AdminServer reads and updates the file according to your instructions using the Progress Explorer and management utilities. The `ubroker.properties` file is installed in the `properties` subdirectory of the Progress installation directory (for example, `$DLC/properties/ubroker.properties` on UNIX, or `%DLC%\properties\ubroker.properties` on NT). In order for the AdminServer to access the properties file, the file must reside in this directory.

NOTE: Before you install a new Unified Broker version, either to overwrite an existing installation or to add additional Progress components to the current installation, make copies of your `ubroker.properties`, `conmgr.properties`, and `proxygen.preferences` files and place them in another directory. This is necessary because the new installation **automatically** upgrades the files in the *`install-path`*`\properties` directory. After you have finished your new installation, replace the newly installed versions of these files with your copies. When you start the AdminServer, your older files will be updated to match the current standards for these files. When you uninstall an existing Progress Version 9 product, the process copies `ubroker.properties`, `conmgr.properties`, and `proxygen.preferences`, the three files in the *`install-path`*`\properties` directory, to `\TEMP`. After installing a new Progress Version 9 product, you can manually copy back the files from `\TEMP`.

Under certain conditions, you might have to modify this file using a text editor. For more information, see the "Using a Text Editor and Configuration Utilities" section. For more information on editing the properties file to configure NameServers, see the "Editing the Properties File" section. For more information on managing the `ubroker.properties` file for your Unified Broker product, see your product documentation.

7.1.6 Using a Text Editor and Configuration Utilities

You can update the `ubroker.properties` file manually by opening it locally on the machine where it resides using any text editor. This is the only method available to update Unified Broker configurations locally on a UNIX platform.

Guidelines For Editing the Properties File

In general, you should update all configurations (local or remote) using the Progress Explorer tool. If you must update a configuration locally using a text editor:

- You should not directly change the values in the `ubroker.properties` file unless you have a complete understanding of how the changes affect Unified Broker components. When possible, always use the Progress Explorer to make all changes to this file.
- Always make a copy of this file, edit the copy, and verify the result before replacing the original with your edited copy.
- For complete definitions of all the properties and detailed information on how to set them, see the comments included in the properties file.
- Verify the result using the appropriate configuration validation utilities.

The file consists of a hierarchical structure of configuration entities, where parent entities provide configuration information that you can override or extend in each child entity. Each configuration entity has a name that begins the entity definition, and the definition contains configuration settings for one or more products or product instances.

Validating the Properties File

Progress provides a set of configuration validation utilities that you can use to verify the correctness of any changes that you make to this file.

NOTE: If you always use the Progress Explorer, you never have to use these utilities.

You can use the:

- **ADAPTCONFIG utility** — To validate SonicMQ Adapter configurations
- **AIACONFIG utility** — To validate AppServer Internet Adapter configurations
- **ASCONFIG utility** — To validate AppServer configurations
- **MSSCONFIG utility** — To validate DataServer for Microsoft SQL Server configurations
- **NSCONFIG utility** — To validate NameServer configurations

- **ODBCONFIG utility** — To validate DataServer for ODBC configurations (on Windows only)
- **ORACONFIG utility** — To validate DataServer for ORACLE configurations
- **WSCONFIG utility** — To validate WebSpeed configurations

When you validate the `ubroker.properties` file, make sure that you run the configuration utilities that apply to all the component configurations stored in the file.

For more information on editing and validating the `ubroker.properties` file to configure a NameServer, see the "Editing the Properties File" section. For more information on editing and validating the file for each Unified Broker product, see your product documentation.

7.1.7 Overview Of Unified Broker Administration

There are two preliminary tasks you must complete before you can begin configuring and operating a Unified Broker product:

- **Configure and set up all machines involved in product installation and operation** — This depends on how you plan to distribute your product and its applications on a network. For more information on configuring Progress products on a network, see Chapter 6, "Configuring Progress."
- **Install the necessary product components** — Typically, this involves installing, on one or more network machines, the Progress Unified Broker product and whatever software components are required to use the product, such as Progress client or Web server software. If you plan to configure fault-tolerant servers or use load balancing, you must install a product that includes load balancing or install the load-balancing option for your product. For more information on Progress product installation, see the relevant chapters of this manual or the *Progress Installation and Configuration Guide Version 9 for UNIX*.

For more information on the distributing resources in a Unified Broker environment, see the "Distributing Machines For a Unified Broker Product" section in this chapter.

Once you complete these preliminary tasks, you can configure and start up Unified Broker instances, as shown in Figure 7–2.

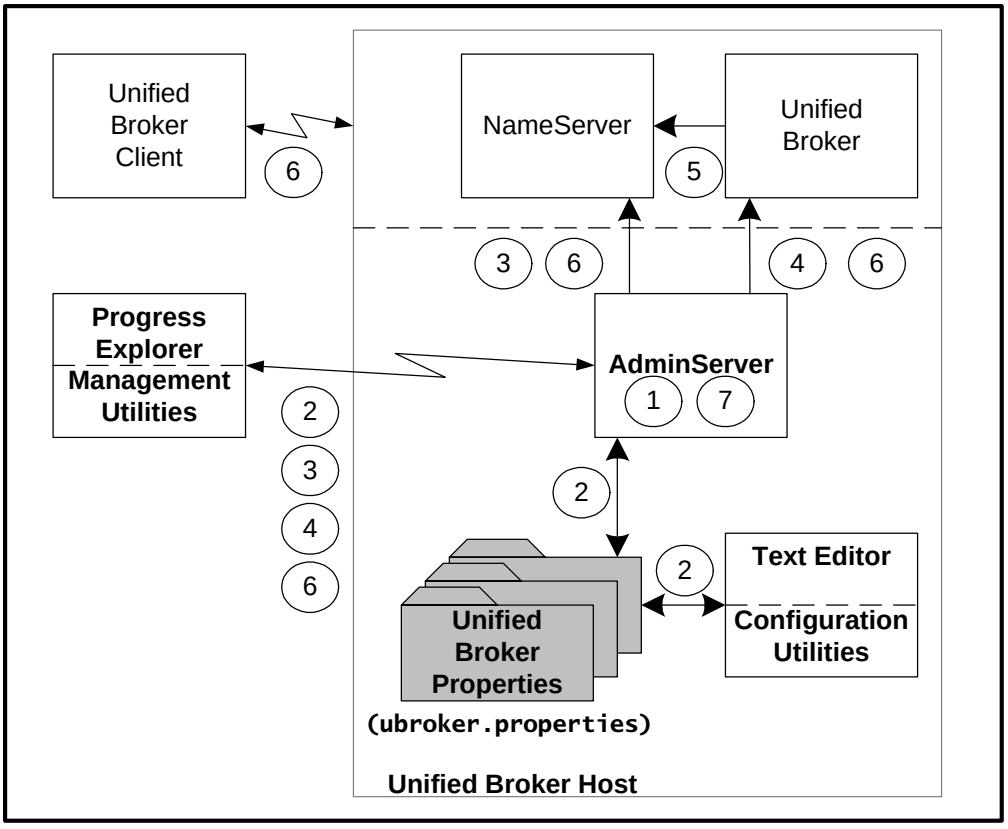


Figure 7–2: Progress Unified Broker Administration Overview

These are the steps, shown as circled numbers, in Figure 7–2:

1 ♦ Start the AdminServer process on each Unified Broker and NameServer installation machine. On NT, Progress installs the AdminServer as an NT service that starts automatically at system boot time. On UNIX, you can have the AdminServer started at system startup by editing your boot script to execute the PROADSV command. For more information on starting the AdminServer, see Chapter 5, "Setting Up Your Environment."

2 ♦ Once the AdminServer is running, use the Progress Explorer to create and modify NameServer and Unified Broker configurations. You can use the Progress Explorer locally on an NT machine or remotely from any Windows machine to access configurations installed on UNIX or NT machines. You can also use a text editor to create these configurations directly in the `ubroker.properties` file that is local to the NameServer or Unified Broker product installation machine, whether or not the AdminServer is running. For more information on configuring NameServer and Unified Broker components, see the "Configuring Progress Unified Broker Components" section.

3 ♦ Using the Progress Explorer (or the NSMAN utility), start the NameServer that you have configured to coordinate client access to one or more Unified Broker instances. For more information, see the "Starting Progress Unified Brokers" section.

NOTE: Figure 7–2 shows the NameServer running on the Unified Broker installation machine. However, the NameServer can run on any network machine that is accessible to all of the clients and Unified Broker instances that it coordinates.

4 ♦ Using the Progress Explorer (or the management utility for your Unified Broker product), start up each Unified Broker instance coordinated by the NameServer that you started in Step 3. As it starts, each Unified Broker instance starts additional processes or accesses resources, depending on the product and its configuration.

5 ♦ After a Unified Broker starts up, it registers its network address and the Application Services it supports with the NameServer that you started in Step 3.

6 ♦ A client can now make a Unified Broker connection request after you ensure that it knows:

- The correct network location of the NameServer to access
- The Application Service name required to connect to the broker that the client needs

At any time after Step 5, you can also use the Progress Explorer or the appropriate management utility to shut down or query the status of any running NameServer or Unified Broker instance.

7 ♦ You can shut down an AdminServer process at any time. If you have not already shut down the NameServer and the Unified Broker instance that it controls, the instance shuts down automatically when you shut down the AdminServer.

During Unified Broker operation, in addition to checking NameServer and Unified Broker status using the Progress Explorer and utilities, you can also review log files being generated by the NameServer and Application Server instance. For more information on working with these files, see the "Configuring Progress Unified Broker Log Files" section.

7.1.8 Distributing Machines For a Unified Broker Product

Figure 7–3 shows the maximum distribution of resources that you can achieve for any single Unified Broker product instance. For WebSpeed the, Unified Broker Client (WebSpeed Messenger) also resides on the same machine as the Web server, and Web clients (browsers, not shown) can reside anywhere on the Internet, intranet, or extranet serviced by the Web server.

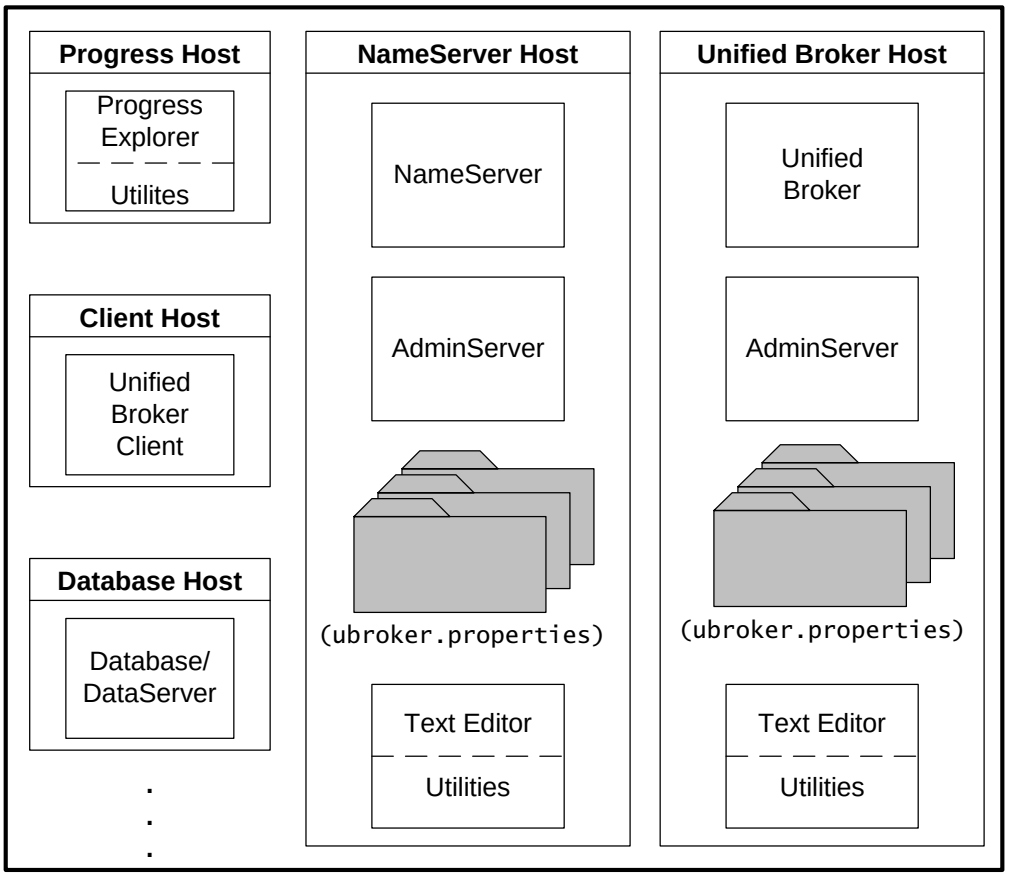


Figure 7–3: Maximum Unified Broker Machine Distribution

The separate database host for a DataServer applies only to WebSpeed or the AppServer. For a DataServer, the Unified Broker host is the DataServer host. Note that whether the target database management system (DBMS) can reside on a separate database host or must reside on the same machine as the DataServer host depends on the DataServer and its platform.

Thus, you can distribute a Unified Broker instance and its management components among three separate machines on the same network. Also, the Unified Broker client and any single Progress data source can reside on separate machines, for a total distribution of five machines plus any machines in the same network that you might require for additional databases, DataServers, or other Unified Broker instances.

Note that for all Unified Broker products, these components must reside on the same machine:

- Unified Broker instance and associated processes
- AdminServer
- Unified Broker properties file (`ubroker.properties`)

Similarly, the NameServer, if it resides on its own machine, is installed with an administration framework, including an AdminServer and its own `ubroker.properties` file. If you use a text editor to modify the `ubroker.properties` file, the editor and configuration utilities must reside on the same machine as the Unified Broker or NameServer instance, or have network file system access to the respective Unified Broker and NameServer installation files.

7.1.9 Configuring NameServer Communications

Both clients and Unified Brokers use User Datagram Protocol (UDP) to communicate with NameServers. UDP is an Internet standard, combined network layer, transport layer, and session layer protocol that provides a mechanism for connectionless communications. The connectionless nature of this protocol affords built-in benefits, such as the ability to implement fault-tolerant NameServers for client connections. For information on connection-level fault tolerance, see the "Understanding Server-level and Connection-level Fault Tolerance" section.

To establish a Unified Broker connection, a Unified Broker client must specify the location of the NameServer that provides the connection. To register with a NameServer, a Unified Broker must specify the location of the NameServer where it needs to register. To specify the NameServer location, both components must know the UDP port number (or service name) on which the NameServer is listening and host address of the machine where it resides.

Specifying NameServer Ports and Hosts

4GL clients provide this information when they specify the CONNECT( ) method, and AppServer Open Clients provide it using an equivalent Open Client method. You must specify this information for NameServers and other Unified Broker components in the Progress Explorer or directly in the Unified Broker properties file (`ubroker.properties`). For more information, see the "Configuring Progress Unified Broker Components" section.

If the component uses a UDP service name rather than a port number, you also must ensure that the `services` file on the component host (or Network Information Services (NIS), if used) properly defines the UDP service name.

Editing the services File

The `services` file stores the service name, port number, and protocol for various services on the network. For each NameServer that the component accesses, the `services` file on the component host must specify a service name associated with the NameServer UDP port number (default, 5162). Thus, if the component connects using the service name `namesv`, you might enter the service name definition for the `services` file extract shown in Figure 7–4.

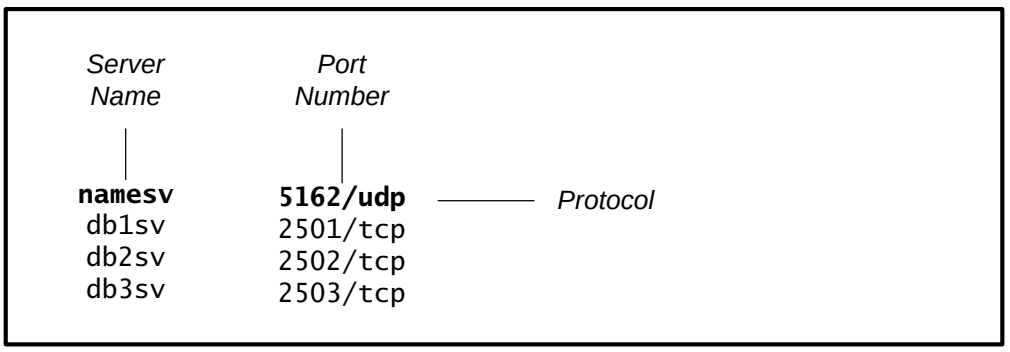


Figure 7–4: Sample Unified Broker Client Services File

7.2 Configuring Progress Unified Broker Components

You can use the Progress Explorer to configure NameServer and Unified Broker instances, locally on NT or remotely for both NT and UNIX hosts. If you plan to configure instances directly on a UNIX host, you must edit the properties file (`ubroker.properties`) for each NameServer and Unified Broker instance directly on the host. The properties file that comes installed with your Unified Broker product includes one sample NameServer and Unified Broker instance for each type of Unified Broker that you can use as a guide.

7.2.1 Downloading NameServer Executables

If you need a NameServer executable, for example, for a different deployment platform, you can download it from the Progress Web site at `www.progress.com`:

1 ♦ Choose the **Tech Support** page.

2 ♦ Choose the **Product Updates and Documentation** page.

3 ♦ Go to the **Downloads** section, and follow the instructions to download the Progress NameServer executable for your Progress version and platform.

7.2.2 Order Of Configuration

To configure a Unified Broker instance, you generally configure components in this order:

1. Controlling NameServer and any replicated or neighbor NameServers

2. Unified Broker product components

In the Progress Explorer, you must have an initial configuration for the controlling NameServer instance to identify it when you configure your Unified Broker product instance. Editing the properties file, you can configure these components in any order. Whatever order you configure these components, you must have the controlling NameServer configured and running before clients can access your Unified Broker instance.

7.2.3 Configuring and Using NameServer Instances

You can configure two types of NameServer instance, determined by its function as the controlling NameServer for a Unified Broker instance:

- **Local** — An instance that runs locally on the host where it is defined

- **Remote** — An instance that references a NameServer defined and running locally on a machine that is remote from the host where the remote instance is defined

When you configure a local NameServer instance, you can set all properties for the NameServer. When you configure a remote NameServer instance, you can only set its location (host and port) properties to identify the local NameServer instance that it references. When you want to start, stop, or obtain status on a running NameServer, you must always perform these actions on a local instance. You cannot start, stop, or obtain status on a remote NameServer instance.

How Unified Brokers Use NameServer Instances

To use a local NameServer instance as its controlling NameServer, a Unified Broker instance must run on the same machine where the local NameServer instance runs. Remote NameServer instances provide a way of having multiple Unified Broker instances use a controlling NameServer that runs on a different machine from the Unified Broker instances.

Whether local or remote, the NameServer instance that you define as the controlling NameServer must be defined on the same machine as the Unified Broker instance it controls. If the controlling NameServer instance is local, it runs on the same machine as the Unified Broker. If the controlling NameServer instance is remote, it references a NameServer running locally on a machine that is remote from the Unified Broker.

Thus, any remote NameServer instance you define must have a corresponding local NameServer instance defined on the machine where it runs, and you must define one such remote NameServer instance on each remote machine where a Unified Broker instance references this same corresponding local NameServer instance as its controlling NameServer.

NameServer Instances and Client Connections

Unified Broker clients do not use local and remote NameServer instances. Clients must direct all connection requests to a NameServer on the machine where it runs, that is, to a NameServer where it is defined as a local instance.

7.2.4 Configuration Using the Progress Explorer Tool

To access the Progress Explorer tool, you can open it in the Microsoft Management Console (MMC) or from the Progress program group on your Windows system. For more information, see Chapter 4, “Administration Utilities.” Once you have opened the Progress Explorer in the MMC, follow these basic steps to configure a Unified Broker instance:

1 ♦ Make sure that the AdminServer process is running on each of the following machines:

- Where the Unified Broker product you want to configure is installed
- Where all NameServers you want to configure are installed

2 ♦ In the Progress Explorer, connect to the running AdminServer processes that you verified in Step 1. The name or IP address of the machine running each of these processes appears as a folder in the tree view.

3 ♦ Expand each machine folder to see a list of folders that show the names of the NameServer and Unified Broker products installed on the machine.

4 ♦ Select the NameServer folder where you want to configure the controlling NameServer for the Unified Broker instance you want to configure. For more information, see the "Configuring the NameServer In the Progress Explorer" section. Repeat this step for any replicated NameServer or NameServer neighbor instances that you need to support the Unified Broker.

5 ♦ Configure your Unified Broker product components. You can configure the Unified Broker product much like the NameServer. For more information, see your product documentation.

For more information on using the Progress Explorer to configure a Unified Broker instance, see the Progress Explorer online help.

7.2.5 Configuring the NameServer In the Progress Explorer

You can use the Progress Explorer to define and configure a NameServer instance. (See the Progress Explorer online help for detailed information.) When you configure a NameServer instance, you can do the following things in each of the property categories:

- **Location** — Set port numbers and connection types.
- **General** — Set the working directory, whether the NameServer starts automatically, and when the NameServer unregisters brokers.
- **Logging Setting** — Set how the NameServer logs events.
- **Advanced Features** — Specify one or more NameServers from the already-configured NameServers to serve as Neighboring NameServers. These are the NameServers that provide connection-level fault tolerance for a Unified Broker application. For more information, see the "Understanding Server-level and Connection-level Fault Tolerance" section.
- **Environment Variables** — Set environment variables for NameServer execution; see Chapter 5, "Setting Up Your Environment," for more information.

7.2.6 Editing the Properties File

The Unified Broker properties file, `ubroker.properties`, stores the configuration definitions for all instances of the NameServer, and all instances of any AppServer, AppServer Internet Adapter, SonicMQ Adapter, WebSpeed Transaction Server, and DataServer products that run on the same machine. Each configuration definition contains property settings, environment variables (to be set when the Unified Broker starts), and references to registry entries for each product instance. Because the same properties file stores configurations for several different products, some sections of the file are generic to several different product broker configurations.

There is one copy of this file for each Progress installation. Thus, if you install the controlling NameServer on a separate machine from the Unified Broker product that it controls, the NameServer and Unified Broker product each have their own `ubroker.properties` file. For more information on the Unified Broker properties file, see the "Unified Broker Properties File (ubroker.properties)" section.

Editing and Validating the Properties File

To edit the properties file directly, use a text editor such as `vi` or Notepad. Once you edit the properties file, use the NSCONFIG command to validate the NameServer configuration information in the file. If the file contains any other Unified Broker configurations, run the configuration validation utilities for those Unified Broker products to ensure that these configurations are still valid. For more information, see the "Using a Text Editor and Configuration Utilities" section.

The NSCONFIG utility displays the property settings associated with a NameServer configuration, and checks that the syntax and values are valid. You must run the NSCONFIG utility locally on the machine on which the NameServer is running. The utility does not run across the network.

This is the syntax to invoke the NSCONFIG utility:

SYNTAX

```
nsconfig
  [
     [
        [ -name name-server ]
        [ -propfile path-to-properties-file ]
        [ -validate ]
     ]
   | -help
  ]
```

`-name` *`name-server`*

Specifies which existing NameServer configuration to examine. The name must match the name of an existing NameServer configuration in the specified properties file. If you do not specify a NameServer, the NSCONFIG utility analyzes all NameServer configurations defined in the properties file specified by the `-propfile` parameter.

`-propfile` *`path-to-properties-file`*

Specifies a filename or pathname to a file that contains the property settings to be validated, for example `test.properties`. If a filename or pathname is not specified, it defaults to the installation version of the `ubroker.properties` file, such as:

- `%DLC%\properties\ubroker.properties` on NT
- `$DLC/properties/ubroker.properties` on UNIX

`-validate`

Checks the syntax and values of property settings defined in the specified properties file.

`-help`

Displays command-line help.

For example, the following command validates the syntax and views the configurations of all NameServer instances defined within the `test.properties` file located in the current working directory:

```
nsconfig -propfile test.properties -validate
```

Once you have configured the controlling NameServer, you can configure your Unified Broker product components. You can configure each Unified Broker product much like the NameServer. For more information, see your product documentation.

7.3 Setting Up the Progress Unified Broker Environment

You might need to perform the following tasks before you start a Unified Broker instance:

- Set the standard Progress environment variables (for example, DLC) on the Unified Broker machine.
- Decide where to put your Unified Broker log files.
- Perform other preparations, depending on your product and configuration.

7.3.1 Setting Environment Variables

The environment variable settings required by a Unified Broker product vary according to the product. You can set them during product installation or when you configure the product. You can override most of these settings using the Progress Explorer or by editing the properties file (`ubroker.properties`).

Working Directory Settings

The properties file initially references a default environment setting for the working directory that you specify during installation. On NT, this is set in the registry and referenced in the properties file using `@{WorkPath}`. On UNIX, this is set in the `proadsv` script used to start the AdminServer and referenced in the properties file using $WRKDIR. You can remove or modify these references in the properties file to establish your own working directory settings for each Unified Broker and NameServer instance.

General Environment Settings

For more information on Progress environment settings, see Chapter 5, “Setting Up Your Environment,” and also see the *Progress Client Deployment Guide* for information on setting up the environment for any Progress clients you are using with your Unified Broker product. For information on other environment variables unique to your Unified Broker environment, see your product documentation.

7.3.2 Configuring Progress Unified Broker Log Files

A Unified Broker installation generates a total of three log files, which (by default) are stored in the working directory with the following default names, where *type* is the Unified Broker type (for example, AS for AppServer) and *instance-name* is the name of the Unified Broker or NameServer instance:

- *instance-name*.broker.log — Unified Broker log file set using the Progress Explorer or the brokerLogFile property in the [Ubroker.*type*.*instance-name*] section (for example, [Ubroker.AS.asbroker1]) of the properties file. This log file records the connection and management activity of the Unified Broker instance.

- *instance-name*.server.log — Server process log file set using the srvrLogFile property in the [Ubroker.*type*.*instance-name*] section (for example, [Ubroker.AS.asbroker1]) of the properties file. This log file records the activity of server processes that directly handle client requests for the Unified Broker instance.

- *instance-name*.NS.log — NameServer log file set using the srvrLogFile property in the [NameServer.*instance-name*] section of the properties file. This log file records the activity of the controlling NameServer for a Unified Broker instance.

For disk management reasons, you might want to specify a nondefault location for the log files used by a Unified Broker. After you decide where you want these log files to reside, you can specify the location for each in the Progress Explorer or by directly editing the ubroker.properties file. For more information, see the "Configuring Progress Unified Broker Components" section.

You can also view these log files using the Progress Explorer or by using your system text file commands and utilities.

The *instance-name*.broker.log and NameServer *instance-name*.log files include:

- Information on when Unified Broker and NameServer processes start up and shut down.

- System messages and information up to a level of detail that you specify using the Progress Explorer or the loggingLevel property in the ubroker.properties file.

The *instance-name*.server.log file includes all trace messages and Progress system messages that are not assigned an output destination by the OUTPUT TO KEEP-MESSAGES statement.

Because log files receive Unified Broker and NameServer startup and shutdown messages, Progress system messages, and trace messages, they can grow quickly. Also, they do not truncate automatically and can grow indefinitely. To manage disk space, periodically trim the file with a text editor after archiving any information that you want to save.

7.3.3 Other Preparations Prior To Startup

Depending on you product, you might need to perform other tasks before starting a Unified Broker instance. For example, you might have to customize the executables that execute the 4GL for an AppServer or WebSpeed application. You typically need to do this only if your application configuration requires nondefault Progress configurable elements, such as a DataServer. For more information, see your product documentation.

7.4 Starting Progress Unified Brokers

You can start a Unified Broker instance using the Progress Explorer or the appropriate command-line management utilities.

7.4.1 Using the Progress Explorer To Start a Unified Broker Instance

To access the Progress Explorer tool, you must open it in the Microsoft Management Console (MMC) on your Windows system. For more information, see Chapter 4, "Administration Utilities." Once you have opened the Progress Explorer in the MMC, follow these basic steps to start and manage a Unified Broker instance:

1 ♦ Make sure that the AdminServer process is running on each of the following machines:

- Where the Unified Broker instance you want to start resides
- Where all required NameServers reside
- Where any Progress databases or other DataServers reside that your Unified Broker instance needs to access

2 ♦ In the Progress Explorer, connect to the running AdminServer processes that you verified in Step 1. The name or IP address of the machine running each of these processes appears as a folder in the tree view.

3 ♦ Expand each machine folder to see a list of folders showing the name of each NameServer and Unified Broker product installed on the machine.

4 ♦ Expand the NameServer folder under each machine to see a list of configured NameServers.

5 ♦ If they are not already running, start each NameServer that you need (more than one if using connection-level fault tolerance) to support your Unified Broker instance. For more information, see the "Starting and Managing a NameServer Using the Progress Explorer" section.

6 ♦ Start each Progress database or DataServer that your application requires. For more information on starting databases and DataServers, see Chapter 8, "Starting and Running Progress," or your product documentation.

7 ♦ Under the machine folder where it resides, expand the Unified Broker product folder for the instance you want to start.

8 ♦ Start the Unified Broker instance. For more information on starting and managing a Unified Broker instance, see your product documentation.

For more information on using the Progress Explorer to start and manage NameServer and Unified Broker products, see the Progress Explorer online help.

7.4.2 Starting and Managing a NameServer Using the Progress Explorer

When you start a NameServer instance with the Progress Explorer, a brief status displays for the selected NameServer in the right pane showing that the NameServer is running.

Using the Progress Explorer you can also invoke the following management functions for the running NameServer instance:

- Stop the NameServer.
- Check the operational status of the NameServer.
- View the log file for the NameServer.
- Delete the NameServer instance.

NOTE: Before you can delete a NameServer instance, you must stop the NameServer and have no running Unified Broker instance still reference the NameServer as its controlling NameServer.

For more information on invoking NameServer management functions in the Progress Explorer, see the Progress Explorer online help.

7.4.3 Using the Management Utilities To Start a Unified Broker Instance

Follow these steps to start a Unified Broker instance using the management utilities:

1 ♦ Make sure that the AdminServer process is running on each of the following machines:

- Where the Unified Broker instance you want to start resides
- Where all required NameServers reside
- Where any Progress databases or other DataServers reside that your Unified Broker instance needs to access

2 ♦ If they are not already running, use the NSMAN utility to start each NameServer that you need (more than one if using connection-level fault tolerance) to support your Unified Broker instance. For more information, see the "Starting and Managing a NameServer Using the NSMAN Utility" section.

3 ♦ Start each Progress database or DataServer that your application requires. For more information on starting databases and DataServers, see Chapter 8, "Starting and Running Progress," or your product documentation.

4 ♦ Using the command-line management utility for your Unified Broker product, start the Unified Broker instance. For more information on starting and managing a Unified Broker instance, see your product documentation.

7.4.4 Starting and Managing a NameServer Using the NSMAN Utility

The NSMAN utility runs on both Windows and UNIX platforms. It allows you to invoke the following management functions on a local or remote NameServer instance:

- Start a NameServer.
- Check the operational status of the NameServer.
- Stop the NameServer.

Unlike the Progress Explorer, the NSMAN utility has no mechanism for viewing log files or deleting configured NameServer instances. If you want to view the NameServer log file or delete the NameServer instance, you must do it manually using operating system commands. To delete the NameServer, you must remove the entry for this NameServer instance in the NameServer properties file. For more information on managing log files, see the "Configuring Progress Unified Broker Log Files" section. For more information on accessing the NameServer properties file, see the "Editing the Properties File" section.

NOTE: Before you can delete a NameServer instance, you must stop the NameServer.

This is the syntax to invoke the NSMAN utility:

SYNTAX

```
nsman
  {
      { -name name-server
        { -kill | -start | -stop |-query }
        [ -host host-name -user user-name | -user user-name ]
        [ -port port-number ]
      }
    | -help
  }
```

`-name` *name-server*

This parameter is required. It specifies the name of the NameServer.

`-kill`

Stops and removes the NameServer from memory, no matter what it is doing.

`-start`

Starts the NameServer.

`-stop`

Tells the NameServer to stop itself.

`-query`

Queries the NameServer for its status.

`-host` *`host-name`*

Specifies the name of the machine where the AdminServer is running. If a host name is not specified, it defaults to the local host name.

`-user` *`user-name`*

Specifies a user name and prompts for a password. A user name and password are required only when you use the `-host` parameter and specify a remote host name. If you specify a remote host name with the `-host` parameter but do not specify a user name with the `-user` parameter, you receive a prompt for a user-name and password.

Windows NT supports three different formats for *`user-name`*:

- A user name as a simple text string, such as "`mary`", implies a local user whose user account is defined on the local NT server machine, which is the same machine that runs the AdminServer.
- A user name as an explicit local user name, in which the user account is defined on the same machine that runs the AdminServer except the user name explicitly references the local machine domain, for example "`\mary`".
- A user name as a user account on a specific NT domain. The general format is *`Domain`*`\`*`User`*, in which the *`User`* is a valid user account defined within the domain and the *`Domain`* is any valid NT Server, including the one where the AdminServer is running.

`-port` *port-number*

Specifies the port number of the machine on which the AdminServer is running. If a port number is not specified, it defaults to 20931.

`-help`

Displays command-line help.

For example, the following command starts a NameServer instance named `NSReservations` on a host named `SMPflights` using UDP port 30000 as user `admin`:

```
nsman -name NSReservations -host SMPflights -port 30000 -user admin -start
```

Note that any `nsman` command that specifies a user name typically prompts for the password as well.

7.5 Understanding Load Balancing

Load balancing is a feature that allows client connection requests to be distributed among multiple Unified Broker instances that support the same Application Service. Load balancing is a NameServer option that comes installed with some products (for example, the WebSpeed Enterprise Transaction Server) or that you must install as an option with others (for example, the AppServer). If you have load balancing with your product, the NameServer assigns client connections to the appropriate Unified Broker instances based on weight factors that you specify.

If the weight factor that you specify for each Unified Broker instance is appropriate in relation to the others, the effect is to assign more connections to broker instances with greater resources, and thus to balance connection load among all the instances. You can set the load-balancing weight factor for each Unified Broker instance in the Progress Explorer or by editing the `priorityWeight` property in the `ubroker.properties` file.

Percentage Weight Factors

Properly specified, weight factors give some sense of the amount of work that an individual Unified Broker instance can handle. For example, Table 7–2 shows the effect of weight factors specified for three Unified Broker instances registered to support the same Application Service.

Table 7–2: Weight Factors Based On Percentage

Unified Broker Name	Weight Factor	% Of Time Selected
AS1	20	20
AS2	20	20
AS3	60	60

The selection algorithm used by the NameServer guarantees that AS1 and AS2 are each selected 20% of the time and AS3 is selected 60% of the time. Thus, if the sum of weight factors for all Unified Broker instances that support the same application adds up to 100, each weight factor specifies the exact percentage of time that the NameServer selects the given Unified Broker instance over time.

Arbitrary Sum Weight Factors

You can specify arbitrary weight factors as any sum of values (not necessarily 100), but the weight of each is always proportional to the sum, as shown in Table 7–3.

Table 7–3: Weight Factors Based On Arbitrary Sums

Unified Broker Name	Weight Factor	% Of Time Selected
AS1	2	2/7
AS2	2	2/7
AS3	3	3/7

Fail-over Weight Factor

You can also specify a fail-over weight factor of zero (0) for a Unified Broker instance that you want to accept connection requests when the NameServer finds no other Unified Broker instance available in the pool.

7.6 Understanding Server-level and Connection-level Fault Tolerance

By default, a Unified Broker instance relies on a single controlling NameServer to resolve client connection requests and a single Unified Broker instance to provide services to the client. You can configure the controlling NameServer so that multiple NameServer instances are available to resolve any client connection request, thus providing connection-level fault tolerance. If your product supports load balancing, you can also configure a single NameServer to resolve each connection request using multiple Unified Broker instances that support the same Application Service, thus providing server-level fault tolerance. Figure 7–5 shows the relationship between these configuration options.

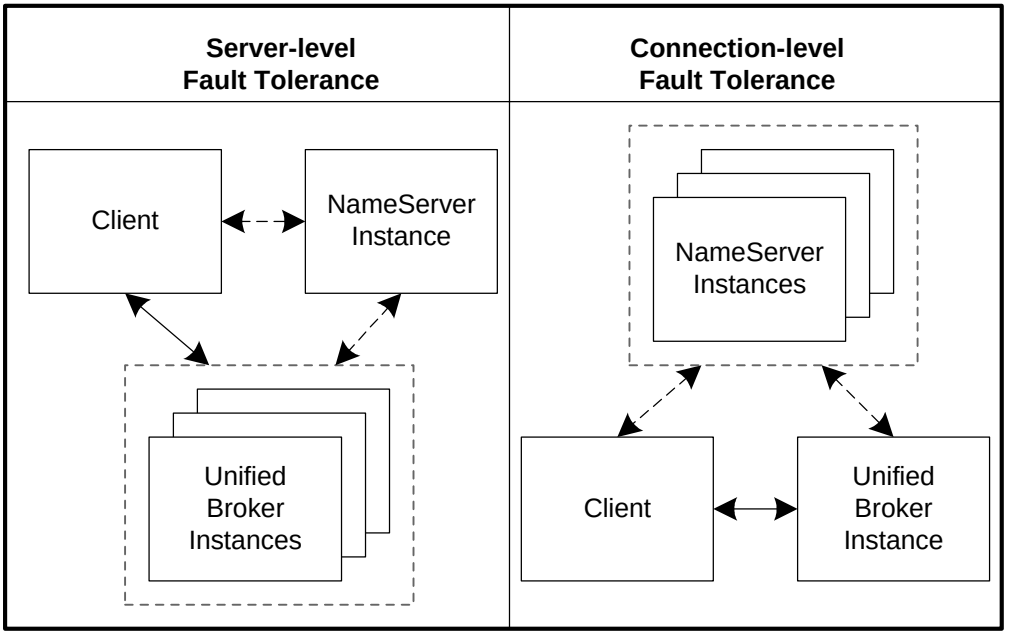


Figure 7–5: Server-level and Connection-level Fault Tolerance

These two levels of fault tolerance operate as follows:

1. **Server-level Fault Tolerance** — *Server-level fault tolerance* allows multiple Unified Broker instances to register with a NameServer for the same Application Service. A client requesting a connection is connected to one of several registered Unified Broker instances that the NameServer determines are available to provide the specified Application Service. If appropriate weight factors are specified, the NameServer also balances connection load among the several broker instances. For more information on load balancing, see the "Understanding Load Balancing" section.

2. **Connection-level Fault Tolerance** — *Connection-level fault tolerance* allows you to configure a collection of NameServers that work together to resolve a client connection request. You can use two different techniques, individually or together, to implement a fault-tolerant NameServer collection. This section describes these techniques and assumes that you are familiar with the documentation on using the NameServer with your Unified Broker product.

You can apply server-level and connection-level fault tolerance individually or together to achieve the level of fault tolerance that your application requires.

7.6.1 Connection-level Fault Tolerance

Connection-level fault tolerance enables a client (AppServer client, SonicMQ Adapter, WebSpeed Messenger, or DataServer client) to have its connection request satisfied by any NameServer from a set of related NameServers. You can configure NameServers for fault-tolerant operation using two different techniques, and you can use these techniques independently or together:

- **NameServer replication** — Where you configure multiple NameServer instances within a single subnet on different machines to listen on the same UDP port. Clients send connection requests and Unified Brokers send registration requests to all NameServer instances using UDP broadcasting. Using UDP broadcasting, the registration information from brokers is replicated on each NameServer that is listening (hence *NameServer replication*). Similarly, each client connection request is sent to each of the replicated NameServers.

- **NameServer neighbors** — Where you configure multiple NameServers on machines located in one or more subnets so that an initial NameServer instance receives the client connection request. If this initial NameServer cannot resolve the request, it passes the request on to a specified list of *NameServer neighbors*. These NameServer neighbors then attempt to resolve the connection request. Each NameServer neighbor represents the controlling NameServer for a separate Unified Broker instance.

Using either or both techniques, a client can get multiple responses from multiple NameServers. The client uses the first response that indicates that the requested Application Service was found. A client only gets a connection error if all NameServers that respond indicate that the Application Service cannot be found.

In general, you can combine NameServer replication with NameServer neighbors to provide connection-level fault tolerance across an entire network. The following sections describe how to implement connection-level fault tolerance using these techniques.

7.6.2 Using UDP Broadcasting

As described earlier, UDP is a connectionless protocol. This feature allows you to configure two types of communications with a NameServer:

- **Host request** — The client or Unified Broker sends a message directly to a NameServer residing on a specific host and listening on a specific port. The IP address represents the actual network location of a specific host. Only the NameServer on the specified host and listening on the specified port receives the message.

- **Broadcast request** — The client or Unified Broker sends a message specifying the UDP broadcast address of the NameServer host and the UDP port number on which the NameServer is listening. The UDP broadcast address represents the entire subnet where a host is located, and you can determine this address using the appropriate operating system commands from any host on the subnet. When a client or Unified Broker sends a UDP broadcast request, every NameServer on any host in the subnet that is listening on the specified port receives the message.

UDP broadcasting insulates the client and Unified Broker from having to know the exact host location of the NameServer. If there is some reason that you need to move the NameServer to a different machine in the same subnet, you can safely do it without having to change your client application or your Unified Broker configuration.

Figure 7–6 shows a client and Unified Broker using UDP broadcasting to communicate with the NameServer. Using the UDP broadcast address, 172.20.255.255, this client and Unified Broker can communicate with a NameServer running on any host in the 172.20 subnet.

Thus, you can use UDP broadcasting to support location transparency for a single NameServer. However, as Figure 7–6 implies, you can also use UDP broadcasting as the basis to support fault-tolerant NameServers using NameServer replication.

7.6.3 Using NameServer Replication

UDP broadcasting supports NameServer replication by allowing a client or Unified Broker request to be received by multiple NameServers listening on the same UDP port and configured on different machines within the same subnet. Because every host on a subnet receives every broadcast request, one or more of these hosts can support a NameServer that receives and handles the same messages. This provides fault tolerance for both a client connection request and a Unified Broker registration request.

Thus, to configure and use replicated NameServers, you must:

- Run each NameServer instance on a separate host located within the same subnet.

- Configure each NameServer instance to listen on the same UDP port.

- Configure each client application to send its connection request and each Unified Broker to send its registration request using the subnet UDP broadcast address instead of the NameServer host address.

There is one broadcast address for each subnet. Using this address and the specified UDP port number, a client or Unified Broker sends a single request that is recognized by every NameServer listening on that port in the subnet.

Figure 7–6 shows a client, a Unified Broker, and two replicated NameServers. The NameServer configurations shown for NameServer NS1 (above the dotted line) appear as they might in the `ubroker.properties` file for each host.

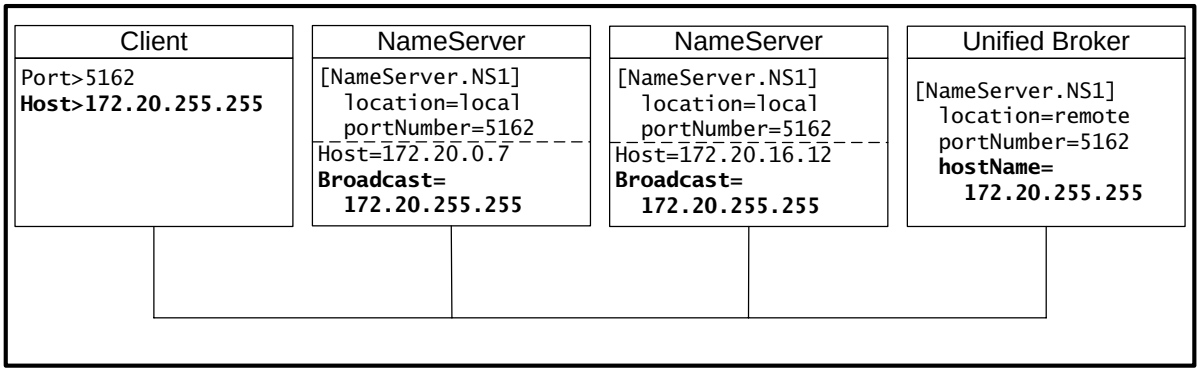


Figure 7–6: NameServer Replication

In Figure 7–6, one NameServer is located on a machine with the IP address 172.20.0.7 and another is located on a machine with the IP address 172.20.16.12. Both NameServers listen on UDP port 5162. The UDP broadcast address for these NameServers is 172.20.255.255. The Unified Broker is configured to register with a controlling NameServer remote from the Unified Broker machine using the UDP broadcast address 172.20.255.255 as the hostName. When the Unified Broker registers with its controlling NameServer using the UDP broadcast, it registers with both replicated NameServers. Similarly, when the client broadcasts its connection request using 172.20.255.255 as the NameServer host name, both replicated NameServers receive the request. The client uses the Unified Broker connection returned by the first NameServer that responds.

Note that if the NameServer at IP address 172.20.0.7 moves to a different host on the subnet, for example, with IP address 172.20.16.5, neither the client application nor the Unified Broker configuration has to change.

Implementing NameServer Replication

To configure and use NameServer replication, follow these steps:

1 ♦ Install the NameServer on each host within a single subnet where you want to replicate a NameServer configuration.

2 ♦ Configure each replicated NameServer to listen on the same UDP port number.

3 ♦ Determine the UDP broadcast address for the subnet where the NameServer hosts reside. For more information, see the "Determining the Broadcast Address" section.

4 ♦ Configure each Unified Broker instance (AppServer, SonicMQ Adapter Broker, WebSpeed Transaction Server, or DataServer) to use a controlling NameServer as follows:

- **Location** — Remote
- **Host name** — The UDP broadcast address that you determined from Step 3
- **Port number** — The UDP port number that you specified in Step 2

5 ♦ Provide connection parameters to the client (AppServer, DataServer, or WebSpeed) that specify the required Application Service name, the broadcast address from Step 3, and the UDP port number that you specified in Step 2.

Determining the Broadcast Address

You can determine the broadcast address of a UNIX machine by using the `netstat` and `ifconfig` commands, as in this example:

```
$ netstat -i
Name  Mtu  Net/Dest   Address    Ipkts   Ierrs Opkts  Oerrs Collis Queue
lo0   8232 loopback    localhost   771334  0     771334 0    0     0
le0   1500 bali        bali       15069970 286170 10019158 1     302211 0
$ ifconfig le0
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
       inet 172.20.0.7 netmask ffff0000 broadcast 172.20.255.255
```

This example shows that the IP address for bali is 172.20.0.7, and its broadcast address is 172.20.255.255.

On NT, you can determine the broadcast address following these steps:

1 ♦ Enter the `ipconfig` command in the console, as shown in this example:

```
C:\>ipconfig

Windows NT IP Configuration

Ethernet adapter CE2XPS1:

        IP Address. . . . . . . . . : 172.18.103.44
        Subnet Mask . . . . . . . . : 255.255.0.0
        Default Gateway . . . . . . : 172.18.0.19
```

2 ♦ For each bit in the Subnet Mask that has a value of 0, convert the corresponding bit in the IP Address to 1.

Note that the IP Address and Subnet Mask are composed of four dot-separated decimal numbers and each decimal number represents an 8-bit binary number. Also note that the decimal number 255 is 11111111 in binary.

In this example, the last two decimal digits of the Subnet Mask are zeros. Since the corresponding bits in the IP Address must be converted to 1, the last two decimal numbers of the IP Address should be 255. Therefore the broadcast address is 172.18.255.255. (For more information on determining broadcast addresses, consult with you network administrator.)

7.6.4 Using NameServer Neighbors

In a typical environment where UDP broadcasting is used, there is at least one NameServer in each subnet where a Unified Broker exists. A client application that wants to make use of Unified Brokers in each subnet can make a separate connection request to the appropriate controlling NameServer for each Unified Broker. However, NameServer neighbors allow the client to make all of its connection requests using a single NameServer address.

NameServer neighbors are alternate NameServers that you specify as part of a NameServer configuration. When a NameServer receives a connection request from a client that it cannot resolve, it automatically passes the request to the specified NameServer neighbors to attempt the resolution.

NameServer neighbors support client applications in a similar way to replicated NameServers in that the client uses the first response returned by a NameServer indicating that the requested Application Service was found. However, unlike replicated NameServers, each NameServer neighbor is typically the controlling NameServer for a separate and distinct Unified Broker configuration that might not support the same Application Services as the others.

You can configure each NameServer neighbor as a NameServer with its own set of NameServer neighbors. Thus, you can link NameServer neighbors to other NameServer neighbors for an arbitrary level of depth. You can also replicate both initial NameServers and NameServer neighbors for maximum fault tolerance.

Follow these steps to configure NameServer neighbors:

1 ♦ Define the NameServer neighbors as local NameServer instances on your network by using the Progress Explorer or by editing the `ubroker.properties` file. These neighbor NameServers can be defined in the same or different subnets in your network.

2 ♦ Specify their names as NameServer neighbors when you define the NameServer in the Progress Explorer, or assign the names in a comma-separated list as the value of the `neighborNameServers` property for the definition in the `ubroker.properties` file.

Figure 7–7 shows how you can do this. This example combines NameServer replication with NameServer neighbors.

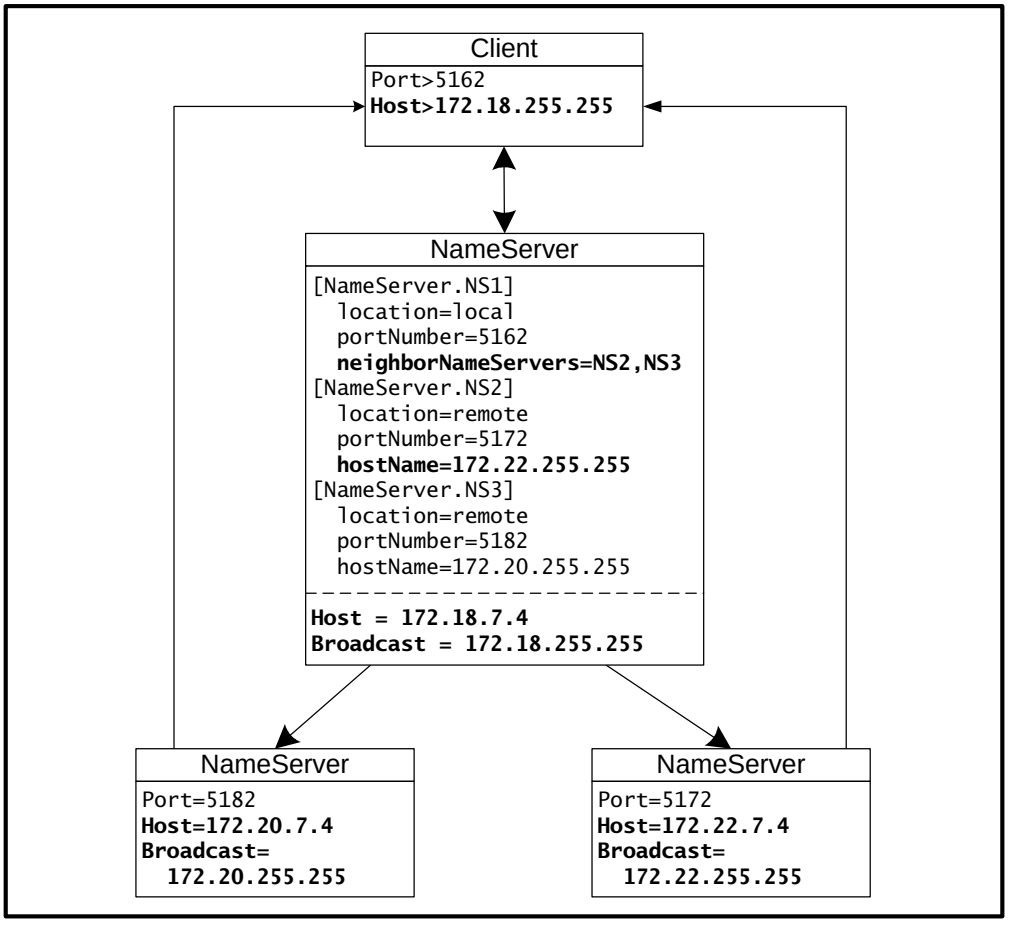


Figure 7–7: NameServer Neighbors

In Figure 7–7, if a client requests a connection to a Unified Broker that supports the Inventory Application Service using the 172.18.255.255 broadcast address, the NameServer at host 172.18.7.4 receives the request. If this NameServer has a Unified Broker that registered for the Inventory Application Service, it returns the location of that Unified Broker back to the client. If it does not have a Unified Broker that registered for the Inventory Application Service, this NameServer forwards the request to its neighbors, specified using the broadcast addresses 172.20.255.255 and 172.22.255.255. In this instance, the NameServers at hosts 172.20.7.4 and 172.22.7.4 receive the request.

NOTE: If you replicate these NameServers, all of the replicated NameServers in each subnet receive the request.

Neither of these NameServers have neighbors of their own, so both of them send a response back to the client. It does not matter if one of the NameServers does not know about the requested Application Service. The client uses the first positive acknowledgement from a NameServer and disregards the rest. The client application only receives an indication that the Application Service was not found if all responding NameServers indicate that the Application Service was not found.

Note that while NameServer neighbors provide the most benefit when using UDP broadcast, there is no requirement to do so. The hostName properties for NS2 and NS3 in Figure 7–7 can explicitly specify 172.20.7.4 and 172.22.7.4. You might want to use NameServer neighbors without broadcasting when you must tie together preconfigured NameServers, but where the performance implications of broadcasting out weigh the benefits. For more information, see the "Performance Implications Of Broadcasting" section.

7.6.5 Performance Implications Of Broadcasting

When you use UDP to send a specific host request, only the specified host examines the message to determine what port it was sent to and whether an application (like the NameServer) is running and listening to that port on the host. However, when you use UDP broadcasting, either for NameServer replication or only to provide location transparency for a single NameServer, every host in the specified subnet examines the message for this same information.

Thus, using UDP broadcast might have a significant impact on the performance of your network if you have a large number of client applications that frequently connect to Unified Brokers. In deciding whether to use UDP broadcasting, you must weigh the benefits of location transparency for a single NameServer or replication of multiple NameServers against the impact on your Unified Broker and network performance.

8

Starting and Running Progress

This chapter describes how to start up and connect to a Progress database.

It contains the following sections:

- Starting Progress
- Running Progress Clients and Servers On a Network

8.1 Starting Progress

Progress startup commands differ with certain operating systems, user interfaces, and network software. In Windows, you can use the Progress Explorer to start a server. See the "Using the Progress Explorer Tool" section in Chapter 4, "Administration Utilities," for information on launching the Progress Explorer. See the Progress Explorer online help for instructions on connecting to an AdminServer.

If you are an international customer, you can set code pages for different application components at startup. You can also set numerical and date/time formats at startup by specifying internationalization parameters. See the *Progress Internationalization Guide* and the *Progress Startup Command and Parameter Reference* for more information on using internationalization parameters at startup.

8.1.1 Startup and Shutdown

You can use either the Client or the Progress Explorer to perform many of the startup and shutdown tasks. These methods provide you with a GUI interface for managing and configuring your databases and servers. If you are not using a Windows environment, or if you prefer a command-line interface, you can choose to enter commands at the command line to perform these tasks. The following sections explain how to use the GUI and command-line interfaces to perform startup and shutdown tasks.

Using the GUI Interface

You can use either the Client or Progress Explorer to perform startup and shutdown tasks, as indicated in Table 8–1. To perform one of the tasks listed using the Client, open the properties of the Client and modify the shortcut target as indicated. See Appendix A, "The Progress Program Group," for more information about modifying icon properties. The Client icon looks like this:



To perform one of the tasks using the Progress Explorer, start the Progress Explorer, select the server you want to start or stop, and follow the instructions in the online help. The Progress Explorer Tool icon looks like this:



NOTE: These instructions refer to the 4GL Client. To perform SQL tasks, you must start the SQL Explorer and use the SQL Client. See the SQL Explorer online help for more information about using the SQL Client.

Table 8–1 summarizes tasks and methods to perform startup and shutdown tasks using the graphical user interface (GUI).

Table 8–1: Windows GUI Startup and Shutdown Commands *(1 of 2)*

Task	Progress Program Group Icon	Action
Start the Progress Procedure Editor and connect to a single-user database.	Client	Modify shortcut target properties: *installation-path*\prowin32.exe *pathname**db-name* -1
Start the Progress Procedure Editor and connect to a multi-user database.	Client	Modify shortcut target properties: *installation-path*\prowin32.exe *pathname**db-name*
Start the ADE Desktop and connect to a single-user database.	Client	Modify shortcut target properties: *installation-path*\prowin32.exe -p _desk.p *pathname**db-name* -1
Start the ADE Desktop and connect to a multi-user database.	Client	Modify shortcut target properties: *installation-path*\prowin32.exe -p _desk.p *pathname**db-name*
Start a Progress batch session and connect to a single-user database.	Client	Modify shortcut target properties: *installation-path*\prowin32.exe -b *pathname**db-name* -1 -p *procedure*
Start a Progress batch session and connect to a multi-user database.	Client	Modify shortcut target properties: *installation-path*\prowin32.exe -b *pathname**db-name* -p *procedure*

Table 8–1: Windows GUI Startup and Shutdown Commands *(2 of 2)*

Task	Progress Program Group Icon	Action
Start a server or broker for a Progress database.	Progress Explorer	See online help. Command-line alternative: `proserve` *`pathname\db-name`*
Shut down a server or broker for a Progress database.	Progress Explorer	See online help. Command-line alternative: `proshut` *`pathname\db-name`*

Using the Command-line Interface

Startup commands start a Progress session and connect you to a database. The following tables summarize the startup commands for each operating system and their functions. For detailed information on these commands and their parameters, see the descriptions of the commands in the *Progress Startup Command and Parameter Reference*.

Table 8–2 summarizes the tasks to perform and the related startup and shutdown commands to use on Windows systems.

Table 8–2: Windows Startup and Shutdown Commands *(1 of 2)*

Task	Command
Start a Windows character Progress Procedure Editor and connect to a single-user database.	`pro` *`db-name`*
Start a Windows character Progress Procedure Editor and connect to a multi-user database.	`mpro` *`db-name`*
Start a Windows Progress character client session in batch mode and connect to a single-user database.	`bpro` *`db-name`* `-p` *`procedure-name`*
Start a Windows Progress character client session in batch mode and connect to a multi-user database.	`mbpro` *`db-name`* `-p` *`procedure-name`*

Table 8–2: Windows Startup and Shutdown Commands *(2 of 2)*

Task	Command
Start a Progress server-group.	`proserve -servergroup` *server-group-name*
Start a server or broker for a multi-user Progress database.	`proserve` *db-name* `-S` *service-name* `-H` *host-name* `-N` *network-type*
Shut down a multi-user server or broker for a Progress database.	`proshut` *db-name*
Start a remote Progress DataServer broker.	`probrkr -S` *service-name* `-H` *host-name* `-N` *network-type*
Start an asynchronous page writer (APW) for a database[1].	`proapw` *db-name*
Start a before-image writer (BIW)[1].	`probiw` *db-name*
Start an after-image writer (AIW)[1].	`proaiw` *db-name*
Start the Progress Watchdog utility[1].	`prowdog` *db-name*
Shut down a remote Progress DataServer.	`proshut` *db-name* `-S` *service-name* `-H` *host-name* `-N` *network-type*
Shut down an APW, AIW, BIW, or Watchdog process[1].	`proshut` *db-name* Choose option 1 (Disconnect a User) to disconnect the process.

[1] Option available only on Enterprise product.

8.1.2 Starting Progress As a Windows NT Service

To run Progress as a Windows NT service, you must start ProService before starting a Progress session. To accomplish this, you can either start the Progress Explorer and connect to an AdminServer or use ProControl or the PCCmd command to start ProService.

ProControl is a graphical administration tool that you can use to manage ProService. *PCCmd* is the command-line utility that you can use to perform many of the same tasks as the graphical tools.

NOTE: ProService is run as an NT service. This means it runs under the system account. It does not run under the account the user is currently logged into. You must grant system access to the directory containing the database for ProService to work properly.

Using the Progress Explorer To Connect To the AdminServer

Follow these steps to connect to the AdminServer with the Progress Explorer:

1 ♦ From the **Start** menu choose **Progress→ Progress Explorer Tool**, or from the Progress Program Group, double-click the Progress Explorer Tool icon. The Progress Explorer Tool icon looks like this:



2 ♦ Start the Progress Explorer and establish a connection to an AdminServer. See Chapter 4, "Administration Utilities," for more information about Progress Explorer, and see the Progress Explorer online help for detailed instructions on connecting to an AdminServer.

Using ProControl To Start ProService

Follow these steps to start ProService using ProControl:

1 ♦ From the **Start** menu choose **Progress→ProControl**; or from the Progress Program Group, double-click the ProControl icon. The ProControl icon looks like this:



2 ♦ Select the Windows host machine where you want to run ProControl, and start ProService. See Chapter 4, "Administration Utilities," for detailed instructions on starting ProService with ProControl.

Using the PCCmd Command To Start ProService

To start ProService with the PCCmd command on the current host, type the following command at the command line or within a .BAT file:

```
PCCmd ProService Start
```

If you use PCCmd within a .BAT file, you can test the ERRORLEVEL to determine success or failure. If ERRORLEVEL is 1, the command failed. If ERRORLEVEL is 0, the command succeeded.

Here is a simple example:

```
If %errorlevel% = = 1 then Echo Passed.
If %errorlevel% = = 0 then Echo Failed.
```

8.1.3 Starting Single-user Progress In Interactive Mode

To start single-user Progress, enter the following commands:

```
prowin32 [ db-name ] -1 [ parameters ]
```

db-name

Specifies the database you want to start. (-db is implicit but can be specified)

parameters

Specifies the startup parameters you want to use to describe system and application characteristics. For a detailed description of the Progress startup parameters, see the *Progress Startup Command and Parameter Reference*, the Progress DataServer Guides (*Progress DataServer for ODBC Guide*, *Progress DataServer for ORACLE Guide*, and *Progress/400 Product Guide*), and the *Progress Client Deployment Guide*.

8.1.4 Starting Single-user Progress In Batch Or Background Mode

Batch or background processing is convenient for large-scale database updates or procedures that you can run unattended (at night, for example).

To start single-user Progress in batch or background mode, enter the following command:

```
prowin32 [ db-name ] -1 -b -p procedure-name [ parameters ]
```

db-name

Specifies the database you want to start.

-b

Specifies that Progress should run in batch mode.

-p *procedure-name*

Specifies the procedure to run at startup.

parameters

Specifies the startup parameters you want to use.

output-file

Specifies the name of the file that receives all output to the default stream.

Using ProControl To Start Single-user Progress In Batch Or Background Mode

You can use the Procontrol utility to create a task that can be started as single-user Progress in batch or background mode. See Appendix B, "The ProControl Utility," for detailed instructions on this topic.

Redirecting Output

You can redirect batch job input and output with the greater than (>) and less than (<) redirection symbols.

Example

```
prowin32 [ db-name ] -1 -b -p sportsbat.p > errlist
```

In this example, Progress starts in batch or background mode against the sports database and automatically runs the sportsbat startup procedure. In addition, the system directs output not otherwise directed with an OUTPUT TO statement to the file named errlist.

8.1.5 Starting the Multi-user Server Or Broker

Before you can run multi-user Progress, you must start the multi-user server process. The server process coordinates all the database requests from all the users using a single database. You can use the Progress Explorer to start the multi-user server process, or you can use the command-line interface. The following sections describe these methods of starting the multi-user server process.

Using the Progress Explorer To Start the Multi-user Server Process

You can use the Progress Explorer to start a multi-user server process.

Start the Progress Explorer and establish a connection to one or more AdminServers. See Chapter 4, "Administration Utilities," for more information about starting Progress Explorer, and see the Progress Explorer online help for detailed instructions on using Progress Explorer to start a multi-user server process.

Using the Command Line Interface To Start the Multi-user Server Process

You can enter the following command at the command line to start the multi-user server process:

```
proserve db-name [ parameters ]
```

db-name

Specifies the database you want to start Progress against. (-db is implicit)

parameters

Specifies the startup parameters for the broker/server. See the *Progress Startup Command and Parameter Reference* and the Progress DataServer Guides (*Progress DataServer for ODBC Guide*, *Progress DataServer for ORACLE Guide*, and *Progress/400 Product Guide*) for a list of broker/server startup parameters.

8.1.6 Starting the Multi-user Server Or Broker As a Windows NT Service

Before you can run multi-user Progress as a Windows NT service, you must create an entry in the registry to enable Progress to run as a Windows NT service. You can create an entry in the registry with either the Progress Explorer or ProControl. The following sections describe these methods.

Using the Progress Explorer To Start the Multi-user Server Or Broker

Start the Progress Explorer and establish a connection to one or more AdminServers. See Chapter 4, "Administration Utilities," for more information about starting Progress Explorer, and see the Progress Explorer online help for detailed instructions on using Progress Explorer to start a multi-user server process.

Using ProControl To Start the Multi-user Server Or Broker

You can create a database entry within ProControl to run multi-user Progress as a Windows NT service. When you create a database entry, ProControl automatically creates an entry in the registry, which enables Progress to run as a Windows NT service. The Progress server process coordinates the database requests from all the users accessing a single database.

See Appendix B, "The ProControl Utility," for detailed instructions on how to create the ProControl database entry.

8.2 Running Progress Clients and Servers On a Network

After setting up your database on the network, you are ready to run Progress. In general, the procedure for running clients and servers on a network of systems is very similar to the procedure for running them on a single system. First, you must start all required database servers and application servers, then start the client sessions that connect to them.

8.2.1 Using Network Startup Parameters

To connect network clients, servers, and application servers, you might have to use a variety of startup parameters to establish and manage network communications among them. The requirements and use of these parameters vary on different operating systems and network environments.

Client Network Parameters

Table 8–3 lists the parameters you use to supply Progress clients with necessary network information.

Table 8–3: Client Network Parameters

Parameter	Syntax
Host Name[1]	-H
Message Buffer Size	-Mm
Network Type	-N
Network Version	-Nv
Service Name	-S

[1] For the TCP network type, this required parameter specifies the machine name (address) where the server runs.

Server Network Parameters

Table 8–4 lists the parameters you use to supply Progress brokers and servers with necessary network information. In a Progress AppServer configuration, use the same parameters to pass information to application brokers and application servers.

Table 8–4: Server Network Parameters

Parameter	Syntax
Host Name	-H
Manual Server	-m2
Secondary Login Broker	-m3
Maximum Clients per Server	-Ma
Minimum Clients per Server	-Mi
Maximum Dynamic Server	-maxport
Minimum Dynamic Server	-minport
Message Buffer Size	-Mm
AppServer Maximum Maintained Prestart Counter	-Mms
Maximum Servers	-Mn
Maximum Servers per Protocol	-Mp
Maximum Servers per Broker	-Mpb
AppServer Maximum Prestart Counter	-Ms
Network Type	-N
Service Name	-S

For more information on the syntax and values for each parameter, see the Progress DataServer Guides (*Progress DataServer for ODBC Guide*, *Progress DataServer for ORACLE Guide*, and *Progress/400 Product Guide*) and the *Progress Client Deployment Guide*.

8.2.2 Specifying the Network Type (-N)

Each Progress executable has a default network type determined by the operating system on which it runs. Table 8–5 lists the default network type that Progress uses on each supported operating system.

Table 8–5: Default Network Types

Operating System (Executable)	Default Network Type
Windows (client or server)	TCP
UNIX (client or server)	TCP
Progress/400 (server)	SNA

8.2.3 Network Addressing With Progress (-S and -H)

In all network environments, you use the Service Name (-S) startup parameter to assign a name to a Progress broker/server. You then address this broker/server from a remote client by using the same value for -S as a startup or database connection parameter. Depending on your network type, you might also have to specify additional addressing criteria for remote Progress clients. In terms of Progress addressing, the TCP protocol uses host addressing.

The TCP protocol requires a remote client to explicitly address the database server machine (or host) on which the server runs. In a TCP network, you must use the Host Name (-H) startup parameter to specify the host address. The -H value is the name assigned to the database server machine in your TCP/IP hosts file.

8.2.4 Starting Applications On a Network

In general, you can start a Progress application on a network by following these steps:

1 ♦ Start each broker or server on its database server machine or AppServer machine.

2 ♦ Start the client applications on the application workstations.

Starting Network Brokers and Servers

You can start most network brokers and servers using either the Progress Explorer (available for Windows systems only) or the PROSERVE command for your database server machine. To use the Progress Explorer, double-click the Progress Explorer icon and follow the directions in the online help for starting brokers and servers. See Chapter 4, "Administration Utilities," for more information about starting the Progress Explorer.

Alternatively, on Windows and UNIX systems you can enter the following command to start brokers for two databases (sports and news) using the TCP network type:

```
proserve sports -S spsrv -H localhost -N TCP -db news -S nwsrv
-H localhost -N TCP
```

Starting TCP/IP Clients On Windows

You can start most network clients using the `MPRO` command for your application workstation. You can do this either by modifying the Client properties or by entering a command at the command line.

To modify the Client icon, display the properties and modify the shortcut target. See Appendix A, "The Progress Program Group," for more information on modifying icon properties. Modify the shortcut target with the following parameters to start a client application named `spapp.p` for two databases (sports and news) managed on a host named `dbmach` using the TCP network type:

```
prowin32.exe sports -p spapp.p -S sportssv -H dbmach -N TCP
     -db news -S newssv -H dbmach -N TCP
```

To use the command line on Windows systems, enter the following command to start a client application named `spapp.p` for two databases (sports and news) managed on a host named `dbmach` using the TCP network type:

```
prowin32 sports -p spapp.p -S sportssv -H dbmach -N TCP
     -db news -S newssv -H dbmach -N TCP
```

8.2.5 Starting Multiple Brokers Using the Same Protocol

You can use either the Progress Explorer or the command-line interface to start multiple brokers that use the same protocol. The `-Mn` parameter and a new parameter, Maximum Servers per Broker (`-Mpb`), determine the number of servers a broker can start. You can use the Progress Explorer to manage and configure server groups.

Using the Progress Explorer to Start Multiple Brokers

You can use the Progress Explorer to start multiple brokers that use the same protocol. Start the Progress Explorer by double-clicking the Progress Explorer Tool icon in the Progress program group. Follow the instructions in the Progress Explorer online help to start and configure brokers. See Chapter 4, "Administration Utilities," for more information about starting the Progress Explorer.

Using the Command-line Interface To Start Multiple Brokers

Use the following commands to start two brokers that use TCP and start multiple servers each:

```
proserve db-name -S server-name -N network-type -H host-name -Mnn -Mpbn
proserve db-name -S server-name -N network-type -H host-name -Mpbn -m3
```

db-name

Specifies the database you want to start. If the database is not in the current directory, you must specify the full pathname of the database.

-S *service-name*

Specifies the database server or broker process service name. You must specify the service name in a TCP network.

-N *network-type*

Specifies the network protocol, which is TCP.

-H *host-name*

Specifies the machine where the database server runs.

-Mn *n*

Specifies the maximum number of remote client servers and login brokers that the broker process can start.

-Mpb *n*

Specifies the number of servers that the login broker can start to serve remote users. This applies to the login broker that is being started.

-m3

Starts the secondary login broker.

For example, you would use the following commands to start two brokers that use TCP and start four servers each:

```
proserve db -S demosv1 -N tcp -H myhost -Mn 9 -Mpb 4
proserve db -S demosv2 -N tcp -H myhost -Mpb 4 -m3
```

As the example shows, the -Mn value must be large enough to account for each additional broker and all servers. If you do not specify -Mpb, the value of -Mn becomes the default.

You must include the -m3 parameter with every secondary broker startup command. While the -Mpb sets the number of servers a broker can start, the -m3 parameter actually starts the secondary broker.

If you start multiple brokers, you should also run the Progress Watchdog process (PROWDOG). PROWDOG enables you to restart a dead secondary broker without shutting down the database server.

8.2.6 Accessing a Server Behind a Firewall

Progress allows you to use the Minimum Dynamic Server Port (-minport) and the Maximum Dynamic Server Port (-maxport) server startup parameters to provide client access to a Progress server that is behind a firewall. This communication is possible only when the access to the server can be limited. You supply this limit when you specify a group of port numbers with the -minport and -maxport parameters.

For example, suppose you start the following two login brokers:

```
proserve db -S demosv1 -N tcp -H myhost -minport 4000 -maxport 4040
proserve db -S demosv2 -N tcp -H myhost -minport 4041 -maxport 4080 -m3
```

A client requesting a connection from the first broker, demosv1, is assigned a port number in the range of 4000 to 4040. The 4000-to-4040 range limits access to the server by limiting communication to just 40 ports.

The default for -minport is 1025 for all platforms. Ports lower than 1025 are usually reserved for system TCP and UDP. The default for -maxport is 2000 for all platforms. Remember that some operating systems choose transient client ports in the 32768-to-65535 range. Choosing a port in this range might produce unwanted results.

8.2.7 Starting and Running Multi-user Progress In Interactive Mode

Enter one of the following commands to start and run Progress in interactive mode:

```
prowin32 [ db-name ] [ parameters ]
```

db-name

Specifies the database you want to start. If the database is not in the current directory, you must specify the full pathname of the database.

parameters

Specifies the startup parameters you want to use.

The database you name when starting multi-user Progress must be in the current directory, or you must specify the full pathname of the database. For example, if you are using UNIX and you log in as sue, the login directory is `/usr/sue`.

8.2.8 Starting and Running Multi-user Progress Clients In Batch Or Background Mode

Before you can start a multi-user Progress batch or background job, you must start the server for the database you want to use. You can start the server either by modifying the Client icon properties or by typing a command at the command line.

Using the Client To Start Multi-user Progress In Batch Or Background Mode

You can modify the Client properties to start multi-user Progress in batch or background mode. To modify the Client icon, display the properties and modify the shortcut target with the following parameters:

```
prowin32 [ database name ] -N network-type
-S service-name -p procedure-name
   [ parameters ]
```

See Appendix A, "The Progress Program Group," for more information about modifying icon properties.

Using the Command-line Interface To Start Multi-user Progress In Batch Or Background Mode

To use the command-line interface to start multi-user Progress in batch or background mode, enter the following command:

```
prowin32 [ database name ] -N network-type
  -S service-name -p procedure-name
   [ parameters ]
```

db-name

Specifies the database you want to start.

-p *procedure-name*

Specifies the procedure to run at startup.

parameters

Specifies the startup parameters you want to use.

error-file

Specifies the file where error messages are sent.

Using Progress Explorer To Start Multi-user Progress In Batch Or Background Mode

You can use the Progress Explorer to create a task that starts multi-user Progress in batch or background mode on a Windows system. See Chapter 4, "Administration Utilities," for information on starting the Progress Explorer. See the Progress Explorer online help for instructions on starting Windows multi-user Progress in batch or background mode.

Using ProControl To Start Multi-user Progress In Batch Or Background Mode

You can use the ProControl utility to create a task that starts multi-user Progress in batch or background mode on a Windows system. See Appendix B, "The ProControl Utility," for detailed instructions on this topic.

9

Progress Event Logging

Progress logs significant database events such as Progress startup parameter settings, startup error messages, shutdown messages, system error messages, and application-related events.

This chapter contains the following sections:

- The Progress Event Log File
- Managing the Progress Event Log File Size
- Event Logging On Windows NT

9.1 The Progress Event Log File

The Progress event log is a text file that contains a history of significant database events, such as Progress startup parameter settings and startup, shutdown, and system error messages. This file has a `.lg` extension.

9.2 Managing the Progress Event Log File Size

The event log (LG) file expands as you use the database. If it becomes too large, you can reduce its size by removing old log entries. To remove log entries from an LG file, use the Progress Log Maintenance (PROLOG) utility or a text editor. Do not remove entries from the log file while the database is in use.

Enter the following command to remove entries from an event log file:

```
prolog database-name
```

The PROLOG utility removes all but the most recent entries from the log file. For more details, see the description of the PROLOG utility in the *Progress Database Administration Guide and Reference*.

9.3 Event Logging On Windows NT

In addition to the Progress event log, the Progress NT Server writes events to the NT Event Log. The *NT Event Log* is the object that enables Windows NT users to view the status of application, security, and system processes, and to view their associated events. Progress is an application process and, as such, it writes Progress events to the Application Event Log. You use the NT Event Viewer to see the Event Log's contents. You can customize the Event Viewer so that it displays only the event types that you want to view. You access the Event Viewer through the Administrative Tools program group.

The components that enable the Progress service to log messages to the NT Application event log database are described in Table 9–1.

Table 9–1: Progress Event Logging Components

Component	Function
NT Event Viewer	The standard NT front-end that enables users to view the NT Event Log.
NT Event Log	The standard Windows NT database that records event information.
CATEGORY.DLL	The Progress resource that contains the 14 categories into which Progress messages might fall.
PROMSGS file	The Progress object that contains a single language version of the Progress messages. Progress supplies a PROMSGS file for each supported language version of Progress. The PROMSGS file is installed to the DLC directory. See Appendix D, "Progress National Language Support," and the *Progress Internationalization Guide* for more information on the PROMSGS file.

9.3.1 Managing Progress Events On Windows NT

You can define the level of event logging that you want your Progress application to support by using either the Event Level Environment Variable (EVTLEVEL) or the Event Level startup parameter (`-evtlevel`). Use the ProControl utility to supply the Event Level as a startup parameter or environment variable. See Chapter 4, "Administration Utilities," and Appendix B, "The ProControl Utility," for information about the ProControl utility.

The valid Event Level values are described in Table 9–2.

Table 9–2: Event Level Values *(1 of 2)*

Value	Description
None	No Progress events are written to the event log.
Brief	Progress messages defined as Error and Warning messages are written to the event log.

Table 9–2: Event Level Values *(2 of 2)*

Value	Description
Normal	Progress messages defined as Error and Warning messages are written to the event log. In addition, any Progress message that is normally written to the log file (`.lg`) is also written to the Event Log. This is the default.
Full	Every message generated by Progress is written to the Event Log. Any Progress messages generated using the Message Statements are also written to the log file.

9.3.2 Understanding the Windows NT Application Event Log Components

The components of the Windows NT Application Event Log are standards defined by Windows NT. Figure 9–1 illustrates the Windows NT Application Event Log components when shown through the Event Viewer.

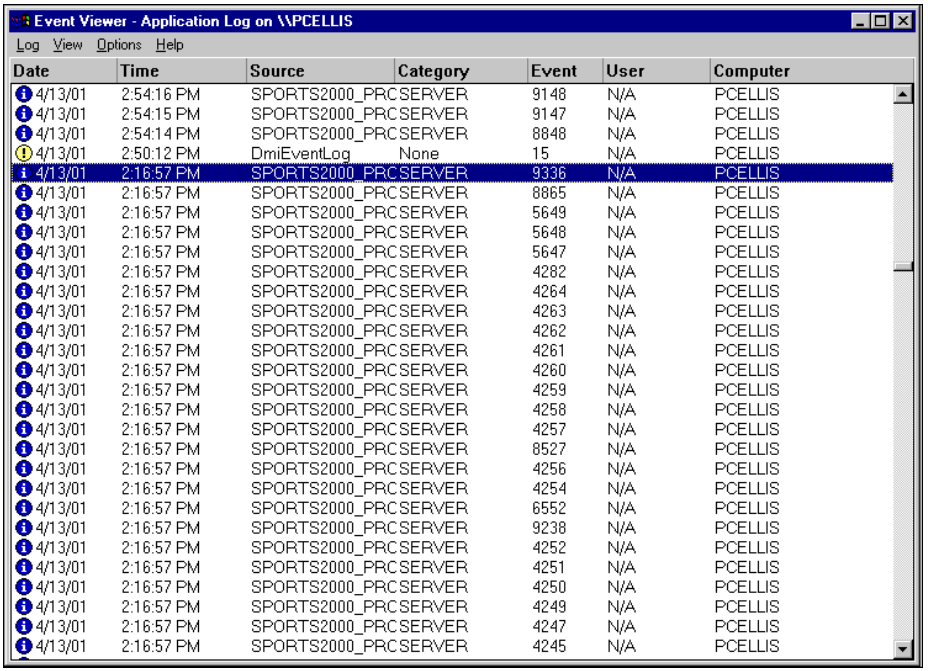


Figure 9–1: Windows NT Application Event Log Components

Table 9–3 describes how Progress uses the NT Application Event Log columns.

Table 9–3: Windows NT Application Event Log Components

Log Component	Log Information
Date	Date the event occurred.
Time	Time the event occurred.
Source	Source of the event. This is the name of the connected Progress database, if a database is connected. If no database is connected, then "Progress" is listed. If you are using the Progress AppServer, "Progress" is also the default source for Progress AppServer messages; however, you can override the default source name by specifying the `-logname` application broker startup parameter.
Category	To help you isolate the cause of the message displayed in the Event Log, Progress supports 14 event categories. The event categories are: AIW, APW, BACKUP, BIW, DATASERVER, MON, OIBRKR, OIDRVR, Progress, RFUTIL, SERVER, SHUT, USER, and PROWDOG. When no database is connected, Progress is specified as the category. All categories reside in a file called `category.dll`. These categories correspond to the existing categories of events that are displayed in the `progress.lg` file (application broker and application server events are displayed in the AppServer log file, `proapsv.lg`). (Note that DATASERVER is not included as a category in the standard `progress.lg` file.)
Event	Associates to the Progress message that was generated. These are the same message numbers that are displayed in the standard database .lg file.
User	Identifies the user logged in to the Windows NT workstation where the event occurred.
Computer	Identifies the name of the Windows NT workstation where the event occurred. The NT Event Viewer enables you to get more information about events by double-clicking on any event.

You can view additional information about an event by double-clicking on it. Windows NT displays the Event Detail screen, as shown in Figure 9–2.

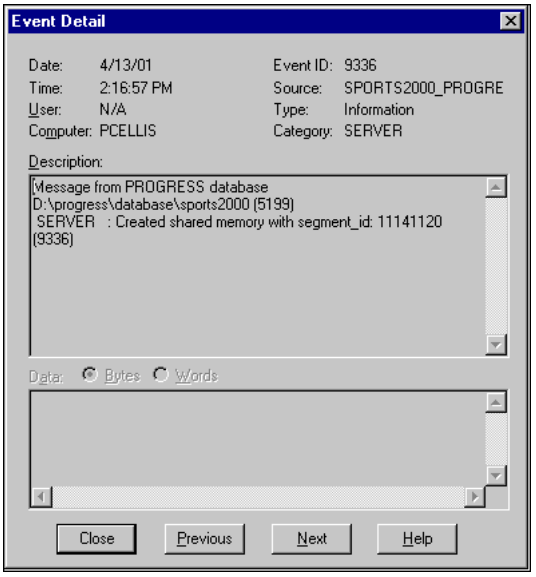


Figure 9–2: Windows NT Application Event Detail Dialog Box

9.3.3 Windows NT Event Log and Registry

Windows NT requires applications that use the Event Log to be bound to all of the necessary components. For Progress this means that the `PROMSGS.DLL` and the `CATEGORY.DLL` must be bound to any Progress database. Progress stores this information in the registry. Progress makes the registry entries and performs any binding operations that are necessary when you initially access a database. When Progress binds the DLL files to the database, it writes the fully qualified pathname to the registry. If you delete the database, you must manually remove the associated data from the registry. If you move the location of the DLLs after you access the database, you must manually edit the registry data. The Progress components can be found in the following location in the registry:

```
HKEY_LOCAL_MACHINE
SYSTEM
  CurrentControlSet
     Services
        EventLog
          Security
          System
          Application
             PROGRESS
             <Database Name>
```

See the Microsoft Windows NT documentation for more information about editing registry files.

When Progress tries to find the DLLs before this information is included in the registry, it performs the search according to these rules:

1. Progress searches the current directory.

2. If the DLL is not in the current directory, Progress searches the directory where the Progress executable is located.

3. If the DLL is not in the same directory as the Progress executable, Progress searches the user's path.

If the DLL is not in the user's path, Progress generates a message stating that the DLL cannot be found, and it writes a message to the Progress log file.

A

The Progress Program Group

The installation process creates a program group called Progress. The group contains icons for the Progress server, Progress clients, and various Progress utilities. This appendix describes the individual icons in the Progress program group and gives instructions on editing icon properties.

This appendix contains the following section:

- Changing Program Item Properties

A.1 Changing Program Item Properties

The installation program automatically creates a Progress program group comprised of program items. You can edit the icons to change item properties like startup parameters and working directories.

To change the properties of an item in Windows or Citrix MetaFrame, follow these steps:

1 ♦ From the desktop, right-click the **Start** button.

2 ♦ From the pop-up menu, choose one of the following:

- **Open All Users** (Windows NT and Windows 2000 operating systems)
- **Open** (Windows 98 and Citrix operating systems)

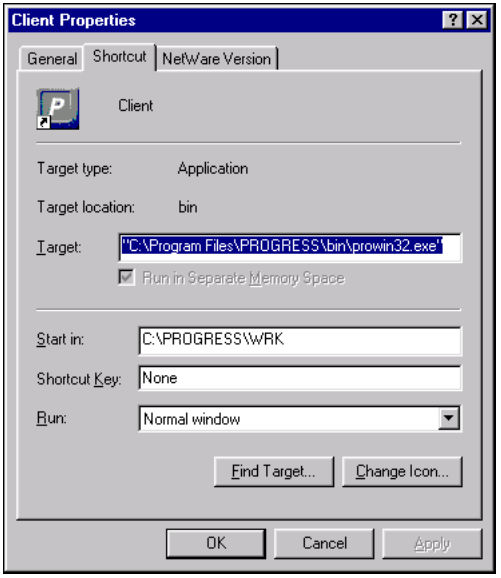
A dialog box that lists all programs in the **Start** menu appears.

3 ♦ Double-click **Programs**, then double-click **Progress**.

4 ♦ Select a Progress program item, and right-click.

5 ♦ Choose **Properties** for the pop-up menu.

6 ♦ Select the **Shortcut** tab. The shortcut properties appear:



7 ♦ Change the **Target** and **Start in** fields to include the startup parameters and working directory appropriate for your installation.

8 ♦ Choose **OK**.

Table A–1 shows the Progress program group components, icon names, and the command you use to start each component. Note that the installation program automatically includes these commands in your installation path. Note also that \DLC corresponds to your Progress installation path.

Table A–1: Progress Program Group Icons *(1 of 2)*

Component Icon Name	Properties
4GL Batch Engine	*installation-path*\bin\prowin32.exe -b
Add Components	*installation-path*\setup.exe -addcomp
AppBuilder	*installation-path*\bin\prowin32.exe -p _ab.p
Application Compiler	*installation-path*\bin\prowin32.exe -p _comp.p
Character Client	*installation-path*\bin_progres.exe
Client	*installation-path*\bin\prowin32.exe
Config	*installation-path*\bin\showcfg.exe
Data Administration	*installation-path*\bin\prowin32.exe -p _admin.p
Data Dictionary	*installation-path*\bin\prowin32.exe -p _dict.p
Desktop	*installation-path*\bin\prowin32.exe -p _desk.p
Help	*windir*\winhelp32.exe *installation-path*\prohelp\techeng.hlp
License Update	*installation-path*\setup.exe -u
Open Interface Broker	*installation-path*\bin\oibrkr32.exe
Probuild	*installation-path*\bin\prowin32.exe *installation-path*\probuild\eucapp\euc.db -RO -p euc.p
ProControl	*installation-path*\bin\proctrl.exe

Table A–1: Progress Program Group Icons *(2 of 2)*

Component Icon Name	Properties
Proenv	*installation-path*\bin\proenv.bat
Progress Explorer Tool	%DLC%\bin\jvmStartw.exe -f %DLC%\proexp.prm
ProxyGen	%DLC%\bin\jvmStartw.exe -f %DLC%\proxygen.prm
Release Notes	write.exe *installation-path*\readme.wri
Report Builder	*installation-path*\bin\prorb32.exe
Results	*installation-path*\bin\prowin32.exe -p results.p
Run-time Client	*installation-path*\bin\prowin32.exe -rr
SpeedStart	*work-directory-path*\sstart.htm
SQL Explorer Tool	%DLC%\bin\jvmStartw.exe -f %DLC%\sqlexp.prm
Translation Manager	*installation-path*\bin\prowin32.exe -p _tran.p
Visual Translator	*installation-path*\bin\prowin32.exe -p _vtran.p
Uninstall	*windir*\IsUninst.exe -fc:\win95\ProgressUninstall9.1D -c "E:\Program Files\PROGRESS\uninst.dll

B

The ProControl Utility

This appendix contains information on using the ProControl utility to start up and connect to a Progress database. The ProControl utility is a tool with graphical user interface and is supplied with the Progress NT Server. The Progress NT Server permits Progress processes to run as registered NT services.

Specifically, this appendix contains the following sections:

- Starting ProService
- Starting Single-user Progress In Batch Or Background Mode
- Starting the Multi-user Server Or Broker
- Starting Windows NT Multi-user Progress In Batch Or Background Mode

NOTE: Using ProControl to perform these tasks ensures that Progress processes run as registered NT services. All of these tasks result in changes to the registry.

B.1 Starting ProService

Follow these steps to access the ProControl Utility and start ProService:

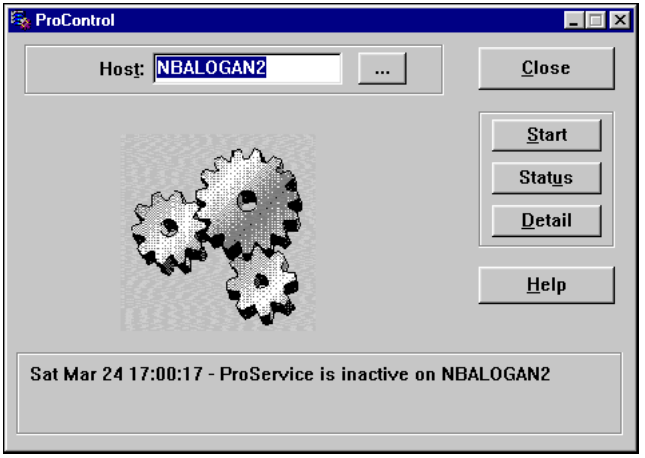
1 ♦ Access the ProControl utility by double-clicking the ProControl icon in the Control Panel, or by double-clicking the ProControl icon in the Progress program group. The ProControl icon looks like this:



Alternatively, you can access ProControl by typing the following command at the command line:

```
proctrl
```

The **ProControl** main window appears:



2 ♦ Select the Windows NT host machine where you want to run ProService by typing the name of the host in the **Host** field, or by selecting the **...** button and choosing a host from the list displayed by the **Select Computer** dialog box.

3 ♦ Choose **Start**.

ProControl displays information in the message area about the status of ProService. In addition, if animation is turned on, the gears on the ProControl Main Window will turn. (You can control animation through the ProControl Options folder.) The message area will indicate that ProService is active:



NOTES

- The Progress NT Server stores server-specific environment information in the NT registry. Each time you use ProControl to perform an Insert, Copy, or Remove operation within one of the ProControl folders, ProControl updates the appropriate subkey of the PSC subkey of the registry key HKEY_LOCAL_MACHINE. Note that when you use ProControl to perform a remove operation, ProControl removes the registry entries, but does not delete the files from the system.

- When you insert a database entry using the Databases folder, ProControl adds the name of the database (and other information) to the Databases subkey of the PSC subkey of the HKEY_LOCAL_MACHINE key. If you then set an environment variable using the Databases Environment folder, ProControl adds that environment variable to the Databases subkey.

- If you do not use ProControl to insert Databases, DataServers, Environment variables, and Other Tasks, the registry is never updated and these processes do not run as NT services. Similarly, if you use some other means to update the registry, Progress cannot guarantee that the product will function properly.

- You manage access to the ProControl utility through the ProControl Access Group field (available through the ProControl Options folder). By default at installation, the ProControl Access Group field is set to the NT group called Everyone.

- You can specify access rights to shared memory for specific databases, DataServers, and tasks by setting the ACCESS environment variable within the Environment folder for that specific object, or globally for the ProService environment through the ProControl Environment folder. See the "Shared Memory" section in Chapter 4, "Administration Utilities," for more information on shared memory.

B.2 Starting Single-user Progress In Batch Or Background Mode

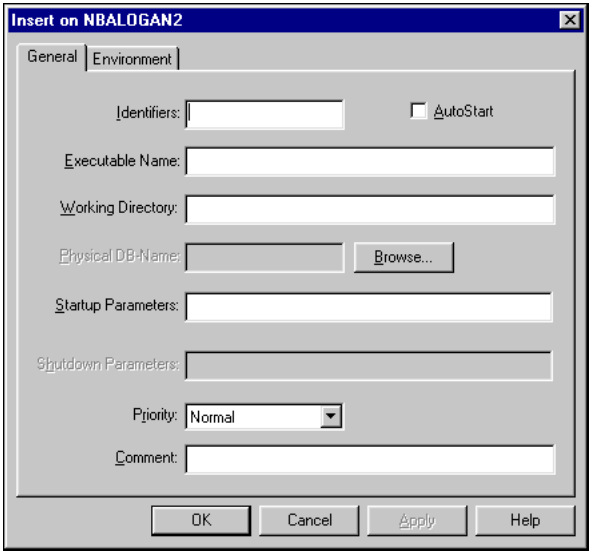
Follow these steps to create a task that can be started as a single-user Progress in batch or background mode by using the ProControl utility:

1 ♦ Access the ProControl utility and start ProService (if it is not already started).

See the "Starting ProService" section earlier in this appendix for information on how to do this.

2 ♦ From the ProControl Main Window, choose **Detail**, then choose the OtherTasks folder.

3 ♦ From the **OtherTasks** folder, choose **Insert**. ProControl displays the **OtherTasks General** folder:



4 ♦ Enter information for the following boxes:

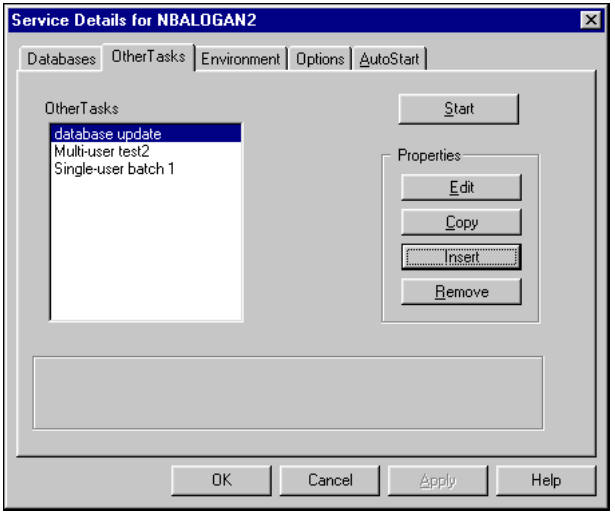
- **Identifiers** — Provides a unique name to identify the task. The name must be unique to the Windows NT host. The Identifier can be up to 30 characters long. For example, you might name a single-user batch session `Single-user Batch 1`. Once you have defined the task, ProControl inserts this name in the selection list. Note that this is the Identifier that you supply when using PCCmd.
- **Executable Name** — To start a single-user batch session, the executable name is:

```
prowin32.exe
```

- **Working Directory** — Specifies the directory that the batch session uses.
- **Browse** — Specifies the executable file and working directory to run.
- **Startup Parameters** — Specifies the Progress startup parameters that you want to use. To start a batch session in single-user mode, you must supply the `-b`, the `-p`, and the `-1` startup parameters and can optionally include the database name:

```
-b -p batch1.p -1
```

5 ♦ Choose **OK**. ProControl displays the **OtherTasks** folder:



6 ♦ Select the task that you want to start from the **OtherTasks** selection list and choose **Start**.

You can specify that this task start automatically by selecting **AutoStart** in the **OtherTasks General** folder. Once you have created this task entry, you can use PCCmd to start it.

B.3 Starting the Multi-user Server Or Broker

To run multi-user Progress as a Windows NT service, you can create a database entry within ProControl. When you create a database entry, ProControl automatically creates an entry in the registry, which enables Progress to run as a Windows NT service.

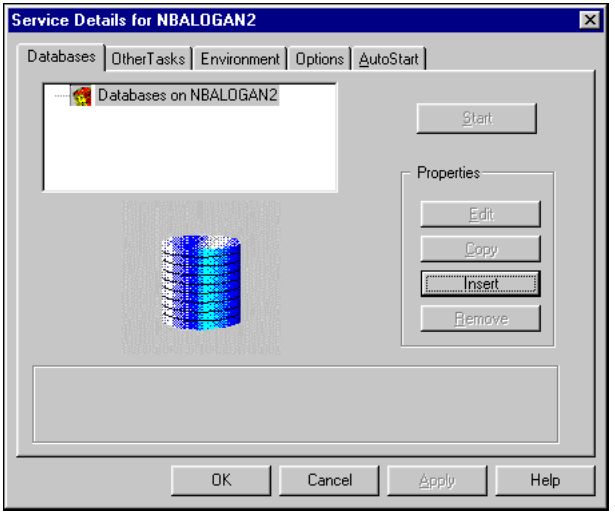
The Progress server process coordinates the database requests from all the users accessing a single database.

Follow these steps to create the ProControl database entry:

1 ♦ Access the ProControl utility and start ProService (if it is not already started).

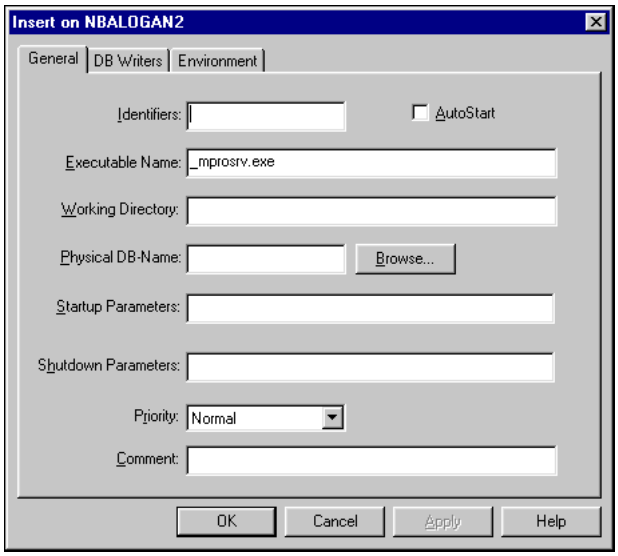
See the "Starting ProService" section for more information about how to use ProControl to start ProService.

2 ♦ From the ProControl Main Window, choose **Detail**, then choose the **Databases folder**. The **Databases** folder appears:



3 ♦ Choose **Insert**.

ProControl displays the **Databases General** folder:



4 ♦ Enter information for the following boxes, then choose **OK**:

- **Identifiers** — Provides a unique name to identify the database. The name must be unique to the host. The identifier can be up to 30 characters long and can include spaces. Note that this corresponds to the Identifier that you supply with the PCCmd command.
- **Executable Name** — Specifies the `_mprosrv` Progress executable. (This is the default.)
- **Working Directory** — Specifies the directory that the service uses, for example, to write before- and after-image files or logs. If you specify a directory other than the directory in which the database resides, you must specify the database's absolute pathname in the **Physical DB-Name** field.
- **Physical DB-Name** — Specifies the database's physical name. If you do not specify a pathname, ProControl appends the directory you specified in the **Working Directory** field.

- **Browse** — Specifies the database file and directory to use.
- **Startup Parameters** — Specifies the Progress server parameters that you want to apply to this database. If you want the database to be visible from a remote host, specify the Service Name (-S), Host Name (-H), and Network Type (-N) startup parameters for the database entry. If you want to activate after-imaging for this database, you must specify the After-image Filename (-a) startup parameter.
- **AutoStart** — Causes Progress to start the database service automatically when you start ProService. Additionally, if you want the database service to start when the host is booted, access **Services** from the Control Panel program group and double-click **Progress Task Manager→ ProService**. In the **Startup Type** section of the **Service** dialog box, choose **Automatic**.

B.4 Starting Windows NT Multi-user Progress In Batch Or Background Mode

Follow these steps to use the ProControl utility to create a task that starts multi-user Progress in batch or background mode on a Windows NT system:

1 ♦ Access the ProControl utility and start ProService (if it is not already started).

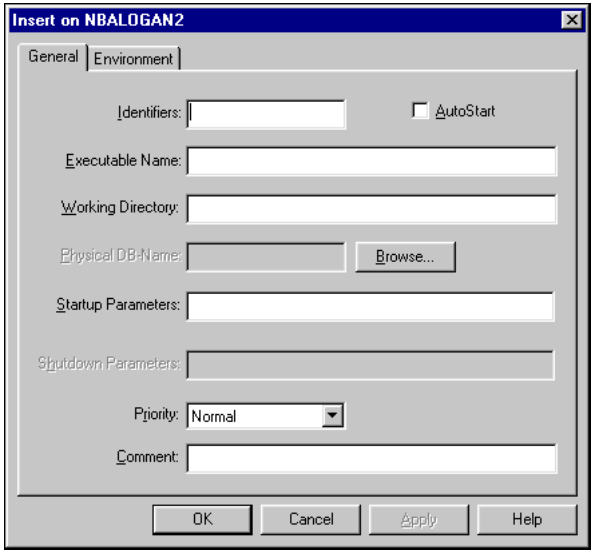
See the "Starting ProService" section earlier in this chapter for information on how to do this.

2 ♦ From the ProControl Main Window, choose **Detail**, then choose the **OtherTasks** folder.

3 ♦ ProControl displays the **OtherTasks** dialog box.

You can specify that this task autostart by selecting **AutoStart** in the **OtherTasks General** folder.

4 ♦ Choose **Insert**. ProControl displays the **OtherTasks General** folder:



5 ♦ Enter information for the following boxes, then choose **OK**:

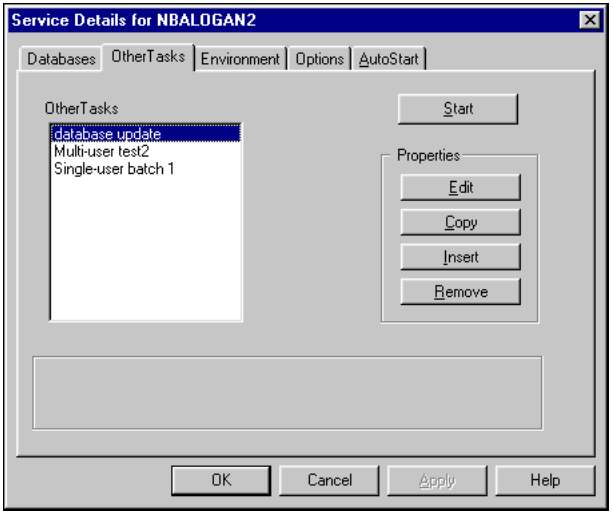
- **Identifiers** — Provides a unique name to identify the task. The name must be unique to the Windows NT host. The Identifier can be up to 30 characters long. For example, you might name a single-user batch session `Multi-user Batch 1`. Once you have defined the task, ProControl inserts this name in the selection list. Note that this corresponds to the Identifier-name used with the PCCmd command.
- **Executable Name** — To start a single-user batch session, the executable name is:

```
prowin32.exe
```

- **Working Directory** — Specifies the directory that the batch session uses.
- **Browse** — Specifies the executable file and working directory to run.
- **Startup Parameters** — Specifies the Progress startup parameters that you want to use. To start a batch session in multi-user mode, you must supply the `-b` and the `-p` startup parameters. For example:

```
-b -p batch1.p
```

6 ♦ From the **OtherTasks** folder, select the task that you want to start from the **OtherTasks** selection list and choose **Start**:



NOTE: You cannot run any task that requires user input or a response to any message. You can only run background or batch tasks that do not require any user action.

Once the task is defined using ProControl, you can use the PCCmd command to start it.

C

The Progress Directory Structure

The Progress installation path contains configuration files and several subdirectories. This appendix describes the subdirectories of the Progress installation path.

Specifically, this appendix contains the following section:

- Installation Path Directory Structure

C.1 Installation Path Directory Structure

The installation path directory contains the Progress executables, several procedure (.p) files, and other related files and subdirectories. The default installation path is C:\Program Files\Progress.

The installation path that you choose is referred to as DLC; for the default Progress installation path, DLC is C:\Program Files\Progress. During installation, the DLC variable (%DLC%) is automatically set to your Progress installation path.

Note that the DLC variable is set in the various command scripts and in the registry; the variable is not set at the system level.

Table C–1 describes a directory tree of the Progress subdirectories.

Table C–1: Progress (%DLC%) Directory Structure *(1 of 2)*

Directory Name	Description
bin	Contains the executable files for Progress, such as PRODB. It also contains batch files and system executables.
certs	Contains the public keys of the Certificate Authorities (CAs) used by Progress clients to perform server-side certificate validation when communicating with secure Web servers using HTTPS.
gui	Contains object (.r), development, and environment (ADE) files for the Progress graphical tools. These tools are compiled to run in graphical mode on Windows NT, Windows 98, Windows 2000, Windows Me, Windows XP, and Citrix MetaFrame; they cannot run in a character environment.
include	Contains the header files required for ESQL-92.
java	Includes the Java files and executables necessary for running Progress products.
jdk	Contains the Java Development Kit files and executables necessary for running Progress products.
jms	Includes files for the 4GL-to-JMS API of the SonicMQ Adapter.
jre	Contains the Java Runtime Environment files and executables necessary for running Progress products.
lib	Contains shared objects necessary for running Progress executables.

Table C–1: Progress (%DLC%) Directory Structure *(2 of 2)*

Directory Name	Description
netsetup	Contains files for the Shared Network Installation Utility.
odbc	Includes files to support ODBC.
ora	Contains files to support the DataServer for ORACLE.
probuild	Includes files that the PROBUILD utility uses when creating custom executables.
proedit	Contains files to support the advanced editing features.
prohelp	Includes the online help and other necessary files for Progress.
prokey32	Contains files for international keyboard support for the 32-bit Windows Character Client.
prolang	Contains the national language support directories.
properties	Contains property files that manage the configuration of Progress services, such as WebSpeed, the NameServer, and the AppServer.
sports	Includes the schema triggers and supplier information for each sample database.
sports2000trgs	Includes the schema triggers for the Sports2000 database.
src	Contains source files for Progress ADE tools, such as the Data Dictionary, Procedure Editor, and Sample Applications.
toolkit	Includes files that help in deploying and encrypting your applications.
tty	Includes mostly object (.r) files for character-mode Progress on Windows NT, Windows 98, and Citrix MetaFrame.
ubqmanager	Includes files used by the AppServer exclusively. Do not modify these files.
webspeed	Includes static files for the WebSpeed Workshop, WebTools, and HTML help.

D

Progress National Language Support

This appendix gives information about Progress messages in different languages. Specifically, this appendix contains the following sections:

- Packaging
- Directory Structure
- Contents Of Each Directory
- Implementing Regional Support
- International Databases
- Progress Messages
- Environment Variables Of the SQL-92 Client
- Regional Parameter Files
- The progress.ini File and the Windows Registry

D.1 Packaging

Table D–1 provides a list of the PROMSGS translations shipped to all Progress users by default.

Table D–1: PROMSGS Translations Shipped With Progress

Supported Languages	
Arabic (ARB)	Japanese (JPN)
Chinese-Simplified (SCH)	Korean (KOR)
Chinese-Traditional (TCH)	Lithuanian (LIT)
Croatian (HRV)	Norwegian (NOR)
Czech (CZE)	Polish (POL)
Danish (DAN)	Portuguese (POR)
Dutch (DUT)	Portuguese-Brazilian (BRZ)
English-American (AME)	Romanian (ROM)
English-International (ENG)	Russian (RUS)
Finnish (FIN)	Serbian (SRB)
French (FRE)	Spanish (SPA)
German (GER)	Spanish-Latin American (SPL)
Greek (GRE)	Spanish-Mexican (MEX)
Hebrew (HBR)	Swedish (SWE)
Hungarian (HUN)	Thai (TAI)
Icelandic (ISL)	Turkish (TUR)
Italian (ITA)	-

D.2 Directory Structure

The %DLC%\prolang\Readme file lists the subdirectories of the \prolang directory by language. Also included is helpful information about code-page tables in convmap.dat. For more information, see the "Installation Path Directory Structure" section in Appendix C, "The Progress Directory Structure."

D.3 Contents Of Each Directory

The prolang directory contains a subdirectory for each national language that Progress supports. In each of these directories, there can be five file types. The variable *lang* stands for language-specific references. Table D–2 provides a description of each file type.

Table D–2: National Language File Descriptions

Filename	Description
empty.db	An empty, language-specific Progress directory containing databases of various block sizes. The database is initialized with an appropriate code page and collation for your language.
promsgs.*lang*	A translated Progress run-time messages file.
lang.pf	A file containing the parameters used to start up Progress with the appropriate settings for your region.
lang.df	A data definition file that can be loaded into an empty Progress database to create a language-specific database. A database created by loading this file is identical to the empty database provided in this directory. You can use this file to create sort ordering variations in the database. In the Asian directories, the file is named _tran.df.
progress.ini	A file containing the parameters used to start up Progress with the appropriate regional settings. For example, the Japanese progress.ini contains Japanese font specifications.

D.4 Implementing Regional Support

The installation utility requires you to install at least one language.

During the installation you must choose a default language. If you want to change the default language after installing Progress, see the *Progress Internationalization Guide* for detailed instructions.

The **International Settings** dialog box of the installation program creates a Progress Startup (`startup.pf`) file to accommodate international conventions such as Date format, Number format, Character set, Collation, and Case.

Once you select the default language, the installation program copies the contents of the `DLC\prolang` directory to `%DLC%`. This affects your `empty.db`, `promsgs`, and `progress.ini` files. For more information, see the "Installation Path Directory Structure" section in Appendix C, "The Progress Directory Structure."

See the *Progress Internationalization Guide* for more information about any of the following files:

- `empty.db` in multiple block sizes
- `startup.pf`
- `promsgs`
- `progress.ini`

D.5 International Databases

As part of the installation media, Progress supplies empty databases that support the language and collation standards of over thirty languages. The databases are located in the `%DLC%\prolang` subdirectories. See Table D–1 for the subdirectory name for your language. For example, the empty database that you might use to build a Russian application is `%DLC%\prolang\rus\empty.db`. For more information, see the "Installation Path Directory Structure" section in Appendix C, "The Progress Directory Structure."

These empty databases provide a database labelled with the appropriate code page and collation table for a language. However, if you are developing applications for a language or region that is not represented in `%DLC%\prolang`, Progress has a utility, PROUTIL, that you can use to set up a unique database. See the *Progress Internationalization Guide* for detailed instructions.

D.6 Progress Messages

The text used in Progress messages is contained in the promsgs file. Progress provides various language editions of the promsgs file in the %DLC%\prolang subdirectories that you select during installation.

Each file has an extension that identifies its language.

To run Progress with a certain version of promsgs, set the PROMSGS environment variable to the appropriate file, as illustrated in the following example:

```
PROMSGS=C:\Program Files\Progress\prolang\ger\promsgs.ger
```

After you set the PROMSGS variable in the progress.ini file, you must run ini2reg to update the registry.

By default, Progress displays messages from %DLC%\promsgs.

D.7 Environment Variables Of the SQL-92 Client

Progress Version 9.1 contains the environment variables SQL_CLIENT_CHARSET and SQL_CLIENT_CHARSET_PROMSGS for SQL-92 clients. You can use these variables to internationalize your applications. These environment variables determine the code page the client uses to display the following:

- Database data from the server
- PROMSGS from the server

NOTES:

- You should set SQL_CLIENT_CHARSET only if you want clients to use a code page that is different from the code page the client operating system uses.
- You should set SQL_CLIENT_CHARSET_PROMSGS only if you want run-time messages to use a code page that is different from either the code page the client operating system uses, or the code page set by SQL_CLIENT_CHARSET.

If you do not set either of these environment variables, then the SQL-92 client code page will correspond to the language of the client operating system.

Code Page Client Uses To Display Data

To display database data from the server, the client uses the code page set by SQL_CLIENT_CHARSET, if you have set this environment variable on the client machine. Otherwise, the client uses the code page of the client's operating system.

If you want to specify an SQL-92 client code page that is different from the client operating system, you can set the SQL_CLIENT_CHARSET environment variable to the name of a Progress code page. When you set this variable to a code page, the SQL-92 server converts text data that is sent *from the server to the client* to the code page set by SQL_CLIENT_CHARSET. The server also uses this code page when it converts text data that is sent *from the client to the server* to the server code page.

Code Page Client Uses To Display PROMSGS From the Server

To display PROMSGS from the server, the client uses the code page set by SQL_CLIENT_CHARSET_PROMSGS, if you have set this environment variable. Otherwise, the client uses the code page set by SQL_CLIENT_CHARSET, if you have set this environment variable. If you have set neither of these environment variables, then the client uses the code page of the client's operating system.

If you want run-time messages at the SQL-92 client to use a different code page from either the client operating system or the code page set by SQL_CLIENT_CHARSET, you can set the SQL_CLIENT_CHARSET_PROMSGS environment variable. When you set this variable to a code page, the SQL-92 server converts run-time messages that are sent *from the server to the client* to the code page set by SQL_CLIENT_CHARSET_PROMSGS.

NOTE: The SQL_CLIENT_CHARSET_PROMSGS environment variable applies to SQLDUMP and SQLLOAD, which are actually SQL-92 applications—specifically, ESQL/C applications.

D.8 Regional Parameter Files

A useful technique for controlling a Progress client session or server is to use a parameter file (`.pf`) with a startup or connection command. Progress provides parameter files that set up Progress sessions appropriately for a wide range of countries. You can use `.pf` files to specify the correct code-page settings for international applications. The setup of the *`install-path`*`\startup.pf` file is based on the installation options that you select.

The international parameter files are located in the `%DLC%\prolang` subdirectories. Parameter files are region- or country-specific rather than language-specific because parameter files set options that can vary from country to country. The `%DLC%\prolang\ger` directory has parameter files for Austria, Germany, and Switzerland to account for the differences among these German-speaking countries. For more information, see the "Installation Path Directory Structure" section in Appendix C, "The Progress Directory Structure."

You should use the parameter file to make sure that the application and database are using the appropriate international settings. Typically, a parameter file for a internationalized application sets the parameters listed in Table D–3.

Table D–3: Startup Parameters For a Deployed Application *(1 of 2)*

Parameter	Description
Internal Code Page (`-cpinternal`)	The code page that Progress uses in memory.
Stream Code Page (`-cpstream`)	The code page for stream I/O.
Case Code Page (`-cpcase`)	A case table in the `convmap.cp` file to use for uppercase/lowercase rules. Case rules are used by the CAPS and LC functions and by the ! formatting character.
Collation Code Page (`-cpcoll`)	A table in the `convmap.cp` file to use for collation rules.
Date Format (`-d`)	The format in which an application displays dates. Specify the format as a three-character string, comprised of the letters d, m, y, in the order that you display the date.
Language (`-lng`)	The initial value for the CURRENT-LANGUAGE function, which determines from which r-code segment Progress reads character-string constants. Specify the language as a character string in quotes.

Table D–3: Startup Parameters For a Deployed Application *(2 of 2)*

Parameter	Description
European Numeric Format (-E)	Progress interprets and displays a comma as a decimal separator and a period as a thousands separator for numeric values.
Fractional Separator (-numdec)	Specifies the numeric value of the character that represents, in formatted text, a number's decimal point. The default decimal point is a period (.).
Thousands Separator (-numsep)	Specifies the numeric value of the character that represents, in formatted text, the thousands separator in numbers. The default thousands separator is a comma (,).

NOTE: You can also use parameter files with Progress utilities, for example, PROSHUT and PROUTIL.

D.9 The progress.ini File and the Windows Registry

The progress.ini file sets up the user interface environment for Progress applications running on Windows and is an important part of deploying a localized application. It controls parts of the environment that vary across locales, and it allows you to specify colors and fonts.

Progress supports the use of the Windows registry on Windows NT, Windows 2000, Windows 98, Windows Me, and Windows XP, and searches the registry first for system configuration information. However, you can still use an initialization file to ensure that deployed applications are configured correctly and consistently at customer sites. The information from the .ini file can be added to the registry upon installation.

Be sure to create and edit the progress.ini file on a system configured like the target system on which you intend to run it. For example, Japanese font names might use Japanese characters. You should edit a progress.ini file for use in Japan on a system supporting Japanese.

If you edit the progress.ini file, run ini2reg to update the registry.

The sections of the progress.ini file that can typically affect a localized application are the [Startup], [WinChar Startup], and [fonts] sections.

[Startup] and [WinChar Startup]

The [Startup] and the [WinChar Startup] sections contain Progress environment-variable settings. The [Startup] section includes the variables for GUI clients, and the [WinChar Startup] section includes the variables for character clients, WebSpeed Agents, and the AppServer.

Table D–4 lists the environment variables that a typical localized application might need.

Table D–4: Environment Variables

Environment Variable	progress.ini File Section	Description
DefaultFont	[Startup]	The default display font.
DefaultFixedFont	[Startup]	The default-fixed display font.
PrinterFont PrinterFont1 PrinterFont2 PrinterFont3	–	The font that the printer uses for the Progress OUTPUT TO PRINTER statement.
PROMSGS	[Startup] [WinChar Startup]	The `promsgs` file that an application should use. For example, for a Progress application to access Swedish translations of Progress error messages, set PROMSGS to `c:\Program Files\Progress\prolang\swe\promsgs.swe`.

[fonts]

The [fonts] section of the `progress.ini` sets the fonts that a Progress application running on that system uses. The default `progress.ini` file that Progress supplies in the United States sets the following fonts:

```
font0=Courier New, size=8
font1=MS Sans Serif, size=8
font2=Courier New, size=8
font3=Courier New, size=8
font4=MS Sans Serif, size=8
font5=MS Sans Serif, size=10
font6=MS Sans Serif, size=8, bold
font7=MS Sans Serif, size=8
```

These font settings might not apply to all the locales where your application will run. Some of the `%DLC%\prolang` directories contain `progress.ini` files with font settings appropriate for that country.

E

Integrating Progress With Windows Explorer

In Windows and Citrix MetaFrame, Microsoft allows applications to integrate certain features with Windows Explorer. Among those features, Progress now supports defined icons, shortcut menus, and property sheets for several of its file types. You can now easily perform an action on a file or view detailed information about a file from Windows Explorer.

This appendix contains the following sections:

- Progress File Types
- Icons
- Shortcut Menus
- Properties

E.1 Progress File Types

Progress supports defined icons and shortcut menus for the following file types:

- Procedure source code file (`.p`)
- Window procedure source code file (`.w`)
- Include file (`.i`)
- Parameter file (`.pf`)
- Configuration file (`.cfg`)
- Database file (`.db`)

Progress also supports specific property information for these file types:

- Compiled procedure code file (`.r`)
- Database file (`.db`)

This information is stored in the registry, separately from your Progress settings.

If another application has already registered a file extension that Progress uses, the Installation Utility asks if you want to overwrite the information for that file extension. If you choose no, Progress does not display the icon, shortcut menu options, or properties information for that file type. If you choose yes, Progress replaces the icon, shortcut menu options, or properties associated with the file extension with Progress-specific information.

The shell integration DLL uses the DLC and, optionally, the PROMSGS environment variables to locate the PROMSGS file. The DLL searches these registry locations for the following variables:

- `HKEY_CURRENT_USER\SOFTWARE\PSC\PROGRESS\9.1D\Startup`
- `HKEY_LOCAL_MACHINE\SOFTWARE\PSC\PROGRESS\9.1D\Startup`

The Installation utility writes the proper values to the above registry locations. However, if after the installation you move Progress to another location (or move or rename the PROMSGS file), you must edit the variables in the registry so that the shell integration DLL can find PROMSGS.

E.2 Icons

Progress now associates each of the Progress-supported file types, except for compiled procedure code files (`.r`), with a unique icon that is displayed in Windows Explorer. You can execute the default action on a file by double-clicking on its icon. To perform other actions, you can right-click on the file and choose one of the options from the shortcut menu. To change the default setting, see the "Shortcut Menus" section.

E.3 Shortcut Menus

A shortcut menu allows you to perform an action on a file by eliminating several steps to accomplish the task. Progress enhances this feature by adding context-specific options for each file type. For example, to edit the Sports database from Windows Explorer, just right-click the sports.db icon and choose **Edit in Data Dictionary** (single-user) from the shortcut menu. If you do not use the shortcut menu, this same action requires several more steps.

To view the shortcut menu for a specific file, right-click the file. A shortcut menu appears with context-specific options.

To add to or change your default shortcut menu options, choose **View→Options→File Types** from Windows Explorer. In the **Registered file types** list, choose the Progress file type you want to modify and click **Edit**.

The command line for each shortcut menu option includes a full path to the Progress executable. If you move this executable to another location, you must modify the path.

E.4 Properties

By default, Microsoft provides general information about a file in its properties sheet. Progress adds an extra page containing specific information for compiled procedure code (`.r`) and database (`.db`) file types. To view a file's properties, right-click on the file and choose Properties from the shortcut menu.

F

Progress Version 9 Typical and Complete Installation Components

When you install Progress Version 9, you can choose from three installation options: Typical, Complete, or Custom. This appendix provides you with a list of the components and subcomponents that you install for each product when you choose either a Typical or Complete installation.

This appendix includes the following sections:

- Progress Version 9 Installation Options
- Progress Product Components and Subcomponents

F.1 Progress Version 9 Installation Options

You can choose from three options when installing Progress Version 9: Typical, Complete, or Custom. The three different installation options allow you to choose the option that is best for you, depending on how many products you are installing, which product components are mandatory and which are optional, and whether all the products reside on the same system.

When you choose the Typical installation option and specify the products you want to install, all mandatory components and subcomponents are installed automatically. Optional components and subcomponents are not included in the installation.

When you choose the Complete installation option and specify the products you want to install, all mandatory and optional components and subcomponents are installed automatically. For this reason, a Complete installation usually requires more disk space than a Typical installation requires.

When you choose the Custom installation option and specify the products you want to install, you can choose the mandatory and optional components and subcomponents to include in the installation. You can also distribute Progress components on different machines and work around disk space limitations. This method is recommended for advanced users.

For a description of the steps to follow when installing Progress Version 9, see Chapter 2, "Installing Progress."

The mandatory and optional components and subcomponents for each Progress Version 9 product are listed, by product, in the following tables.

F.2 Progress Product Components and Subcomponents

The following tables list the components and subcomponents that are installed for each product depending on whether you choose the Complete or the Typical installation option.

F.2.1 4GL Development

Table F–1 lists the 4GL Development components and subcomponents. When you choose the Complete installation option and install 4GL Development, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–1: 4GL Development Components and Subcomponents *(1 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client — 4GL	M
		Graphical Client	M
		XML	M
ActiveX Control Support	M	ActiveX Control Development Support	M
		ActiveX Control Runtime Support	M
ADE Source Code	M	**4GL Profiler**	**O**
		ADE Common Source	M
		ADM Source Code	**O**
		DB Administration Source	**O**
		Demo Applications	**O**
		Editor Source	**O**
		ProTools Source Code	M
AS/400 DataServer	**O**	–	–

Table F–1: 4GL Development Components and Subcomponents *(2 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Character Base Tools	**O**	**ADM Runtime — CHAR**	**O**
		Compile Tool — CHAR	M
		Procedure Editor — CHAR	M
Character Database Admin Tools	**O**	–	–
Character Image — Dev	**O**	–	–
Character Runtime Client — Dev	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M
Database Server Component	M	4GL Database	M
		4GL Server	M
		Database Server	M

Table F–1: 4GL Development Components and Subcomponents *(3 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Graphical Base Tools — Dev	M	**ADM Runtime — GUI**	O
		Advanced Editing	M
		Base ADE	M
		Compile Tool — GUI	M
		Desktop	M
		Help Compiler — GUI	M
		Procedure Editor — GUI	M
Intersolv ODBC Driver Support	**O**	–	–
MS SQL Server DataServer Client	**O**	–	–
NetSetup	**O**	–	–
ODBC DataServer Client	**O**	–	–
Open Interface Driver	**O**	–	–
Open DataSever Client	**O**		
Oracle DataServer Client	**O**	–	–
ProBuild Utility	**O**	–	–
ProControl Utility	**O**	–	–

Table F–1: 4GL Development Components and Subcomponents *(4 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress Explorer Tools	M	Administration Server	M
		Explorer Tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	**All Languages**	O
Progress SQL ESQL/C Clients	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JBDC Client	**M**
Progress SQL ODBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL OBDC Client	**M**
Report Engine	M	–	–

Table F–1: 4GL Development Components and Subcomponents *(5 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
SQL Database Server — Development	**O**	**Database Server**	**M**
		Database Utilities	**M**
		JDK	**M**
		Progress Class Files	**M**
		Progress Databases	**M**
		SQL Server	**M**

NOTE: When you choose a Complete installation and install 4GL Development, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.2 AppServer Internet Adapter

Table F–2 lists the AppServer Internet Adapter components. When you choose the Complete installation option and install the AppServer Internet Adapter, all components listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–2: AppServer Internet Adapter Components and Subcomponents

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
AppServer Internet Adapter	M	–	–
Common Files (minimum)	M	–	–
Progress Messages (PROMSGS)	M	**All Languages**	O

F.2.3 AppServer Internet Adapter/S

Table F–3 lists the AppServer Internet Adapter/S components. When you choose the Complete installation option and install the AppServer Internet Adapter/S, all components listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–3: AppServer Internet Adapter/S Components and Subcomponents

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
AppServer Internet Adapter/S	M	–	–
Common Files (minimum)	M	–	–
Progress Messages (PROMSGS)	M	All Languages	O

F.2.4 Client Networking

Table F–4 lists the Client Networking components and subcomponents. When you choose the Complete installation option and install Client Networking, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–4: Client Networking Components and Subcomponents *(1 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ActiveX Control Support	M	ActiveX Control Runtime Support	M
AS/400 DataServer Client	**O**	–	–

Table F–4: Client Networking Components and Subcomponents *(2 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Character Base Tools	**O**	**ADM Runtime — CHAR**	**O**
		Compile Tool — CHAR	**M**
		Procedure Editor — CHAR	**M**
Character Database Admin Tools	**O**	–	–
Character Image — Dev	**O**	–	–
Character Runtime Client — Dev	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Database Administration Tools	**O**	**4GL Database**	**M**
		Base ADE	**M**
		Database Utilities	**M**
		Graphical Administration Tools	**M**
Graphical Base Tools—Client	M	**ADM Runtime — GUI**	O
		Base ADE	M
		Compile Tool — GUI	O
		Desktop	O
		Help Compiler — GUI	O
		Procedure Editor — GUI	M

Table F–4: Client Networking Components and Subcomponents *(3 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Intersolv ODBC Driver Support	**O**	–	–
MS SQL Server DataServer Client	**O**	–	–
NetSetup	**O**	–	–
ODBC DataServer Client	**O**	–	–
Open Interface Driver	**O**	–	–
Oracle DataServer Client	**O**	–	–
ProBuild Utility	**O**	–	–
Progress Messages (PROMSGS)	M	**All Languages**	O
Progress SQL ESQL/C Clients	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
Progress SQL ODBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**

Table F–4: Client Networking Components and Subcomponents *(4 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Report Engine	M	–	–
Runtime Client	M	Base Client — RT	M
		Graphical Client	M
		XML	M

NOTE: When you choose a Complete installation and install Client Networking, all components and subcomponents listed in the table are installed. when you choose a typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.5 Client-side Security

Table F–5 lists the Client-side Security components. When you choose the Complete installation option and install the Debugger, all components listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–5: Client-side Security Components and Subcomponents

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Secure AppServer Client	M	–	–

F.2.6 Debugger

Table F–6 lists the Debugger components. When you choose the Complete installation option and install the Debugger, all components listed are installed. Choosing the Typical installation option includes the Common Files and Debugger components and excludes the NetSetup component.

Table F–6: Debugger Components

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Common Files	M	–	–
Debugger	M	–	–
NetSetup	O	–	–

F.2.7 Enterprise Database

Table F–7 lists the Enterprise Database components and subcomponents. When you choose the Complete installation option and install the Enterprise Database, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

NOTE: When you install a Progress database product, the **Select Server Engines** dialog box will appear, asking you to choose to install either the 4GL or SQL-92 server engine, or both (the default selection is to install both the 4GL and SQL-92 server engines). Your choice of server engines will determine whether you have 4GL or SQL-92 access to the Progress data files. See Chapter 2, "Installing Progress," for more information on selecting server engines.

Table F–7: Enterprise Database Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL	**O**	**4GL Server**	**M**
		ActiveX Control Runtime Support	**M**
		ADM Runtime — CHAR	**O**
		ADM Runtime — GUI	**O**
		AS/400 Client	**O**
		Base ADE	**M**
		Base Client — DA	**M**
		Character Administration Tools	**O**
		Character Client — 4GL	**O**
		Character Image	**O**
		Compile Tool — CHAR	**O**

Table F–7: Enterprise Database Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL (cont.)	–	**Compile Tool — GUI**	**O**
		Desktop	**M**
		Graphical Administration Tools	**M**
		Graphical Client	**M**
		Help Compiler — GUI	**O**
		MS SQL Server DataServer Client	**O**
		ODBC DataServer Client	**O**
		Open Interface Driver	**O**
		Oracle Client	**O**
		Procedure Editor — CHAR	**O**
		Procedure Editor — GUI	**M**
		Report Engine	**M**
		Server Networking	**M**
		XML	**M**
Bastille Tools	M	Database Server	**M**
		Database Utilities	**M**
		Progress Databases	**M**
Common Files	M	–	–

Table F–7: Enterprise Database Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Legacy Version 8.2 DB Utilities	**O**	–	–
Legacy Version 8.3 DB Utilities	**O**	–	–
NetSetup	**O**		
ProBuild Utility	**O**	–	–
ProControl Utility	**O**	–	–
Progress Explorer Tools — DB	M	Administration Server	M
		Explorer Tools	M
		JRE	M
		Progress Class Files	M
Progress Messages (PROMSGS)	M	**All Languages**	O
SQL	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
		SQL ODBC Client	**M**
		SQL Server	**M**

NOTE: When you choose a Complete installation and install the Enterprise Database, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.8 ESQL-89/C

Table F–8 lists the ESQL-89/C components. Choosing the Complete or the Typical installation option results in the installation of all components listed.

Table F–8: ESQL-89/C Components

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Common Files	M	–	–
ESQL-89/C	M	–	–

F.2.9 NameServer

Table F–9 lists the NameServer components and subcomponents. Choosing the Complete or the Typical Installation option results in the installation of all components and subcomponents listed.

Table F–9: NameServer Components and Subcomponents

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Common Files	M	–	–
NameServer	M	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer Tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	All Languages	O

F.2.10 NameServer Load Balancing Option

Table F–10 lists the AppServer Load Balancing components. When you choose the Complete installation option and install the AppServer Load Balancing option, all components listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–10: AppServer Load Balancing Option Components and Subcomponents

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Common Files	M	–	–

F.2.11 ODBC Enterprise DataServer

Table F–11 lists the ODBC Enterprise DataServer components and subcomponents. When you choose the Complete installation option and install the ODBC Enterprise DataServer, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–11: ODBC Enterprise DataServer Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ActiveX Control Support	M	ActiveX Control Runtime Support	M
AS/400 DataServer Client	**O**	–	–
Character Base Tools	**O**	**ADM Runtime — CHAR**	**O**
		Compile Tool — CHAR	**M**
		Procedure Editor — CHAR	**M**
Character Database Admin Tools	**O**	–	–
Character Image — Dev	**O**	–	–
Character Runtime Client — Dev	**O**	–	–
Common Files	M	–	–
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M

Table F–11: ODBC Enterprise DataServer Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Graphical Base Tools	M	**ADM Runtime — GUI**	O
		Base ADE	M
		Compile Tools — GUI	O
		Desktop	M
		Help Compiler — GUI	O
		Procedure Editor — GUI	M
MS SQL Server DataServer	**O**	–	–
NetSetup	**O**		
ODBC DataServer	M	–	–
ODBC DataServer Drivers	M	–	–
ODBC DataServer — Client	M	–	–
Oracle DataServer Client	**O**	–	–
ProBuild Utility	M	–	–
ProControl Utility	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	All Languages	O

Table F–11: ODBC Enterprise DataServer Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Report Engine	M	–	–
Runtime Client	M	Base Client — RT	M
		Graphical Client	M
		XML	M
Schema Holder Server	M	–	–
Server Networking	M	–	–

NOTE: When you choose a Complete installation and install the ODBC Enterprise DataServer, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.12 ODBC Personal DataServer

Table F–12 lists the ODBC Personal DataServer components and subcomponents. When you choose the Complete installation option and install the ODBC Personal DataServer, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–12: ODBC Personal DataServer Components and Subcomponents
(1 of 2)

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ActiveX Control Support	M	ActiveX Control Runtime Support	M
AS/400 DataServer Client	**O**	–	–
Character Base Tools	**O**	**ADM Runtime — CHAR**	**O**
		Compile Tool — CHAR	**M**
		Procedure Editor — CHAR	**M**
Character Database Admin Tools	**O**	–	–
Character Image — Dev	**O**	–	–
Character Runtime Client — Dev	**O**	–	–
Common Files	M	–	–
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M

Table F–12: ODBC Personal DataServer Components and Subcomponents
(2 of 2)

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Graphical Base Tools	M	**ADM Runtime — GUI**	O
		Base ADE	M
		Compile Tool — GUI	O
		Desktop	M
		Help Compiler — GUI	O
		Procedure Editor — GUI	M
MS SQL Server DataServer Client	**O**	–	–
NetSetup	**O**		
ODBC DataServer — Client	M	–	–
Oracle DataServer Client	**O**	–	–
ProBuild Utility	M	–	–
ProControl Utility	**O**	–	–
Progress Messages (PROMSGS)	M	All Languages	O
Report Engine	M	–	–
Runtime Client	M	Base Client — RT	M
		Graphical Client — RT	M
		XML	M
Schema Holder Server	M	–	–

NOTE: When you choose a Complete installation and install the ODBC Personal DataServer, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.13 Open Client Toolkit

Table F–13 lists the Open Client Toolkit components and subcomponents. Choosing the Complete or the Typical installation option results in the installation of all components and subcomponents listed.

Table F–13: Open Client Toolkit Components

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Open Client Common	M	–	–
Open Client Runtime	M	JRE	M
		Open Client Toolkit	M
		Progress Class Files	M
Proxy Generator	M	JDK	M
		JRE	M
		Proxy Generator	M
Secure AppServer Client	M	–	–

F.2.14 Oracle DataServer

Table F–14 lists the Oracle DataServer components and subcomponents. When you choose the Complete installation option and install the Oracle DataServer, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–14: Oracle DataServer Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ActiveX Control Support	M	ActiveX Control Runtime Support	M
AS/400 DataServer Client	**O**	–	–
Character Base Tools	**O**	**ADM Runtime — CHAR**	**O**
		Compile Tool — CHAR	**M**
		Procedure Editor — CHAR	**M**
Character Database Admin Tools	**O**	–	–
Character Image — Dev	**O**	–	–
Character Runtime Client — Dev	**O**	–	–
Common Files	M	–	–
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M

Table F–14: Oracle DataServer Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Graphical Base Tools	M	**ADM Runtime — GUI**	O
		Base ADE	M
		Compile Tools — GUI	O
		Desktop	M
		Help Compiler — GUI	O
		Procedure Editor — GUI	M
MS SQL Server DataServer Client	**O**	–	–
NetSetup	**O**	–	–
ODBC DataServer Client	**O**	–	–
Oracle DataServer	M	–	–
Oracle DataServer Client	M	–	–
ProBuild Utility	M	–	–
ProControl Utility	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	All Languages	O
Report Engine	M	–	–

Table F–14: Oracle DataServer Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Runtime Client	M	Base Client — RT	M
		Graphical Client	M
		XML	M
Schema Holder Server	M	–	–
Server Networking	M	–	–

NOTE: When you choose a Complete installation and install the Oracle DataServer, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.15 Personal Database

Table F–15 lists the Personal Database components and subcomponents. When you choose the Complete installation option and install the Personal Database, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

NOTE: When you install a Progress database product, the **Select Server Engines** dialog box will appear, asking you to choose to install either the 4GL or SQL-92 server engine, or both (the default selection is to install both the 4GL and SQL-92 server engines). Your choice of server engines will determine whether you have 4GL or SQL-92 access to the Progress data files. See Chapter 2, "Installing Progress," for more information on selecting server engines.

Table F–15: Personal Database Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL	**O**	4GL Server	**M**
		ActiveX Control Runtime Support	**M**
		ADM Runtime — CHAR	**O**
		ADM Runtime — GUI	**O**
		AS/400 Client	**O**
		Base ADE	**M**
		Base Client — DA	**M**
		Character Administration Tools	**O**
		Character Client — 4GL	**O**
		Character Image	**O**
		Compile Tool — CHAR	**O**

Table F–15: Personal Database Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL (cont.)	–	Compile Tool — GUI	**O**
		Desktop	**M**
		Graphical Administration Tools	**M**
		Graphical Client	**M**
		Help Compiler — GUI	**O**
		MS SQL Server DataServer Client	**O**
		ODBC DataServer Client	**O**
		Open Interface Driver	**O**
		Oracle Client	**O**
		Procedure Editor — CHAR	**O**
		Procedure Editor — GUI	**M**
		Report Engine	**M**
		Server Networking	**M**
		XML	**M**
Bastille Tools	M	Database Server	**M**
		Database Utilities	**M**
		Progress Databases	**M**
Common Files	M	–	–

Table F–15: Personal Database Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Legacy Version 8.2 DB Utilities	**O**	–	–
Legacy Version 8.3 DB Utilities	**O**	–	–
NetSetup	**O**	–	–
ProBuild Utility	**O**	–	–
ProControl Utility	**O**	–	–
Progress Explorer Tools — DB	M	Administration Server	M
		Explorer Tools	M
		JRE	M
		Progress Class Files	M
Progress Messages (PROMSGS)	M	All Languages	**O**
SQL	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
		SQL ODBC Client	**M**
		SQL Server	**M**

NOTE: When you choose a Complete installation and install the Personal Database, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.16 Progress AppServer

Table F–16 lists the Progress AppServer components and subcomponents. When you choose the Complete installation option and install the AppServer, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–16: Progress AppServer Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL to JMS Adapter	M	Progress Class Files	M
		Sonic MQ Adapter	M
ActiveX Control Support	M	ActiveX Control Development Support	M
		ActiveX Control Runtime Support	M
AppServer Internet Adapter	M	–	–
AppServer Runtime Client	M	–	–
AS/400 DataServer Client	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Intersolv ODBC Driver Support	O	–	–
MS SQL Server DataServer Client	**O**	–	–
NameServer	M	–	–
ODBC DataServer Client	**O**	–	–
Open AppServer	M	Application Server	M

Table F–16: Progress AppServer Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
		Base ADE	M
		Common Broker	M
		Progress Class Files	M
		XML	M
Oracle DataServer Client	**O**	–	–
ProBuild Utility	**O**	–	–
ProControl Utility	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer Tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	**All Languages**	O
Progress SQL ESQL/C Clients	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**

Table F–16: Progress AppServer Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress SQL ODBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**

NOTE: When you choose a Complete installation and install the Progress AppServer, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.17 ProVision

Table F–17 lists the ProVision components and subcomponents. When you choose the Complete installation option and install ProVision, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–17: ProVision Components and Subcomponents *(1 of 6)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client — 4GL	M
		Graphical Client	M
		XML	M
4GL to JMS Adapter	**O**	Progress Class Files	**M**
		Sonic MQ Adapter	**M**

Table F–17: ProVision Components and Subcomponents *(2 of 6)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ActiveX Control Support	M	ActiveX Control Development Support	M
		ActiveX Control Runtime Support	M
ADE Source Code — Development	M	**4GL Profiler**	O
		ADE Common Source	M
		ADM Source Code	O
		DB Administration Source	O
		Demo Applications	O
		Editor Source	O
		ProTools Source Code	M
		Sample Application	O
AppBuilder	M	AppBuilder Core	M
		Base ADE	M
		Client/Server AppBuilder	M
AppServer Internet Adapter	M	–	–
AS/400 DataServer Client	**O**	–	–
Character Base Tools	**O**	**ADM Runtime — CHAR**	**O**
		Compile Tool — CHAR	**M**
		Procedure Editor — CHAR	**M**

Table F–17: ProVision Components and Subcomponents *(3 of 6)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Character Database Admin Tools	**O**	–	–
Character Image — Dev	**O**	–	–
Character Runtime Client — Dev	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M
Database Server Component	M	4GL Database	M
		4GL Server	M
		Database Server	M
Debugger	M	–	–
Development Server	**O**	**Application Server**	**M**
		Common Broker	**M**
		Progress Class Files	**M**
		XML	**M**

Table F–17: ProVision Components and Subcomponents *(4 of 6)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Graphical Base Tools-DEV	M	ADM Run-time — GUI	**O**
		Advanced Editing	M
		Base ADE	M
		Compile Tool — GUI	M
		Desktop	M
		Help Compiler — GUI	M
		Procedure Editor — GUI	M
InterSolv ODBC Driver Support	**O**	–	–
MS SQL Server DataServer Client	**O**	–	–
NameServer	**O**	–	–
NetSetup	**O**	–	–
ODBC DataServer Client	**O**	–	–
Open Interface Driver	**O**	–	–
Oracle DataServer Client	**O**	–	–
ProBuild Utility	**O**	–	–
ProControl Utility	**O**	–	–

Table F–17: ProVision Components and Subcomponents *(5 of 6)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress Explorer Tools	M	Administration Server	M
		Explorer Tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	All Languages	O
Progress SQL ESQL/C Clients	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
Progress SQL ODBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**
Report Builder	**O**	**Report Builder**	**M**
		Report Builder Engine	**M**

Table F–17: ProVision Components and Subcomponents *(6 of 6)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Results	**O**	**Base ADE**	**M**
		Results (Graphical)	**M**
SQL Database Server — Development	**O**	**Database Server**	**M**
		Database Utilities	**M**
		JDK	**M**
		Progress Class Files	**M**
		Progress Databases	**M**
		SQL Server	**M**

NOTE: When you choose a Complete installation and install ProVision, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.18 ProVision Development Server

Table F–18 lists the ProVision Development Server components and subcomponents. When you choose the Complete installation option and install the ProVision Development Server, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–18: ProVision Development Server Components and Subcomponents *(1 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client — 4GL	M
		Graphical Client	M
		XML	M

Table F–18: ProVision Development Server Components and Subcomponents *(2 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL to JMS Adapter	M	Progress Class Files	M
		Sonic MQ Adapter	M
ActiveX Control Support	M	ActiveX Control Runtime Support	M
AppServer Internet Adapter	M	–	–
AS/400 DataServer Client	**O**	–	–
Character Base Tools	**O**	ADM Runtime-CHAR	**O**
		Compiler Tool-CHAR	M
		Procedure Editor-CHAR	M
Character Database Admin Tool	**O**	–	–
Character Image - Dev	**O**	–	–
Character Runtime Client - Dev	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M

Table F–18: ProVision Development Server Components and Subcomponents *(3 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Database Server Component	M	4GL Database	M
		4GL Server	M
		Database Server	M
Debugger	**O**	–	–
InterSolv ODBC Driver Support	**O**	–	–
MS SQL Server DataServer Client	**O**	–	–
NameServer	M	–	–
ODBC DataServer Client	**O**	–	–
Open AppServer	M	Application Server	M
		Base ADE	M
		Common Broker	M
		Progress Class Files	M
		XML	M
Oracle DataServer Client	**O**	–	–
ProBuild Utility	**O**	–	–
ProControl Utility	**O**	–	–

Table F–18: ProVision Development Server Components and Subcomponents *(4 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	**All Languages**	**O**
Progress SQL ESQL/C Clients	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
Progress SQL ODBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**

Table F–18: ProVision Development Server Components and Subcomponents *(5 of 5)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
SQL Database Server — Development	**O**	**Database Server**	**M**
		Database Utilities	**M**
		JDK	**M**
		Progress Class Files	**M**
		Progress Databases	**M**
		SQL Server	**M**

NOTE: When you choose a Complete installation and install the ProVision Development Server, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.19 ProVision Plus

Table F–19 lists the ProVision Plus components and subcomponents. When you choose the Complete installation option and install ProVision Plus, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–19: ProVision Plus Components and Subcomponents *(1 of 7)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client — 4GL	M
		Graphical Client	M
		XML	M
4GL to JMS Adapter	**O**	Progress Class Files	**M**
		Sonic MQ Adapter	**M**

Table F–19: ProVision Plus Components and Subcomponents *(2 of 7)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ActiveX Control Support	M	ActiveX Control Development Support	M
		ActiveX Control Runtime Support	M
ADE Source Code — Development	M	**4GL Profiler**	O
		ADE Common Source	M
		ADM Source Code	O
		DB Administration Source	O
		Demo Applications	O
		Editor Source	O
		ProTools Source Code	M
		Sample Application	O
AppBuilder — Plus	M	AppBuilder Core	M
		Base ADE	M
		Client/Server AppBuilder	M
		WebSpeed AppBuilder	M
AppServer Internet Adapter	M	–	–
AS/400 DataServer Client	**O**	–	–

Table F–19: ProVision Plus Components and Subcomponents *(3 of 7)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Character Base Tools	**O**	**ADM Runtime — CHAR**	**O**
		Compile Tool — CHAR	**M**
		Procedure Editor — CHAR	**M**
Character Database Admin Tools	**O**	–	–
Character Image — Dev	**O**	–	–
Character Runtime Client — Dev	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M
Database Server Component	M	4GL Database	M
		4GL Server	M
		Database Server	M
Debugger	M	–	–

Table F–19: ProVision Plus Components and Subcomponents *(4 of 7)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Development Server	**O**	**Application Server**	**M**
		Common Broker	**M**
		Progress Class Files	**M**
		XML	**M**
Graphical Base Tools — Dev	M	**ADM Runtime — GUI**	O
		Advanced Editing	M
		Base ADE	M
		Compile Tool — GUI	M
		Desktop	M
		Help Compiler — GUI	M
		Procedure Editor — GUI	M
Intersolv ODBC Driver Support	**O**	–	–
Messenger Component	**O**	–	–
MS SQL Server DataServer Client	**O**	–	–
NameServer	**O**	–	–
NetSetup	**O**	–	–
ODBC DataServer Client	**O**	–	–
Open Client Runtime	M	JRE	M
		Open Client Toolkit	M
		Progress Class Files	M

Table F–19: ProVision Plus Components and Subcomponents *(5 of 7)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Open Interface Driver	**O**	–	–
Oracle DataServer Client	**O**	–	–
ProBuild Utility	**O**	–	–
ProControl Utility	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	**All Languages**	O
Progress SQL ESQL/C Clients	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**

Table F–19: ProVision Plus Components and Subcomponents *(6 of 7)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress SQL ODBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**
Proxy Generator	M	JDK	M
		JRE	M
		Proxy Generator	M
Report Builder	**O**	**Report Builder**	**M**
		Report Builder Engine	**M**
Results	**O**	**Base ADE**	**M**
		Results (Graphical)	**M**
SQL Database Server — Development	**O**	**Database Server**	**M**
		Database Utilities	**M**
		JDK	**M**
		Progress Class Files	**M**
		Progress Database	**M**
		SQL Server	**M**

Table F–19: ProVision Plus Components and Subcomponents *(7 of 7)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Transaction Server	**O**	**Progress Database**	**M**
		Web Static	**M**
		WebSpeed Runtime	**M**
		WebSpeed Server Development	**M**
		XML	**M**
WebClient Assembler	M		
WebSpeed Workshop	M	**Advanced Editing**	M
		Progress Databases	M
		Workshop	M

NOTE: When you choose a Complete installation and install ProVision Plus, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.20 ProVision Plus Development Server

Table F–20 lists the ProVision Plus Development Server components and subcomponents. When you choose the Complete installation option and install the ProVision Plus Development Server, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–20: ProVision Plus Development Server Components and Subcomponents *(1 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client — 4GL	M
		Graphical Client	M
		XML	M
4GL to JMS Adapter	M	Progress Class Files	M
		Sonic MQ Adapter	M
ActiveX Control Support	M	ActiveX Control Runtime Support	M
AppServer Internet Adapter	M	–	–
AS/400 DataServer Client	**O**	–	–
Character Base Tools	**O**	ADM Runtime–CHAR	O
		Compile Tool–CHAR	M
		Procedure Editor-CHAR	M
Character Database Admin Tools	**O**	–	–
Character Image-Dev	**O**	–	–
Character Runtime Client–Dev	O	–	–
Client Networking	M	–	–

Table F–20: ProVision Plus Development Server Components and Subcomponents *(2 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Common Files	M	–	–
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M
Database Server Component	M	4GL Database	M
		4GL Server	M
		Database Server	M
Debugger	**O**	–	–
Intersolv ODBC Driver Support	**O**	–	–
Messenger Component	M	–	–
MS SQL Server DataServer Client	**O**	–	–
NameServer	M	–	–
ODBC DataServer Client	**O**	–	–
Open AppServer	M	Application Server	M
		Base ADE	M
		Common Broker	M
		Progress Class Files	M
		XML	M

Table F–20: ProVision Plus Development Server Components and Subcomponents *(3 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Oracle DataServer Client	**O**	–	–
ProBuild Utility	**O**	–	–
ProControl Utility	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	**All Languages**	**O**
Progress SQL ESQL/C Clients	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**

Table F–20: ProVision Plus Development Server Components and Subcomponents *(4 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress SQL ODBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**
SQL Database Server-Development	**O**	**Database Server**	**M**
		Database Utilities	**M**
		JDK	**M**
		Progress Class Files	**M**
		Progress Databases	**M**
		SQL Server	**M**
Transaction Server	M	Progress Databases	M
		Web Static	M
		WebSpeed Runtime	M
		WebSpeed Server Development	M
		XML	M

NOTE: When you choose a Complete installation and install the ProVision Plus Development Server, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.21 Query/Results

Table F–21 lists the Query/Results components and subcomponents. When you choose the Complete installation option and install Query/Results, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–21: Query/Results Components and Subcomponents *(1 of 2)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ActiveX Control Support	M	ActiveX Control Runtime Support	M
AS/400 DataServer Client	**O**	–	–
Character Base Tools	**O**	**ADM Runtime — CHAR**	**O**
		Compile Tool — CHAR	**M**
		Procedure Editor — CHAR	**M**
Character Database Admin Tools	**O**	–	–
Character Image — Dev	**O**	–	–
Character Runtime Client — Dev	**O**	–	–
Common Files	M	–	–
Database Administration Tools	**O**	**4GL Database**	**M**
		Base ADE	**M**
		Database Utilities	**M**
		Graphical Administration Tools	**M**

Table F–21: Query/Results Components and Subcomponents *(2 of 2)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Graphical Base Tools	M	**ADM Runtime — GUI**	**O**
		Base ADE	M
		Compile Tool — GUI	**O**
		Desktop	M
		Help Compiler — GUI	**O**
		Procedure Editor — GUI	M
MS SQL Server DataServer Client	**O**	–	–
NetSetup	**O**	–	–
ODBC DataServer Client	**O**	–	–
Oracle DataServer Client	**O**	–	–
ProBuild Utility	**O**	–	–
Progress Messages (PROMSGS)	M	All Languages	O
Query Client	M	Base Client — Query	M
		Graphical Client	M
		XML	M
Report Engine	M	–	–
Results	M	Base ADE	M
		Results (Graphical)	M

NOTE: When you choose a Complete installation and install Query/Results, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.22 Report Builder

Table F–22 lists the Report Builder components and subcomponents. Choosing the Complete option results in the installation of all components and subcomponents listed; choosing the Typical option excludes the NetSetup component.

Table F–22: Report Builder Components and Subcomponents

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Common Files	M	–	–
NetSetup	O	–	–
Progress Messages (PROMSGS)	M	All Languages	O
Report Builder	M	Report Builder	M
		Report Builder Engine	M

F.2.23 Secure AppServer

Table F–23 lists the Secure AppServer components and subcomponents. Choosing the Complete option results in the installation of all components and subcomponents listed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–23: Secure AppServer Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL to JMS Adapter	M	Progress Class Files	M
		Sonic MQ Adapter	M
ActiveX Control Support	M	ActiveX Control Development Support	M
		ActiveX Control Runtime Support	M

Table F–23: Secure AppServer Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
AppServer Internet Adapter/S	M	–	–
AppServer runtime Client	M	–	–
AS/400 DataServer Client	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Intersolv ODBC Driver Support	**O**	–	–
MS SQL Server DataServer Client	**O**	–	–
NameServer	M	–	–
ODBC DataServer Client	**O**	–	–
Open AppServer	M	Application Server	M
		Base ADE	M
		Common Broker	M
		Progress Class Files	M
		XML	M
Oracle DataServer Client	**O**	–	–
ProBuild Utility	**O**	–	–
ProControl Utility	**O**	–	–

Table F–23: Secure AppServer Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	All Languages	O
Progress SQL ESQL/C Clients	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
Progress SQL ODBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**

F.2.24 SonicMQ Adapter

Table F–24 lists the SonicMQ Adapter components and subcomponents. Choosing the Complete option results in the installation of all components and subcomponents listed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–24: SonicMQ Adapter Components and Subcomponents

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL to JMS Adapter	M	Progress Class Files	M
		SonicMQ Adapter	M
Common Files	M	–	–
NameServer	M	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	**All Languages**	O

F.2.25 SQL Server7 Enterprise

Table F–25 lists the SQL Server7 Enterprise components and subcomponents. (The DataServer for Microsoft SQL Server is compatible with Microsoft SQL Server 7 and later.)

Choosing the Complete option results in the installation of all components and subcomponents listed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–25: SQL Server7 Enterprise Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ActiveX Control Support	M	ActiveX Control Runtime Support	M
AS/400 DataServer Client	**O**	–	–
Character Base Tools	**O**	**ADM Runtime — CHAR**	**O**
		Compile Tool — CHAR	**M**
		Procedure Editor — CHAR	**M**
Character Database Admin Tools	**O**	–	–
Character Image — Dev	**O**	–	–
Character Runtime Client — Dev	**O**	–	–
Common Files	M	–	–

Table F–25: SQL Server7 Enterprise Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M
Graphical Base Tools	M	**ADM Runtime — GUI**	**O**
		Base ADE	M
		Compile Tool — GUI	**O**
		Desktop	M
		Help Compiler — GUI	**O**
		Procedure Editor — GUI	M
MS SQL Server DataServer	**M**	–	–
MS SQL Server DataServer Client	**M**	–	–
MS SQL Server DataServer Drivers	M	–	–
ODBC DataServer Client	**O**	–	–
Oracle DataServer Client	**O**	–	–
ProBuild Utility	M	–	–

Table F–25: SQL Server7 Enterprise Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ProControl Utility	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	**All Languages**	O
Report Engine	M	–	–
Runtime Client	M	Base Client — RT	M
		Graphical Client	M
		XML	M
Schema Holder Server	M	–	–
Server Networking	M	–	–

F.2.26 SQL Server7 Personal

Table F–26 lists the SQL Server7 Personal components and subcomponents. (The DataServer for Microsoft SQL Server is compatible with Microsoft SQL Server 7 and later.)

Choosing the Complete option results in the installation of all components and subcomponents listed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–26: SQL Server7 Personal Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ActiveX Control Support	M	ActiveX Control Runtime Support	M
AS/400 DataServer Client	**O**	–	–
Character Base Tools	**O**	**ADM Runtime — CHAR**	**O**
		Compile Tool — CHAR	**M**
		Procedure Editor — CHAR	**M**
Character Database Admin Tools	**O**	–	–
Character Image — Dev	**O**	–	–
Character Runtime Client — Dev	**O**	–	–
Common Files	M	–	–

Table F–26: SQL Server7 Personal Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M
Graphical Base Tools	M	**ADM Run-time — GUI**	O
		Base ADE	M
		Compile Tool — GUI	O
		Desktop	M
		Help Compiler — GUI	O
		Procedure Editor — GUI	M
MS SQL Server DataServer Client	M	–	–
ODBC DataServer Client	O	–	–
Oracle DataServer Client	O	–	–
ProBuild Utility	M	–	–
ProControl Utility	**O**	–	–
Progress Messages (PROMSGS)	M	**All Languages**	O

Table F–26: SQL Server7 Personal Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Report Engine	M	–	–
Runtime Client	M	Base Client — RT	M
		Graphical Client	M
		XML	M
Schema Holder Server	M	–	–

F.2.27 SQL-92 Client Access

Table F–27 lists the SQL-92 Client Access components and subcomponents. When you choose the Complete installation option and install SQL Client Access, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–27: SQL-92 Client Access Components and Subcomponents *(1 of 2)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Common Files	M	–	–
NetSetup	**O**	–	–
Progress Explorer Tools — SQL	M	Explorer tools	M
		JRE	M
		Progress Class Files	M
Progress Messages (PROMSGS)	M	**All Languages**	O

Table F–27: SQL-92 Client Access Components and Subcomponents *(2 of 2)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress SQL ESQL/C Clients	M	ESQL Client	M
		JRE	M
		Progress Class Files	M
		SQL Common	M
Progress SQL JDBC Clients	M	JRE	M
		Progress Class Files	M
		SQL Common	M
		SQL JDBC Client	M
Progress SQL ODBC Clients	M	JRE	M
		Progress Class Files	M
		SQL Common	M
		SQL ODBC Client	M

NOTE: When you choose a Complete installation and install SQL Client Access, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.28 Toolkit

Table F–28 lists the Toolkit components. Choosing the Complete option results in the installation of all components; choosing the Typical option excludes the NetSetup component.

Table F–28: Toolkit Components

Component	Mandatory/ Optional	Subcomponent	Mandatory/Optional
Common Files	M	–	–
NetSetup	O	–	–
Toolkit	M	–	–

F.2.29 Translation Manager

Table F–29 lists the Translation Manager components and subcomponents. Choosing the Complete option results in the installation of all components and subcomponents listed; choosing the Typical option excludes the NetSetup component.

Table F–29: Translation Manager Components and Subcomponents

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ActiveX Control Support	M	ActiveX Control Runtime Support	M
Common Files	M	–	–
NetSetup	O	–	–
Translation Manager	M	Base ADE	M
		Translation Manager	M
Visual Translator	M	–	–

F.2.30 Visual Translator

Table F–30 lists the Visual Translator components and subcomponents. When you choose the Complete installation option and install the Visual Translator, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–30: Visual Translator Components and Subcomponents *(1 of 2)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client — GL	M
		Graphical Client	M
		XML	M
ActiveX Control Support	M	ActiveX Control Runtime Support	M
Common Files	M	–	–
Database Server Component	M	4GL Database	M
		4GL Server	M
		Database Server	M
Graphical Base Tools	M	**ADM Runtime — GUI**	O
		Base ADE	M
		Compile Tool — GUI	O
		Desktop	M
		Help Compiler — GUI	O
		Procedure Editor — GUI	M
NetSetup	**O**	–	–
ProBuild Utility	**O**	–	–

Table F–30: Visual Translator Components and Subcomponents *(2 of 2)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress Messages (PROMSGS)	M	All Languages	O
Report Engine	M	–	–
SQL Database Server	**O**	Database Server	**M**
		Database Utilities	**M**
		JRE	**M**
		Progress Class Files	**M**
		Progress Databases	**M**
		SQL Server	**M**
Visual Translator	M	–	–

NOTE: When you choose a Complete installation and install the Visual Translator, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.31 WebClient

Table F–31 lists the WebClient component. When you choose the Complete or the Typical installation option and install the WebClient, the components listed are installed.

Table F–31: WebClient Component

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Secure AppServer Client	M	–	–
WebClient-Common	M	–	–

F.2.32 WebSpeed Development Server

Table F–32 lists the WebSpeed Development Server components and subcomponents. When you choose the Complete installation option and install the WebSpeed Development Server, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–32: WebSpeed Development Server Components and Subcomponents *(1 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client — 4GL	M
		Graphical Client	M
		XML	M
4GL to JMS	M	Progress Class Files	M
		SonicMQ Adapter	M
ActiveX Control Support	M	ActiveX Control Runtime Support	M
AS/400 DataServer Client	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M
Database Server Component	M	4GL Database	M
		4GL Server	M
		Database Server	M

Table F–32: WebSpeed Development Server Components and Subcomponents *(2 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Debugger	**O**	–	–
Intersolv ODBC Driver Support	**O**	–	–
Messenger Component	M	–	–
MS SQL Server DataServer Client	**O**	–	–
NameServer	M	–	–
ODBC DataServer Client	**O**	–	–
Open AppServer	M	Application Server	M
		Base ADE	M
		Common Broker	M
		Progress Class Files	M
		XML	M
Oracle DataServer Client	**O**	–	–
ProBuild Utility	**O**	–	–
ProControl Utility	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M

Table F–32: WebSpeed Development Server Components and Subcomponents *(3 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress Messages (PROMSGS)	M	**All Languages**	O
Progress SQL ESQL/C Clients	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
Progress SQL ODBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**
SQL Database Server — Development	**O**	**Database Server**	**M**
		Database Utilities	**M**
		JDK	**M**
		Progress Class Files	**M**
		SQL Server	**M**

Table F–32: WebSpeed Development Server Components and Subcomponents *(4 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Transaction Server	M	Progress Databases	M
		Web Static	M
		WebSpeed Runtime	M
		WebSpeed Server Development	M
		XML	M

NOTE: When you choose a Complete installation and install the WebSpeed Development Server, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** are installed.

F.2.33 WebSpeed Enterprise Transaction Server

Table F–33 lists the WebSpeed Enterprise Transaction Server components and subcomponents. When you choose the Complete installation option and install the WebSpeed Enterprise Transaction Server, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–33: WebSpeed Enterprise Transaction Server Components and Subcomponents *(1 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client — 4GL	M
		Graphical Client	M
		XML	M

Table F–33: WebSpeed Enterprise Transaction Server Components and Subcomponents *(2 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL to JMS Adapter	M	Progress Class Files	M
		SonicMQ Adapter	M
ADE Source Code	**O**	**4GL Profiler**	**O**
		ADE Common Source	**O**
		Demo Applications	**O**
		Editor Source	**O**
AS/400 DataServer Client	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Intersolv ODBC Driver Support	O	–	–
Messenger Component	M	–	–
MS SQL Server DataServer Client	**O**	–	–
NameServer	M	–	–
ODBC DataServer Client	**O**	–	–
Open AppServer	M	Application Server	M
		Base ADE	M
		Common Broker	M
		Progress Class Files	M
		XML	M
Oracle DataServer Client	**O**	–	–

Table F–33: WebSpeed Enterprise Transaction Server Components and Subcomponents *(3 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ProBuild Utility	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	**All Languages**	O
Progress SQL ESQL/C Clients	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
Progress SQL ODBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**

Table F–33: WebSpeed Enterprise Transaction Server Components and Subcomponents *(4 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Transaction Server	M	Progress Databases	M
		Web Static	M
		WebSpeed Runtime	M
		WebSpeed Server Development	M
		XML	M

NOTE: When you choose a Complete installation and install the WebSpeed Enterprise Transaction Server, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.34 WebSpeed Messenger

Table F–34 lists the WebSpeed Messenger components. Choosing the Complete or the Typical option results in the installation of all components listed.

Table F–34: WebSpeed Messenger Components and Subcomponents

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Messenger Component	M	–	–
Progress Messages (PROMSGS)	M	All Languages	O
Web Static	M	–	–
WebSpeed Common	M	–	–

F.2.35 WebSpeed Transaction Server

Table F–35 lists the WebSpeed Transaction Server components and subcomponents. When you choose the Complete installation option and install the WebSpeed Transaction Server, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–35: WebSpeed Transaction Server Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client — GL	M
		Graphical Client	M
		XML	M
4GL to JMS Adapter	M	Progress Class Files	M
		SonicMQ Adapter	M
ADE Source	**O**	**4GL Profiler**	**O**
		ADE Common Source	**O**
		Demo Applications	**O**
		Editor Source	**O**
AS/400 DataServer Client	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Intersolv ODBC Driver Support	**O**	–	–
Messenger Component	M	–	–
MS SQL Server DataServer Client	**O**	–	–
NameServer	M	–	–

Table F–35: WebSpeed Transaction Server Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
ODBC DataServer Client	**O**	–	–
Open AppServer	M	Application Server	M
		Base ADE	M
		Common Broker	M
		Progress Class Files	M
		XML	M
Oracle DataServer Client	**O**	–	–
ProBuild Utility	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	All Languages	O
Progress SQL ESQL/C Clients	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**

Table F–35: WebSpeed Transaction Server Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress SQL JDBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
Progress SQL ODBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**
Transaction Server	M	Progress Databases	M
		Web Static	M
		WebSpeed Runtime	M
		WebSpeed Server Development	M
		XML	M

NOTE: When you choose a Complete installation and install the WebSpeed Transaction Server, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.36 WebSpeed Workshop

Table F–36 lists the WebSpeed Workshop components and subcomponents. When you choose the Complete installation option and install WebSpeed Workshop, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

Table F–36: WebSpeed Workshop Components and Subcomponents *(1 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL Client	M	Base Client — GL	M
		Graphical Client	M
		XML	M
4GL to JMS Adapter	**O**	Progress Class Files	**M**
		SonicMQ Adapter	**M**
ActiveX Control Support	M	ActiveX Control Development Support	M
		ActiveX Control Runtime Support	M
ADE Source Code	M	**4GL Profiler**	O
		ADE Common Source	M
		ADM Source Code	O
		DB Administration Source	O
		Demo Applications	O
		Editor Source	O
		ProTools Source Code	M

Table F–36: WebSpeed Workshop Components and Subcomponents *(2 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
AppBuilder — WebSpeed	M	AppBuilder Core	M
		Base ADE	M
		WebSpeed AppBuilder	M
AS/400 DataServer Client	**O**	–	–
Client Networking	M	–	–
Common Files	M	–	–
Database Administration Tools	M	4GL Database	M
		Base ADE	M
		Database Utilities	M
		Graphical Administration Tools	M
Database Server Component	M	4GL Database	M
		4GL Server	M
		Database Server	M
Development Server	**O**	Application Server	**M**
		Common Broker	**M**
		Progress Class Files	**M**
		XML	**M**
Intersolv ODBC Driver Support	**O**	–	–
Messenger Component	**O**	–	–
MS SQL Server DataServer Client	**O**	–	–

Table F–36: WebSpeed Workshop Components and Subcomponents *(3 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
NameServer	**O**	–	–
NetSetup	**O**	–	–
ODBC DataServer Client	**O**	–	–
Oracle DataServer Client	**O**	–	–
ProBuild Utility	**O**	–	–
Progress Explorer Tools	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
		Ubroker Tools	M
Progress Messages (PROMSGS)	M	All Languages	O
Progress SQL ESQL/C Clients	**O**	**ESQL Client**	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
Progress SQL JDBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**

Table F–36: WebSpeed Workshop Components and Subcomponents *(4 of 4)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Progress SQL ODBC Clients	**O**	**JRE**	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL ODBC Client	**M**
SQL Database Server — Development	**O**	**Database Server**	**M**
		Database Utilities	**M**
		JDK	**M**
		Progress Class Files	**M**
		Progress Databases	**M**
		SQL Server	**M**
Transaction Server	**O**	**Progress Databases**	**M**
		Web Static	**M**
		WebSpeed Runtime	**M**
		WebSpeed Server Development	**M**
		XML	**M**
WebSpeed Workshop	M	Advanced Editing	M
		Procedure Editor — GUI	M
		Progress Databases	M
		Workshop	M

NOTE: When you choose a Complete installation and install WebSpeed Workshop, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

F.2.37 Workgroup Database

Table F–37 lists the Workgroup Database components and subcomponents. When you choose the Complete installation option and install the Workgroup Database, all components and subcomponents listed are installed. Choosing the Typical installation option includes all components and subcomponents except those that appear in **bold** text.

NOTE: When you install a Progress database product, the **Select Server Engines** dialog box will appear, asking you to choose to install either the 4GL or SQL-92 server engine, or both (the default selection is to install both the 4GL and SQL-92 server engines). Your choice of server engines will determine whether you have 4GL or SQL-92 access to the Progress data files. See Chapter 2, "Installing Progress," for more information on selecting server engines.

Table F–37: Workgroup Database Components and Subcomponents *(1 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL	**O**	**4GL Server**	**M**
		ActiveX Control Runtime Support	**M**
		ADM Runtime — CHAR	**O**
		ADM Runtime — GUI	**O**
		AS/400 Client	**O**
		Base ADE	**M**
		Base Client — DA	**M**
		Character Administration Tools	**O**
		Character Client — 4GL	**O**
		Character Image	**O**
		Compile Tool — CHAR	**O**

Table F–37: Workgroup Database Components and Subcomponents *(2 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
4GL (cont)	**O**	**Compile Tool — GUI**	**O**
		Desktop	**M**
		Graphical Administration Tools	**M**
		Graphical Client	**M**
		Help Compiler — GUI	**O**
		MS SQL Server DataServer Client	**O**
		ODBC DataServer Client	**O**
		Open Interface Driver	**O**
		Oracle Client	**O**
		Procedure Editor — CHAR	**O**
		Procedure Editor — GUI	**M**
		Report Engine	**M**
		Server Networking	**M**
		XML	**M**
Bastille Tools	M	Database Server	M
		Database Utilities	M
		Progress Databases	M
Common Files	M	–	–

Table F–37: Workgroup Database Components and Subcomponents *(3 of 3)*

Component	Mandatory/ Optional	Subcomponent	Mandatory/ Optional
Legacy Version 8.2 DB Utilities	**O**	–	–
Legacy Version 8.3 DB Utilities	**O**	–	–
NetSetup	**O**	–	–
ProBuild Utility	**O**	–	–
ProControl Utility	**O**	–	–
Progress Explorer Tools — DB	M	Administration Server	M
		Explorer tools	M
		JRE	M
		Progress Class Files	M
Progress Messages (PROMSGS)	M	All Languages	O
SQL	**O**	ESQL Client	**M**
		JRE	**M**
		Progress Class Files	**M**
		SQL Common	**M**
		SQL JDBC Client	**M**
		SQL ODBC Client	**M**
		SQL Server	**M**

NOTE: When you choose a Complete installation and install the Workgroup Database, all components and subcomponents listed in the table are installed. When you choose a Typical installation, all components and subcomponents listed **except** those that appear in **bold** text are installed.

Index

A

Access rights
  networks 6–26

ADAPTCONFIG utility 7–11

ADAPTMAN utility 7–6

Addressing
  Progress brokers/servers 8–13
  Progress hosts 8–13

AdminServer 6–2
  changing the
    Startup setting 5–12
  starting 5–10
  stopping 5–10

After-image writers (AIW) 6–12

AIACONFIG utility 7–11

AIAMAN utility 7–6

Application services 7–9
  default service 7–9

Application workstations
  defined 6–13

Applications
  starting on networks 8–13

AppServer Internet Adapter
  components installed during Typical or Complete installation F–7

AppServer Internet Adapter/S
  components installed during Typical or Complete installation F–8

AppServer Load Balancing Option
  components installed during Typical or Complete installation F–17

AppServer Maximum Prestart Counterl (-Ms) parameter 8–12

Arbitrary sum weight factors 7–32

AS/400
  default network type 8–13

ASBMAN utility 7–6

ASCONFIG utility 7–11

Asynchronous page writers (APWs) 6–12

B

Background mode
  multi-user Progress 8–17, 8–18, B–8
    Windows NT B–8
  single-user Progress 8–8, 8–9, B–4
    Windows NT B–4

Batch mode
multi-user Progress 8–17, 8–18, B–8
Windows NT B–8
setup.iss 2–35
single-user Progress B–4
Windows NT B–4

Batch startup commands
single-user 8–4

Before-image Filename (-g) startup parameter 6–24

Before-image writers 6–12

BPRO utility 8–8

Broadcast request to NameServer 7–35

Brokers
addressing 8–13
multi-broker access 8–14
starting 8–14
servers for remote users 8–9

C

Character sets
international settings 2–23, 2–33

Citrix MetaFrame
system requirements 1–11

Client Networking
components installed during Typical or Complete installation F–8

Client startup parameters
networks 8–11

Client/Server configurations 6–13
samples 6–15
terminology 6–13

Client/Server Progress
addressing 8–13
broker/server addressing 8–13
execution 8–11
starting clients and servers 8–11

Clients
defined 6–13
starting
TCP/IP on UNIX 8–14

Client-side security
components installed during Typical or Complete installation F–11

Command line
setup.iss
batch mode 2–35

Compatibility
4GL clients and database servers 1–15

Complete installation 2–8, 2–20 to 2–27

Components
installed during a Typical or Complete installation F–1 to F–86
4GL Development F–3 to F–7
AppServer Internet Adapter F–7
AppServer Internet Adapter/S F–8
AppServer Load Balancing Option F–17
Client Networking F–8 to F–11
Client-side Security F–11
Debugger F–12
Enterprise Database F–13 to F–15
ESQL-89/C F–16
NameServer F–16
ODBC Enterprise DataServer F–18 to F–20
ODBC Personal DataServer F–21 to F–23
Open Client Toolkit F–23
Oracle DataServer F–24 to F–26
Personal Database F–27 to F–29
Progress AppServer F–30 to F–32
ProVision F–32 to F–37
ProVision Development Server F–37 to F–41
ProVision Plus F–41 to F–47
ProVision Plus Development Server F–48 to F–51
Query/Results F–52, F–53
Report Builder F–54
Secure AppServer F–54 to F–56
SonicMQ Adapter F–57

SQL Server 2000 Enterprise F–58 to F–60
SQL Server 2000 Personal F–61 to F–63
Toolkit F–65
Translation Manager F–65
Visual Translator F–66, F–67
WebClient F–67
WebSpeed Development Server F–68 to F–71
WebSpeed Enterprise Transaction Server F–71 to F–74
WebSpeed Messenger F–74
WebSpeed Transaction Server F–75 to F–77
WebSpeed Workshop F–78 to F–82
Workgroup Database F–82 to F–85

Configuration utilities 6–3
ADAPTCONFIG 6–3
ADAPTMAN 6–3
AIACONFIG 6–3
ASBMAN 6–3
ASCONFIG 6–3
DBMAN 6–4
MSSCONFIG 6–4
MSSMAN 6–4
NSCONFIG 6–4
NSMAN 6–4
ODBCONFIG 6–4
ODBMAN 6–4
ORACONFIG 6–4
ORAMAN 6–4
PROADSV 6–4
WSCONFIG 6–4
WTBMAN 6–4

Configurations
client/server 6–13
displaying information 4–5
network 6–16
shared memory 6–8

Configuring
NameServers
using Progress Explorer 7–21

Configuring Progress on an NOS 6–25

Configuring Unified Broker components 7–18

Configuring Unified Broker products 7–1

Connection-level fault tolerance 7–7
two techniques 7–34
understanding 7–33

Control number
displaying 4–4

Controlling NameServer 7–7

Creating
and configuring a Version 9 server 5–13

Custom installation 2–8, 2–28 to 2–34

D

Data Administration Tool
creating a database by using 5–13

Data Dictionary
creating a database by using 5–13

Database
sample application C–3

Database server machines
defined 6–13
multi-process
defined 6–14
single-process
defined 6–14

Databases
monitoring 6–12
startup 8–13

Date format
international settings 2–23, 2–33

DBMAN 6–4

Debugger
components installed during Typical or Complete installation F–12

Default service 7–9

DefaultFont environment variable
  international progress.ini files D–9

Directory structure
  Progress C–1

Disk space requirements 1–12

Distributing processes
  Unfied Broker 7–16

DLC
  directory structure C–2
  setting the environment variable 5–4

Downloading executables for
 heterogeneous environments 2–50

E

Editing ubroker.properties 7–22

Enterprise Database
  components installed during Typical or
    Complete installation F–13 to F–15

Environment variables
  Java 5–2
  setting 5–2
  Unified Broker settings 7–24
  Windows 5–3

ESQL-89/C
  components installed during Typical or
    Complete installation F–16

European Numeric Format (-E) D–8

Expiration date
  displaying 4–4

F

Fail-over weight factor 7–32

File types
  supported by Progress E–2

Files
  log
    configuring 7–25
  network sharing 6–26
  protocols, *See* protocols file
  services, *See* services file
  ubroker.properties 7–9

Finishing installing Progress 2–34

H

-H startup parameter
  client connection 8–11
  server startup 8–12
  specifying 8–13

Host addressing 8–13

Host Name (-H) parameter
  client connection 8–11
  server startup 8–12
  specifying 8–13

Host request to NameServer 7–35

I

Icons
  used by Progress E–3

Installation
  date
    displaying 4–4
  displaying configuration information 4–5

Installation path (DLC)
  directory structure C–2

Installing
  Visual Translator 2–3
  WebClient 2–3

Installing Progress 2–1 to 2–51
  and performing post-installation tasks
    2–49
  and viewing registry information 2–48
    for ODBC driver 2–48

as a Complete installation 2–20 to 2–27
as a Custom installation 2–28
as a Typical installation 2–20 to 2–27
Components
in a Typical or Complete installation F–1 to F–86
finishing 2–34
in a heterogeneous environment 2–50
on a network 3–1
options F–2
products to a current installation 2–39
to begin 2–7 to 2–9

Interactive mode
multi-user Progress 8–17

International
databases D–4
parameter files D–7

ISS batch mode 2–35 to 2–37

J

Java
SDK requirement 1–3
system requirements 1–2

Java Development Kit (JDK)
environment variables 5–2

Java Runtime Environment (JRE)
environment variables 5–2

JDKHOME
setting 5–2

L

LAN Manager
resource sharing 6–26

LAN Progress
startup parameters 8–11

Language
selecting default 2–22, 2–32
support D–1

License
displaying current information 4–3
information during installation 2–14
updating information 4–2

Licensed user counts 4–5

Licenses 4–5

Load balancing
arbitrary sums 7–32
fail-over 7–32
percentage-based 7–32
priority weight factors 7–31

Local NameServer instances 7–19

Location transparency 7–7

Log (.lg) files
commands to removing entries 9–2

Log files
Unified Broker 7–25

M

-m2 startup parameter 8–12

-m3 startup parameter 8–12

-Ma startup parameter 8–12

Machine addressing 8–13

Machine distribution
Unified Broker products 7–16

Maintaining two versions of Progress 5–14

Mandatory components 2–30

Manual Server (-m2) startup parameter 8–12

Maximum Clients per Server (-Ma) parameter
server startup 8–12

Maximum Dynamic Server (-Maxport) parameter
server startup 8–12

Maximum Servers (-Mn) parameter
server startup 8–12

Maximum Servers per Broker (-Mpb) startup parameter
server startup 8–12

Maximum Servers per Protocol (-Mp) parameter
server startup 8–12

-Maxport startup parameter 8–12

MBPRO command 8–18

AppServer Maximum Maintained Prestart Counter (-Mms) parameter 8–12

Message Buffer Size (-Mm) parameter
client connection 8–11
server startup 8–12

-Mi startup parameter
server startup 8–12

Minimum Clients per Server (-Ma) parameter
server startup 8–12

Minimum Dynamic Server (-Minport) parameter
server startup 8–12

-Minport startup parameter
server startup 8–12

-Mm startup parameter
client connection 8–11
server startup 8–12

-Mms startup parameter 8–12

-Mn startup parameter
server startup 8–12

Modifying
a Progress installation 2–51

-Mp startup parameter
server startup 8–12

-Mpb startup parameter
server startup 8–12

MPRO command 8–14

MPRO utility 8–17

-Ms startup parameter
server startup 8–12

MSSCONFIG utility 7–11

MSSMAN utility 7–6

Multi-threaded servers
defined 6–13

Multi-user Progress 8–17, 8–18
commands to start
batch mode 8–17
interactive mode 8–17
NT service 8–10, B–8
starting the server 8–9
startup commands
batch mode 8–17
broker 8–10
interactive mode 8–17
NT service 8–9, 8–10, B–8
server 8–9, 8–10

N

-N startup parameter
client connection 8–11
defaults 8–13
server startup 8–12
specifying 8–13

NameServer
accessing its port and host 7–17
broadcast address 7–37
client connections 7–20
components installed during Typical or Complete installation F–16
configuring
using Progress Explorer 7–21

configuring communications 7–17
editing the services file 7–18
features 7–7
instance configuration 7–19
log file 7–25
neighbors 7–34
example 7–40
using 7–38
replication 7–34
example 7–36
using 7–35
starting instances
using NSMAN utility 7–29
using Progress Explorer 7–27
UDP broadcasting 7–35

NameServers
downloading executables 7–19

Neighbor NameServers 7–34

NETBIOS network types 6–17

Network
file server 3–2
shared installation 3–1

Network configurations 6–16

Network file servers
applications 6–19, 6–21
as database server machine 6–20
defined 6–14

Network operating system
access rights 6–26
accessing resources 6–25
Novell NetWare 6–26
Progress startup 8–11
resource sharing 6–26
LAN Manager 6–26

Network startup parameters 8–11

Network Type (-N) parameter
client connection 8–11
defaults 8–13
server startup 8–12
specifying 8–13

Network Type (-N) startup parameter
protocol support 6–17

Network types
defined 6–17

Network Version (-Nv) startup parameter
client connection 8–11

Networks
access rights 6–26
accessing resources 6–25
Novell NetWare 6–26
applications
starting 8–13
client startup parameters 8–11
client/server
startup 8–13
database server machines 6–13
file servers 6–14
PC LAN permissions 6–26
Progress startup parameters 8–11
resource sharing 6–26
LAN Manager 6–26
server startup parameters 8–12
types 6–17

Novell NetWare
resource sharing 6–26

NSCONFIG utility 7–11
syntax 7–22
using 7–22

NSMAN utility 7–6
syntax 7–29
using 7–29

NT
default network type 8–13

Number format
international settings 2–23, 2–33

-Nv startup parameter 8–11

O

ODBC Enterprise DataServer
components installed during Typical or Complete installation F–18 to F–20

ODBC Personal DataServer
components installed during Typical or Complete installation F–21 to F–23

ODBCONFIG utility 7–12

ODBMAN utility 7–6

Open Client Toolkit
components installed during Typical or Complete installation F–23
Windows NT on Intel requirements 1–5, 1–6, 1–7, 1–11

Oracle DataServer
components installed during Typical or Complete installation F–24 to F–26

ORACONFIG utility 7–12

ORAMAN utility 7–6

P

Parameter file
typical startup parameters D–7, D–8

Percentage-based weight factors 7–32

Permissions
networks 6–26

Personal Database
components installed during Typical or Complete installation F–27 to F–29

Post-installation tasks 2–49

Preinstallation tasks 2–3 to 2–6
for SQL-92 install 2–5
for WebSpeed install 2–4
on a shared network 3–2, 3–3, 3–4

PrinterFont environment variable
international progress.ini files D–9

Priority weight factor
load balancing, *See* Weight factors

priorityWeight property 7–31

PRO utility 8–8

PROAB section in progress.ini 2–34

Procedure Editor startup commands
UNIX, Windows 8–4

Process distribution
Unified Broker 7–16

ProControl utility 8–9, 8–18, B–4, B–8

PRODB
creating a database by using 5–13

Product configurations
ubroker.properties file 6–4

Product Life Cycle 1–15, 1–18

Product name
displaying 4–4

Program Group
Progress A–1 to A–4
changing properties for items in the A–2, A–3

Program item properties
setting 5–3

Progress
broker/server addressing 8–13
client/server startup 8–11
configuring on an NOS 6–25
directory structure C–1
icons E–3
installing 2–1 to 2–51
integration with Windows Explorer E–1 to E–3
maintaining two versions of 5–14
messages D–5
network addressing 8–13

network startup parameters 8–11
Product Life Cycle 1–15, 1–18
Program Group A–1 to A–4
icons A–3
starting multi-user
batch mode 8–17, B–8
interactive mode 8–17
NT service 8–18
starting single-user
batch mode 8–8, 8–9, B–4
interactive mode 8–8
supported file types E–2
Typical and Complete installation components F–1 to F–86
Update utility 4–2

Progress AppServer
components installed during Typical or Complete installation F–30 to F–32
configuring 7–1

Progress client/server
network file server
application files 6–19, 6–21
network file server and
as database server machine 6–20
running on a LAN 8–11
sample configurations 6–17
startup parameters 8–11
with Progress AppServer 6–22

Progress DataServers
configuring with NameServer 7–1

Progress Explorer 6–2
configuring
NameServers 7–21
configuring Unified Broker products 7–20
framework 6–1, 6–2

Progress installation
TCP/IP networks 6–23

Progress network types
NETBIOS 6–17
SNA 6–17
TCP 6–17

Progress sample configurations 6–17
network file servers
application files 6–19, 6–21
as database server machine 6–20
with Progress AppServer 6–22

Progress Unified Broker
administration framework 7–2

Progress Unified Broker, *See* Unified Broker

Progress Update utility 4–1

Progress.ini file
editing D–8
Windows 5–3

Prolang directory D–3

PROLOG
removing log entries 9–2

PROMON utility 6–12

PROMSGS D–5
environment variable D–9
removing prior to a reinstall 2–6
shell integration E–2
translations D–2

Properties
of application icons A–3

Properties file
information added by Progress E–3
ubroker.properties 6–4
Unified Broker 7–9
editing and validating 7–22

PROSERVE command 8–14

PROSERVE utility 8–14

ProVision
components installed during a Typical or Complete installation F–32 to F–37

ProVision Development Server
components installed during Typical or Complete installation F–37 to F–41

ProVision Plus
components installed during Typical or Complete installation F–41 to F–47

ProVision Plus Development Server
components installed during Typical or Complete installation F–48 to F–51

PROWDOG utility 6–12

Q

Query/Results
components installed during Typical or Complete installation F–52, F–53

R

Regional support
installing PROMSGS D–4

Registry
editing D–8
Windows 5–3

Remote NameServer instances 7–19

Removing log file entries
PROLOG 9–2

Replicated NameServers 7–34

Report Builder
components installed during Typical or Complete installation F–54

Resources
network sharing 6–26

Response file
creating automatically 2–36
creating manually 2–36

Running Progress 5–14

S

-S startup parameter 8–13
client connection 8–11
server startup 8–12

Sample application database C–3

Saving an existing Progress installation 2–5, 2–7

SDK system requirements 1–3

Secondary Login Broker (-m3) startup parameter
server startup 8–12

Secure AppServer
components installed during Typical or Complete installation F–54 to F–56

Serial number
displaying 4–4

Server startup parameters
networks 8–12

Server-level fault tolerance 7–7
understanding 7–33

Servers
addressing 8–13
defined 6–13
network 6–13
starting 8–14

Service Name (-S) startup parameter 8–13
client connection 8–11
server startup 8–12

services file 7–18

Setting
environment variables 5–2
Java environment variables 5–2
JDKHOME 5–2

Setting up the Progress environment 5–1 to 5–8
and environment variables 5–2
and Java environment variables 5–2
and the DLC variable 5–4
for the [Startup] section 5–3, 5–4, 5–6, 5–7
for the [WinChar Startup] section 5–3, 5–4, 5–6, 5–7
program item properties 5–3

Setup.iss
sample file 2–37

Shared memory
architecture 6–11
configurations 6–8
defined 6–8

Shared network
preinstallation tasks when installing on a 3–2, 3–3, 3–4

Shared Network Installation utility 2–9, 3–1 to 3–5

Sharing a Progress installation on a network 3–1 to 3–5

Shortcut menu
using with Progress E–3

Show Configuration (SHOWCFG) utility 4–2, 4–4, 4–5

Single-threaded servers
defined 6–13

Single-user Progress
startup commands
background mode 8–8, B–4
batch mode 8–8, 8–9, B–4
interactive mode 8–8
NT service 8–9, B–4

SNA network types 6–17

SonicMQ Adapter
components installed during Typical or Complete installation F–57

SQL Server 2000 Enterprise
components installed during Typical or Complete installation F–58 to F–60

SQL Server 2000 Personal
components installed during Typical or Complete installation F–61 to F–63

SQL-92
system requirements 1–3

SQL-92 Client Access
components installed during Typical or Complete installation F–63, F–64

Starting
clients and servers 8–11
network applications 8–13

Starting Progress
multi-user
batch mode 8–17, B–8
interactive mode 8–17
NT service 8–10, 8–18
with a server 8–9
single-user 8–9
batch mode 8–8, B–4
interactive mode 8–8

Starting Unified Broker instances 7–26

Startup commands
summary 8–2

Startup parameters
client 8–11
deployed applications D–7
server 8–12

System requirements 1–2
Citrix MetaFrame 1–9, 1–11
disk space 1–12
Java 1–2
SQL-92 1–3
Windows 2000 1–6
Windows 95 1–8
Windows 98 1–7
Windows NT on Intel 1–4, 1–5, 1–6

T

TCP network types 6–17

TCP/IP networks
  installing Progress 6–23
  network files 6–25
  preparing for Progress 6–25
  without file server 6–24

Toolkit
  components installed during Typical or Complete installation F–65

Translation Manager
  components installed during Typical or Complete installation F–65

Types of Unified Brokers 7–8

Typical installation 2–8, 2–20 to 2–27

U

ubroker.properties file 6–4, 7–9
  editing, *See* Properties file
  hierarchy 6–5
  products supported 6–6
  validating, *See* Properties file

UDP broadcasting 7–35
  performance implications 7–41

Unified Broker
  administration overview 7–12
  administration task summary 7–3
  application services 7–9
  clients 7–4
  command-line configuration 7–11
  command-line management 7–6
  configuring the components 7–18
  environment setup 7–24
  log files 7–25
  machine and process distribution 7–16
  NameServer instances 7–20
  order of configuration 7–19
  product features 7–8
  Progress Explorer configuration 7–20
  properties file 7–9
    guidelines for editing 7–11
    validating 7–11
  setting environment variables 7–24
  setting the working directory 7–24
  starting instances
    using management utilities 7–28
    using Progress Explorer 7–26
  types 7–8

Uninstalling Progress 2–51 to 2–57
  by using the Uninstall or Add/Remove Programs Utility 2–52 to 2–54
  manually 2–54 to 2–57

UNIX
  default network type 8–13
  starting network clients 8–14

Updating
  license information 4–2
  Progress installation 4–1 to 4–5

Upgrading
  a Progress installation 2–51 to 2–57

User licenses, *See* Licenses

Utilities
  BPRO 8–8
  MBPRO 8–18
  MPRO 8–17
  PRO 8–8
  ProControl 8–9, 8–18, B–4, B–8
  PROMON 6–12
  PROSERVE 8–14
  PROWDOG 6–12
  Watchdog 6–12

V

Version number
  displaying 4–4

Viewing registry information 2–48

Visual Translator
  components installed during a Typical or Complete installation F–66, F–67

W

Watchdog utility 6–12

WebSpeed Development Server
components installed during Typical or Complete installation F–68 to F–71

WebSpeed Enterprise Transaction Server
components installed during Typical or Complete installation F–71 to F–74

WebSpeed Messenger
components installed during Typical or Complete installation F–74

WebSpeed Transaction Server
components installed during Typical or Complete installation F–75 to F–77
configuring 7–1

WebSpeed Workshop
components installed during Typical or Complete installation F–78 to F–82

Windows 95
system requirements 1–8

Windows 98
system requirements 1–6, 1–7

Windows Explorer
integrating with Progress E–1 to E–3

Windows NT on Intel
system requirements 1–4, 1–5, 1–6

Workgroup Database
components installed during Typical or Complete installation F–82 to F–85

Working directory 5–9
Unfied Broker setting 7–24

WSCONFIG utility 7–12

WTBMAN utility 7–6